Joint stereo camera calibration and multi-target tracking using the linear-complexity factorial cumulant filter

Joint stereo camera calibration and multi-target tracking using the linear-complexity factorial cumulant filter

Mark Campbell and Daniel E. Clark

***Abstract*—The calibration of an unknown sensor, such as a camera, is a key issue in the sensor fusion domain. This paper addresses this problem by expanding upon previously introduced work. This method uses a unified Bayesian framework with an alternative parameterisation known as disparity space to calibrate an unknown camera's spatial parameters in reference to a known camera. Here, the recently developedLinear-Complexity Cumulant (LCC) filter is used to improve the both the multi-target tracking and calibration facets of the framework. The new implementation is compared against a Probability Hypothesis Density (PHD) method upon simulated data.**

I. Introduction

Visual target tracking using multiple cameras is of great interest in a wide variety of applications such as visual servoing [1], robotic navigation [2], gaze tracking [3] and also numerous other civilian and defence scenarios. Multiple cameras are used in order to obtain a target's 3-D parameters (position, velocity) as depth is not inherently obtainable from a single camera. In a typical stereo camera configuration, two cameras observe the same scene but separated by some offset and/or rotation. If the spatial parameters between the cameras are well known, the cameras are calibrated and target's 3-D parameters can be determined through triangulation [4] or projective geometry [5].

The work in this paper expands upon the work in [6], in which a unified Bayesian framework was proposed to jointly estimate the cameras' spatial parameters and track the multiple targets present in the observed 3-D scene. The framework uses an alternative state space, called the disparity space [7], which allows for certain parts of the estimation process to be transformed into a linear Gaussian form so that traditional tracking algorithms such as the Kalman filter can used. The framework utilizes an approach that was initially developed for group object tracking through the use of hierarchical cluster point processes [8, 9, 10]. In this case, the number of groups is restricted to one allowing each object to represent a possible sensor configuration [11]. This *single cluster* method has been applied to applied to multiple sensor fusion problems including: Simultaneous Localisation and Mapping (SLAM) [12], correcting image drift in astronomical applications [13, 14], estimating drift in microscopy image sequences [15] and multi-sensor localisation [16].

In order to track multiple targets present in the scene, the framework utilizes the Probability Hypothesis Density (PHD) filter [17, 18], which is a popular low-cost solution to the multi-target estimation problem. The PHD uses Poisson modelling assumptions to propagate the first order information, the mean number of targets, of the multi-target density. A newer development in multi-target tracking is the Linear-Complexity Cumulant (LCC) filter [19], which is a point process derived filter that propagates both the first and second order factorial cumulants of the multi-target density, in linear complexity with respect to the number of targets. This achieved by using a Panjer point process prior as opposed to the Poisson assumptions used in the PHD filter [17]. This paper integrates the framework in [6] with the LCC filter [19]. This implementation aims to provide a more robust framework as the LCC filter has been shown to cope better with different false alarms models as well as lower probabilities of detection [19, 20].

This paper is structured as follows: Object triangulation and the disparity space parameterisation is described in Section II. A brief overview of the LCC filter is shown in Section IV. Section III details how the joint multi-target and camera calibration algorithm is structured. The new aclcc implementation is compared against the current PHD implementation and results are displayed in Section V. Finally conclusions are in Section VI.

Daniel Clark is with Telecom SudParis, France. Email: daniel.clark@telecom-sudparis.eu and Mark Campbell is with Heriot-Watt University, UK. Email: mc318@hw.ac.uk. This work was supported by AFSOR grant FA9550-18-C- 0201.

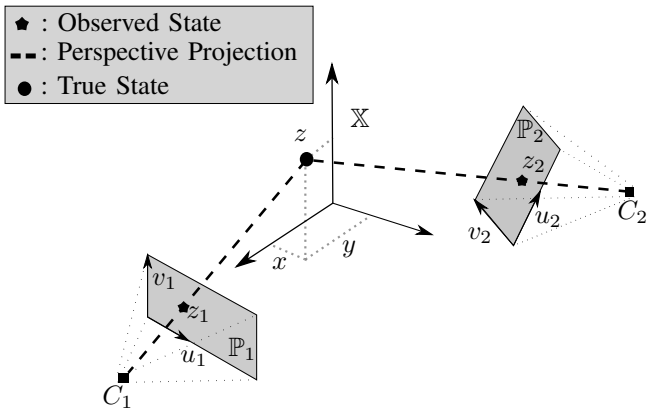

Fig. 1: Two cameras, C_1 and C_2 observing the same object x in the real world $\mathbb{X}$

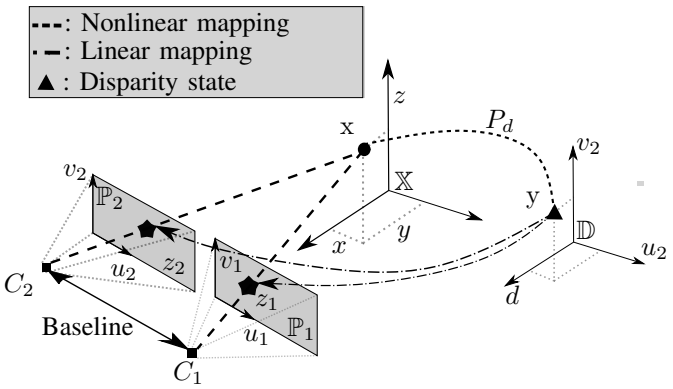

Fig. 2: A rectified camera pair, (C_1, C_2) with their related disparity space $\mathbb{D}$

II. Disparity Space

Traditional methods, such as those mentioned above [4, 5], for calculating an object's location in the real world 3-D space from a calibrated stereo camera configuration are highly useful in a number of scientific and engineering disciplines. Fig.1 illustrates a typical triangulation scenario, where the same point x in the real world $\mathbb{X} = \mathbb{R}^3$ is projected onto a camera's image plane $\mathbb{P}$ to obtain a projection point z. Recovering the point x from the the projections z_1 and z_2 can be performed if the respective perspective transformations are known [5]. However if integration into a Bayesian estimation paradigm is desired, then the uncertainties on the various parameters need to be modelled. Due to the non-linearity of the camera perspective projection, these uncertainties are highly range dependent [21]. This leads to difficulty when including in a Bayesian framework.

The disparity space introduced in [7] is an alternative parameteristion of the standard camera perspective projection that resolves the non-linear and non-Gaussian aspect of the projection, allowing for standard Bayesian tracking methods to be used. In standard projective geometry, a 2-D point $\mathrm{x} = [x, y]^T$ can represented using a triple $\bar{\mathrm{x}} = [\alpha x, \alpha y, \alpha]^T$, where $\alpha \in \mathbb{R}\backslash\{0\}$. Any such point is referred to as the *homogeneous coordinates* of the point x. The camera projection matrix P is a 3×4 matrix that describes the linear mapping between the homogeneous coordinates of a 3-D point in the real world $\bar{\mathrm{x}} = [x, y, z, 1]^T$ to a 2-D point in the image plane $\bar{\mathrm{z}} = [\bar{u}, \bar{v}, \bar{w}]^T$:

$$\bar{\mathrm{z}} \propto P\bar{\mathrm{x}}, \tag{1}$$

Note that $\propto$ denotes equality up to a scaling factor.

The concepts of disparity space and a rectified camera pair are closely related as shown in Fig.2. Assuming, that the projection matrix P_1 of the first camera C_1 is of the form $P_1 = K[I, 0]$, then the camera pair $[C_1, C_2]$ is horizontally rectified if the projection matrix P_2 of the second camera C_2 is of the form $P_2 = K[I, t]$, where $t = [b, 0, 0]^T$. This parameter b is known as the baseline. Hence if the cameras $[C_1, C_2]$ are horizontally rectified (Fig.2) with respective image planes $\mathbb{P}_1$ and $\mathbb{P}_2$, the real world point x is projected into $\mathbb{P}_1$ and $\mathbb{P}_2$ at the points $z_1 = [u_1, v_1]^T$ and $z_2 = [u_2, v_2]^T$. The real world point x can also be represented in the disparity space $\mathbb{D}$ by a point y:

$$y = [u_2, v_2, d]^T, \tag{2}$$

where $d = u_1 - u_2$. This also known as the disparity, as it measures the difference in view of x between the two cameras.

The transformation from the real world to the disparity space P_d can be expressed in terms of the camera projection matrices P_1 and P_2 and since the cameras are horizontally rectified:

$$(P_2)_i = (P_1)_i., \tag{3}$$

for $i = 2, 3$, where $(P)_i$ is the ith row of the matrix P, allowing the matrix P_d to be expressed as:

$$P_d = \begin{bmatrix} (P_2)_1. \\ (P_2)_2. \\ (P_1)_1 - (P_2)_1. \\ (P_2)_3. \end{bmatrix}. \tag{4}$$

Since the matrix P_d exists, the disparity space $\mathbb{D}$ can be used as an alternative space for triangulation and any point in $\mathbb{X}$ can be found using the inverse transformation of P_d. Another step to allow for triangulation to occur is to link the disparity space with the image planes. This is achieved through applying orthographic projections, H_1 and H_2, to the disparity space point y:

$$H_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad H_2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{5}$$

A fuller explanation and verification of the disparity space parameteristion can be seen in [6, 7].

III. Camera Calibration From Moving Objects

In order to calibrate the camera pair, the following assumptions must be stated:

- The origin of the chosen coordinate system is aligned with the first camera C_1
- The camera state vector $\mathbb{S}_2$ of the second camera C_2 contains its position and orientation in the disparity space $\mathbb{D}$ with the form $\mathbb{S}_2 = [u_1, v_1, d, \dot{u_1}, \dot{v_1}, \dot{d}]^T$.

To model the camera state vector, a particle filter is used such that at time k, the camera state vector is represented by a set of N particles y_k^i where $1 \leq i \leq N$. Each of these particles represent a possible camera state and each of the particles comes with a corresponding weight w_k^i and and a set of multi-object estimation statistics Θ_k^i. These statistics depend on the multi-object filter of choice: for the PHD filter, Θ_k^i is simply the intensity μ_k of the target population at time k or for the LCC it denotes the first $c_k^{(1)}$ and second $c_k^{(1)}$ order factorial cumulants, dependent on the parameter y_k^i. Similarly, the functions `Prediction` and `Update` are the respective prediction and update functions of the filter of choice, and the function `MOL` is the filter-specific multi-object likelihood of the respective filter.

Input
Set of particles $\{(y_k^i, w_k^i, \Theta_k^i)\}_{i=1}^N$
Set of measurements Z_{k+1}

Prediction
for $1 \leq i \leq N$ **do**
  Sample $y_{k+1}^i \sim \mathcal{N}(\mathbf{y}_k^i, \Sigma_s)$
  $\Theta_{k+1|k}^i \leftarrow$ `Prediction` (Θ_k^i)
end for

Update
for $1 \leq i \leq N$ **do**
  $\Theta_{k+1}^i \leftarrow$ `Update` $(\Theta_{k+1|k}^i, Z_{k+1})$
  $w_{k+1}^i \leftarrow$ `MOL` $(\Theta_{k+1|k}^i, Z_{k+1})$
end for

Resampling
$N_{\text{eff}} \leftarrow \dfrac{1}{\sum_{i=1}^N (w_{k+1}^i)^2}$
if $N_{\text{eff}} \leq r \cdot N$ **then**
  $\{y_{k+1}^i\}_{i=1}^N \leftarrow$ `Resampling` $(\{(y_{k+1}^i, w_{k+1}^i)\}_{i=1}^N)$
  for $1 \leq i \leq N$ **do**
    $w_{k+1}^i \leftarrow \frac{1}{N}$
  end for
end if

Algorithm 1: Algorithm for camera calibration.

The multi-object likelihood of the LCC [20] filter L of a given sensor state y at time k is given by:

$$L_k^y = -1^{|Z_k|}\left(1 + (\frac{1}{\beta_{k|k-1}})\right)(\lambda_{c,k} + \mu_k^d)^{(\alpha_{k|k-1}+|Z_k|)} \alpha_{k|k-1}^{(|Z|)} \prod_{z \in Z_k}\left[(\frac{1}{\beta_{k|k-1}})\mu_k^{z_j}\right] \tag{6}$$

where $\beta_{k|k-1} = (c_{k|k-1}^{(1)} + \lambda_{c,k})/(c_{k|k-1}^{(2)} + c_{c,k}^{(2)})$ and $\alpha_{k|k-1}^{(|Z|)}$ is a Pochhammer function, also known as a rising factorial.

Throughout this article, the roulette method was used to perform the resampling, but other approaches can be used instead [22]. Furthermore, resampling is only performed if the effective sample size N_{eff} is below a limit $N_{\text{eff}} \leq r \cdot N$. Where r is a user-defined parameter $0 \leq r \leq 1$. In this article, $r = 0.5$ was used.

IV. The Linear-Complexity Cumulant (LCC) Multi-Target Filter

A new linear-complexity multi-target filter was proposed in [19] that propagates the first and second order factorial cumulants, which shall be quickly summarized in pseudo-code here. Please refer to [19] for a full derivation. It should be noted that a Gaussian Mixture (GM) implementation of the filter is used here, where each individual possible target x is represented by a Gaussian. The the first order factorial moment $c^{(1)}$ at a time step k is the sum of the weights of the N_k Gaussians,

$$c^{(1)} = \sum_{i=1}^{N_k} w_k^{(i)} \mathcal{N}(x; m_k^{(i)}, P_k^{(i)}) \tag{7}$$

Each Gaussian $\mathcal{N}(x; m_k^{(i)}, P_k^{(i)})$ has a mean state vector $m_k^{(i)}$, covariance $P_k^{(i)}$ and weight $w_k^{(i)}$.

The LCC filter propagates the first and second order factorial cumulants $c^{(1)}$ and $c^{(2)}$ of a target process, from which the mean μ_k and variance σ_k^2 on the number of targets at time k can be extracted as seen in Eq.IV .

$$\mu_k = c^{(1)} \quad \text{and} \quad \sigma_k^2 = c^{(2)} + c^{(1)} \tag{8}$$

Input
Posterior target process: $\{w_{k-1}^{(i)}, m_{k-1}^{(i)}, P_{k-1}^{(i)}\}_{i=1}^{N_{k-1}}, c_{k-1}^{(2)}$
Newborn process: $\{w_{\mathrm{b},k-1}^{(i)}, m_{\mathrm{b},k-1}^{(i)}, P_{\mathrm{b},k-1}^{(i)}\}_{i=1}^{N_{\mathrm{b},k-1}}, c_{\mathrm{b},k}^{(2)}$
Transition Matrix: F_{k-1}, Transition Noise: Q_{k-1}
Probability of Target Survival: $p_{s,k}$
Survival process
$\mathrm{c}_{k-1}^{(1)} = \sum_{i=1}^{N_{k-1}} w_{k-1}^{(i)}$
for $1 \leq i \leq N_{k-1}$ **do**
$w_{k|k-1}^{(i)} = p_{\mathrm{s},k} w_{k-1}^{(i)}$
$m_{k|k-1}^{(i)} = F_{k-1} m_{k-1}^{(i)}$
$P_{k|k-1}^{(i)} = F_{k-1} P_{k-1}^{(i)} F_{k-1}^{T} + Q_{k-1}$
end for
$c_{\mathrm{s},k}^{(2)} = p_{\mathrm{s},k}^{2} c_{k-1}^{(2)}$
Newborn process
for $1 \leq j \leq N_{\mathrm{b},k-1}$ **do**
$w_{k|k-1}^{(n_{k-1}+j)} = w_{\mathrm{b},k-1}^{(j)}$
$m_{k|k-1}^{(n_{k-1}+j)} = m_{\mathrm{b},k-1}^{(j)}$
$P_{k|k-1}^{(n_{k-1}+j)} = P_{\mathrm{b},k-1}^{(j)}$
end for
$N_{k|k-1} = N_{k-1} + N_{\mathrm{b},k-1}$
$c_{k|k-1}^{(2)} = c_{\mathrm{b},k}^{(2)} + c_{\mathrm{s},k}^{(2)}$

Output
Predicted process: $\{w_{k|k-1}^{(i)}, m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\}_{i=1}^{N_{k|k-1}}, c_{k|k-1}^{(2)}$

Algorithm 2: Time prediction (time k).

Input
Predicted process: $\{w_{k|k-1}^{(i)}, m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\}_{i=1}^{N_{k|k-1}}, c_{k|k-1}^{(2)}$
Current measurements: $Z_k = \{z_j\}_{j=1}^{M_k}$
Clutter process: $\lambda_{c,k}, c_{c,k}^{(2)}$
Target Observation Matrix: H_k, Observation Noise: R_k
Probability of Target Detection: $p_{d,k}$
Panjer parameters
$c_{k|k-1}^{(1)} = \sum_{i=1}^{N_{k|k-1}} w_{k|k-1}^{(i)}$
$\alpha_{k|k-1} = (c_{k|k-1}^{(1)} + \lambda_{c,k})^2 / (c_{k|k-1}^{(2)} + c_{c,k}^{(2)})$
Missed detection and measurement terms
for $1 \leq i \leq N_{k|k-1}$ **do**
$w_{\phi,k}^{(i)} = (1 - p_{\mathrm{d},k}) w_{k|k-1}^{(i)}$
$m_{\phi,k}^{(i)} = m_{k|k-1}^{(i)}$
$P_{\phi,k}^{(i)} = P_{k|k-1}^{(i)}$
end for
$\mu_k^{\phi} = (1 - p_{\mathrm{d},k}) c_{k|k-1}^{(1)}$
$\mu_k^{d} = p_{\mathrm{d},k} c_{k|k-1}^{(1)}$
for $1 \leq j \leq M_k$ **do**
for $1 \leq i \leq N_{k|k-1}$ **do**
$y_k^{(i,j)} = z_j - H_k m_{k|k-1}^{(i)}$
$S_k^{(i)} = H_k P_{k|k-1}^{(i)} H_k^{T} + R_k$
$K_k^{(i)} = P_{k|k-1}^{(i)} H_k^{T} [S_k^{(i)}]^{-1}$
$w_{\mathrm{d},k}^{(i,j)} = p_{\mathrm{d},k} w_{\mathrm{d},k|k-1}^{(i,j)} \mathcal{N}(z; y_k^{(i,j)}, S_k^{(i)}) / (\mu_k^{z} + \lambda_{\mathrm{c},k})$
$m_{\mathrm{d},k}^{(i,j)} = m_{k|k-1}^{(i)} + K_k^{(i)} y_k^{(i,j)}$
$P_{\mathrm{d},k}^{(i,j)} = (I - K_k^{(i)} H_k) P_{k|k-1}^{(i)}$
end for
$\mu_k^{z_j} = \sum_{i=1}^{N_{k|k-1}} w_{\mathrm{d},k}^{(i,j)}$
end for
Corrective terms
$\ell_1(\phi) := (\alpha_{k|k-1} + |Z|) / (\alpha_{k|k-1} + \mu_k^{d} + \lambda_{\mathrm{c},k})$
$\ell_2(\phi) := (\alpha_{k|k-1} + |Z|) / (\alpha_{k|k-1} + \mu_k^{d} + \lambda_{\mathrm{c},k})^2$

Algorithm 3: Data update, first part: Compute single target - single measurement updates and corrective terms ℓ_1, ℓ_2

Missed detection terms
for $1 \leq i \leq N_{k|k-1}$ **do**
$w_k^{(i)} = \ell_1(\phi) w_{\phi,k}^{(i)}$
$m_k^{(i)} = m_{\phi,k}^{(i)}$
$P_k^{(i)} = P_{\phi,k}^{(i)}$
Association terms
for $1 \leq j \leq M_k$ **do**
$w_k^{(i \cdot n_{k|k-1}+j)} = w_{\mathrm{d},k}^{(i,j)}$
$m_k^{(i \cdot n_{k|k-1}+j)} = m_{\mathrm{d},k}^{(i,j)}$
$P_k^{(i \cdot n_{k|k-1}+j)} = P_{\mathrm{d},k}^{(i,j)}$
end for
end for
$N_k = N_{k|k-1} + N_{k|k-1} M_k$
$c_k^{(1)} = \sum_{i=1}^{N_k} w_k^{(i)}$
$c_k^{(2)} = (\mu_k^{\phi})^2 l_2(\phi) - \sum_{z \in Z_k} \left(\frac{\mu_k^z}{\mu_k^z + \lambda_{\mathrm{c},k}(z)} \right)^2$

Output
Posterior target process: $\{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{N_k}, c_k^{(2)}$

Algorithm 3: Data update, second part: Update component weights.

V. Numerical Experiments

Two scenarios are presented here to compare the performance of each filter, under simulations that vary parameters such as the the clutter rate and probability of detection. These scenarios intend to show differences of performance of the Probability Hypothesis Density (PHD) and Linear-Complexity Cumulant (LCC) implementations for different settings.

Targets are generated within a three-dimensional scenario within the 3-D "real world" region $[-25, 25] \times [-25, 25] \times [40, 150]$ (m $\times$ m $\times$ m), measured from the centre of the reference camera. Each target is described by its state vector $x = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^{\mathrm{T}}$, where $[x, y, z]$ specifies a position in 3-D Cartesian coordinates and $[\dot{x}, \dot{y}, \dot{z}]$ are the respective velocities. Each target is a "dynamic" target moving with nearly constant velocity model. The probability of survival is set as $p_s = 0.99$ for the targets.

The stereo camera configuration, which follows closely from the experiments performed in [6], is as follows: The cameras are located at $0m$ on both the y and z axes, at -20cm and 20cm along the x axis. The two cameras are respectively rotated $\pi/12$ and $-\pi/12$ radians around the y axis. In each Monte Carlo (MC) run, the second camera's position and orientation is modified with the following respective standard deviations: 0.2m, 0.5cm and 0.2cm on x, y and z as well as $\pi/180$ rad, $\pi/90$ rad and $\pi/180$ rad for the rotations about x, y and z. A camera obtains measurements in the disparity space $\mathbb{D}$ from the real world trajectories through the method mentioned above in Sec.II Target measurements are generated, or detected, with a probability p_d. False alarms are added to these measurements, according to a Poisson distribution with rate λ_c and are placed randomly across the surveillance region and transformed accordingly. Each scenario is simulated for $T = 80$ s time steps.

Case 1: $N_t = 7$ targets are initially generated in the scene and are detected with $p_d = 0.95$. The clutter rate is $\lambda_c = 1$. This is intended to demonstrate an close to an ideal scenario.

Case 2: A total of $N_t = 7$ targets are generated in the scene and are detected with $p_d = 0.80$. The clutter rate is $\lambda_c = 15$.

For both filters: the pruning threshold is $\tau_{\text{prn}} = 10^{-5}$, merging distance is $\tau_{\text{mrg}} = 7$. Measurements are gated with gate-size probability of $p_{\text{gate}} = 0.999$. Both filters are implemented using a GM approach. Since the parent process is implemented using a Sequential Monte Carlo (SMC) particle filter approach, a preset number of particles shall be used – 100 MC particles.

For each case, 50 MC runs are performed, each with independently generated clutter and (target-originated) measurements for each case. For both filters, performance is evaluated in terms of:

- Root Mean Square Error (RMSE) of the estimated camera state obtained from the Maximum A Posteriori (MAP) estimate of the parent process in both position and orientation.
- Estimated target cardinality obtained from the multi-object filter
- Execution time (per time step).

All results are averaged over these 50 MC runs.

A. Results

Case 1 : As seen in Figs.34, both filters are able to accurately estimate the sensor state (under 1.5cm). However LCC implementation is performs better (Fig.4), the longer the experiment runs. As expected, both filters run at approximately the same speed, as the LCC filter only has to propagate an additional number, the second order cumulant $c^{(2)}$. The difference in the inital RMSE estimates could be due to a number of factors. One such factor is that the particles are selected using a MAP approach with no averaging over the particles. Since the values for the likelihoods of the multi-object filters will be different, a different particle may be selected. This requires further work to analyse to exact reason.

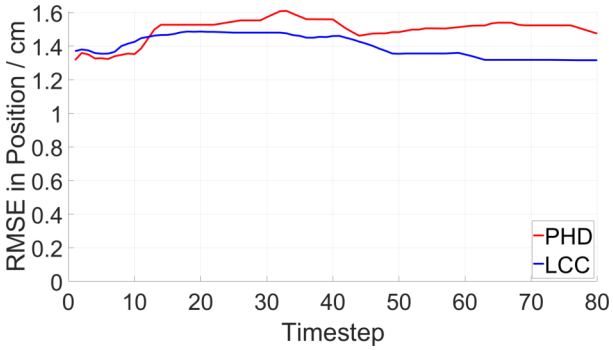

Fig. 3: RMSE of camera position estimates for case 1

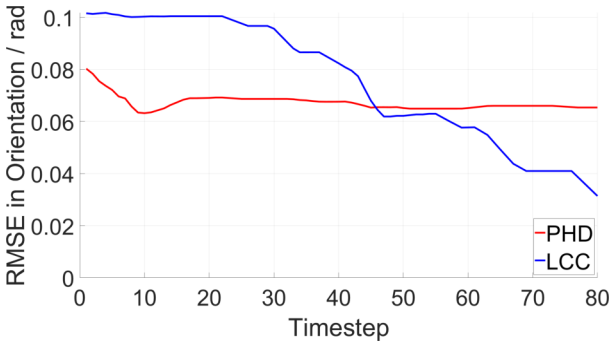

Fig. 4: RMSE of camera orientation estimates for case 1

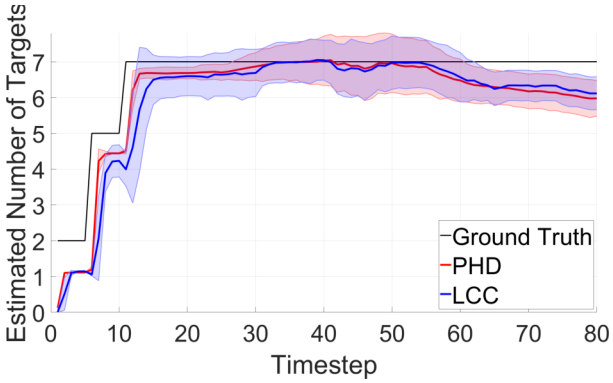

Fig. 5: Estimated target cardinality for case 1

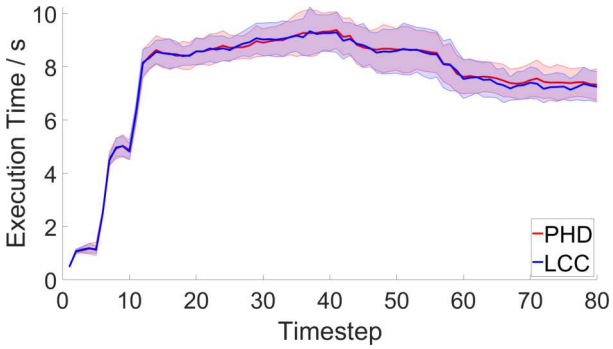

Fig. 6: Execution time per time step for case 1

Case 2 : Similar statements to those made in case 1 can be made for this scenario. Although with the clutter rate increased, there is a noticeable difference in performance of the two filters, as seen in Figs.7–8, the LCC filter is able to maintain a more accurate estimates of the sensor state. Again both filters show similar performance in estimating the target cardinality and show almost identical execution time.

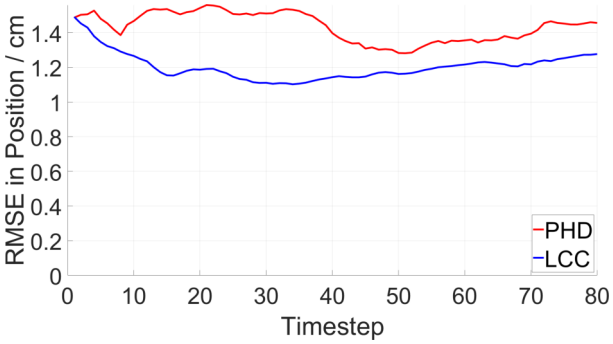

Fig. 7: RMSE of camera position estimates for case 2

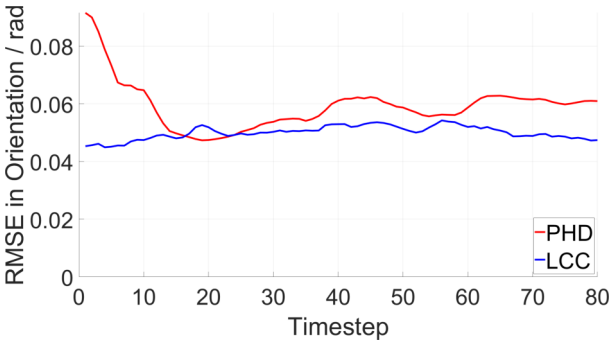

Fig. 8: RMSE of camera orientation estimates for case 2

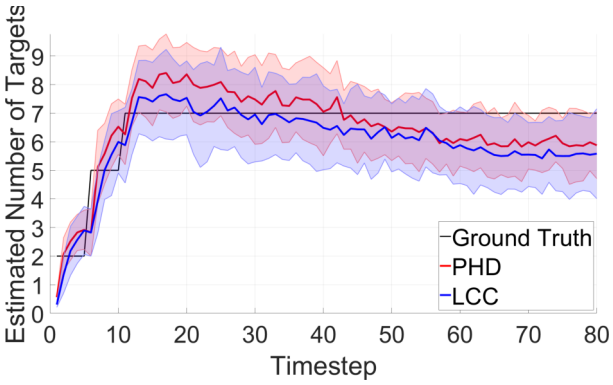

Fig. 9: Estimated target cardinality for case 2

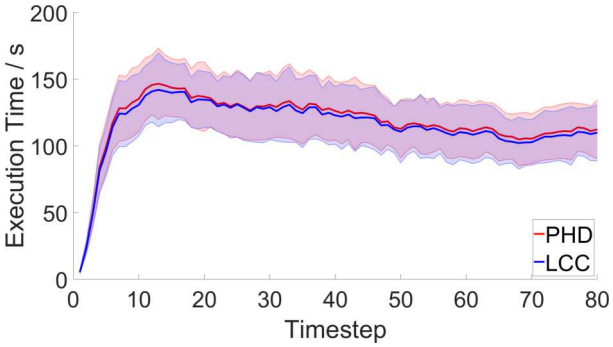

Fig. 10: Execution time per time step for case 2

VI. Conclusions

An improvement to existing an camera calibration method has been presented. This was done by integrating the Linear-Complexity Cumulant (LCC) filter [19] into a Bayesian multi-object tracking and sensor calibration technique [6]. This allows first and second order information to be propagated in close to identical computational complexity when compared to an existing PHD method. The results presented here show that the LCC implementation outperforms the traditional PHD approach in terms of accuracy.

References

[1] F. Chaumette and S. Hutchinson. "Visual servo control. II. Advanced approaches [Tutorial]". In: *IEEE Robotics Automation Magazine* 14.1 (2007), pp. 109–118. ISSN: 1070-9932. DOI: 10.1109/MRA.2007.339609.

[2] R. Mur-Artal and J. D. Tardós. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262. ISSN: 1552-3098. DOI: 10.1109/TRO.2017.2705103.

[3] S.-W. Shih and J. Liu. "A novel approach to 3-D gaze tracking using stereo cameras". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34.1 (2004), pp. 234–245. ISSN: 1083-4419. DOI: 10.1109/TSMCB.2003.811128.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[5] H.S.M. Coxeter. *Projective Geometry*. New York, NY, USA: Springer, 2003.

[6] J. Houssineau et al. "A unified approach for multi-object triangulation, tracking and camera calibration". In: *Signal Processing, IEEE Transactions on* (2016). to appear. (arXiv:1410.2535v1).

[7] J. Houssineau, S. Ivekovic, and D. E. Clark. "Disparity space: A parameterisation for Bayesian triangulation from multiple cameras". In: *Information Fusion, Proceedings of the 15th International Conference on*. July 2012, pp. 1734–1740.

[8] A. Swain and D. Clark. "Extended object filtering using spatial independent cluster processes". In: *2010 13th International Conference on Information Fusion*. 2010, pp. 1–8. DOI: 10.1109/ICIF.2010.5711886.

[9] A. Swain and D. E. Clark. *First-moment filters for spatial independent cluster processes*. 2010. DOI: 10.1117/12.850034. URL: https://doi.org/10.1117/12.850034.

[10] A. Swain and D. Clark. "The PHD filter for extended target tracking with estimable extent shape parameters of varying size". In: *2012 15th International Conference on Information Fusion*. 2012, pp. 1111–1118.

[11] A. Swain and D. E. Clark. "The single-group PHD filter: An analytic solution". In: *Information Fusion, Proceedings of the 14th International Conference on*. 2011, pp. 1–8.

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

[13] O. Hagen et al. "Joint Estimation of Telescope Drift and Space Object Tracking". In: *2016 IEEE Aerospace Conference*. to appear. 2016.

[14] M. Campbell et al. "Image Registration Using Single Cluster PHD Methods". In: *Advanced Maui Optical and Space Surveillance Technologies Conference*. 2017.

[15] Isabel Schlangen et al. "Marker-Less Stage Drift Correction in Super-Resolution Microscopy Using the Single-Cluster PHD Filter". In: *IEEE Journal of Selected Topics in Signal Processing* 10 (Jan. 2015), pp. 1–1. DOI: 10.1109/JSTSP.2015.2506402.

[16] M. Üney, B. Mulgrew, and D. E. Clark. "A Cooperative Approach to Sensor Localisation in Distributed Fusion Networks". In: *IEEE Transactions on Signal Processing* 64.5 (2016), pp. 1187–1199. ISSN: 1053-587X. DOI: 10.1109/TSP.2015.2493981.

[17] R. P. S. Mahler. "Multitarget Bayes Filtering via First-Order Multitarget Moments". In: *Aerospace and Electronic Systems, IEEE Transactions on* 39.4 (2003), pp. 1152–1178.

[18] B. Ristic, D. E. Clark, and N. Gordon. "Calibration of multi-target tracking algorithms using non-cooperative targets". In: *Selected Topics in Signal Processing, IEEE Journal of* 7.3 (2013), pp. 390–398.

[19] D. E. Clark and F. De Melo. "A Linear-Complexity Second-Order Multi-Object Filter via Factorial Cumulants". In: *2018 21st International Conference on Information Fusion (FUSION)*. 2018. DOI: 10.23919/ICIF.2018.8455331.

[20] D.E. Clark and M. Campbell. "Joint Multi-Target Tracking and Parameter Estimation with the Second-Order Factorial Cumulant Filter". In: *22nd International Conference on Information Fusion (FUSION)*. 2019.

[21] G. Sibley, L. Matthies, and G. Sukhatme. "Bias Reduction and Filter Convergence for Long Range Stereo". In: *Robotics Research*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 285–294. ISBN: 978-3-540-48113-3.

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