Distributed Object Tracking based on Cubature Kalman Filter

Venkata Pathuri Bhuvana*†, Melanie Schranz*, Mario Huemer‡ and Bernhard Rinner*
*Alpen Adria University, Institute of Networked and Embedded Systems, Klagenfurt, Austria.
†University of Genova, Department of Marine Engineering, electrical, electronics, and telecommunications, Italy.
‡Johannes Kepler University, Institute of Signal Processing, Linz, Austria
{venkata.pathuri, melanie.schranz, bernhard.rinner }@aau.at, mario.huemer@jku.at

***Abstract*—In this work, we propose the cubature Kalman filter (CKF) based distributed object tracking algorithm in a visual sensor network (VSN). A VSN consists of several distributed smart cameras having the ability to process and analyze the retrieved data locally. The first objective is to optimize the tracking process within the VSN through the CKF. Under the conditions of non-linear motion and observation model, the CKF based method features a considerably better tracking accuracy than the extended Kalman filter (EKF) based method in terms of the mean square error (MSE). Although, the particle filter (PF) based method shows better performance than the CKF, it is computationally very complex. The second objective is to optimize the object tracking by aggregating the tracking results from multiple cameras. Assuming the VSN is a multi-camera network with overlapping field of views (FOVs), cameras having the same object in their FOV exchange their local estimates of the object's position and velocity. During the estimation process, each of the participating cameras aggregates the received states via a consensus algorithm. Thus, the object's real state is more accurately predicted by the resulting joint state than it would be by processing only a single camera's observation.**

I. Introduction and Motivation

Distributed estimation of an object's state is an extensively studied topic in the field of object tracking in visual sensor networks (VSN). A VSN is a camera network of spatially distributed smart cameras without a central processing unit. The individual cameras autonomously process and analyze the retrieved data locally. To achieve a decision within the network, e.g. on the object's state, the cameras work cooperatively. In such a distributed approach, in a first step, the tracking is done locally on each camera having a specific object in its field of view (FOV). Secondly, the preprocessed data is exchanged by the cameras having the same object in their FOV. Thus, the local estimated states are aggregated through a consensus algorithm to a joint state as proposed by [1]. In [2], the authors integrated the consensus algorithm to a distributed Kalman filtering approach forming the Kalman Consensus Filter (KCF). However, the KCF in [2] assumes a linear state model for the tracked objects.

However, the movement of real objects exhibits non-linear behavior. In such cases, the extended Kalman filter (EKF) based object tracking can be used, as in [3]. A drawback of the EKF is its limited accuracy due to the inherent linearization errors. More efficient non-linear filters, such as the particle filter (PF) [4] can be used for distributed object tracking, but PFs generally feature a considerably higher computational complexity compared to Kalman filters. Recently, the cubature Kalman filter (CKF) [5] has been developed as an computationally efficient solution for non-linear state estimation.

The objective of this paper is to propose a distributed CKF based object tracking approach in a VSN with overlapping FOVs. Each camera can detect and thus, track objects in their FOV. Since we have calibrated cameras, each camera is able to measure the coordinates of the object's position on the image plane. The CKF state estimator runs locally on each camera having an object in its FOV to estimate the object's position on the ground plane. Then, the cameras exchange the locally estimated object's states among themselves. In a final step, the participating cameras aggregate the received states with the local state by applying a consensus algorithm. Hence, at the end of consensus, all the cameras in the VSN are in knowledge of the resulting global state of all objects in the network.

II. System Model

In this work, we consider a VSN consisting of a fixed set of calibrated smart cameras c_i where $i = 1, 2, \cdots, C$ with overlapping FOVs as illustrated in Figure 1. The smart cameras have the ability to analyze and process the retrieved data locally [6]. In addition, the object's trajectory can be fully

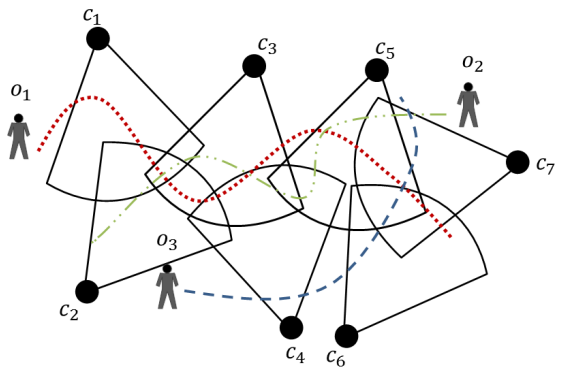

Fig. 1. Visual sensor network consisting of spatially distributed smart cameras.

observed during its movement through the given network. The task of the VSN is to monitor the given environment and to identify and track a specific object o_k where $k = 1, 2 \cdots, K$. This is achieved by a distributed tracking algorithm performed by each of the cameras c_i in the network. As these cameras are calibrated, there exists a homography to calculate the object's position on the ground plane. The re-identification of

an object at any camera is a typical feature of the distributed tracking. Re-identification can be achieved with the help of the position on the ground plane in case of overlapping FOVs, or appearance features calculated already in the tracking process. Here, We assumed perfect object re-identification.

The state of an object o_k comprises of its position and the velocity on the ground plane. Thus, the state at time t is described as $\mathbf{x}_t^{ik} = [x\ y\ \dot{x}\ \dot{y}]$ where k and i represents the identity of the object k and the camera i, respectively. The state transition or motion model is

$$\mathbf{x}_{t+1}^{ik} = \begin{bmatrix} x_t^{ik} + \delta\dot{x}_t^{ik} + \frac{\delta^2}{2}\ddot{x} \\ y_t^{ik} + \delta\dot{y}_t^{ik} + \frac{\delta^2}{2}\ddot{y} \\ \dot{x}_t^{ik} + \delta\ddot{x} \\ \dot{x}^{ik}{}_t + \delta\ddot{y} \end{bmatrix}, \tag{1}$$

where $\ddot{x}$ and $\ddot{y}$ are the acceleration of the object in x and y directions which are modeled by the independent and identically distributed (IID) white Gaussian noise vector with covariance $\mathbf{Q}_t^{ik}$. δ is the time interval between the two measurements. Hence, the motion of the objects is modeled by a constant velocity with Gaussian distributed acceleration. The state of the object is generally estimated from a set of the measurements taken at each time step t. The system's measurement equation is

$$\mathbf{y}_t^{ik} = \mathbf{h}_t^{ik}\left(\mathbf{x}_t^{ik}\right) + \mathbf{v}_t^{ik}, \tag{2}$$

where $\mathbf{v}_t^{ik}$ is an IID measurement noise vector with covariance $\mathbf{R}_t^{ik}$. The measurement function $\mathbf{h}_t^{ik}$ is the non-linear homography function which converts the object's coordinates from the image to the ground plane.

A. Sequential Non-Linear Bayesian Estimation

The Bayesian minimum mean square error method estimates (MMSE) the state $\mathbf{x}_t^{ik}$ as the expectation of the posterior probability density function (PDF) $p\left(\mathbf{x}_t^{ik} \mid \mathbf{y}_{1:t}^{ik}\right)$ of $\mathbf{x}_t^{ik}$, given all the measurements from time 1 to t [7]. Under this assumption, the sequential Bayesian estimation has two primary steps at every time instance t:

- State prediction: Predict the state at t from the posterior PDF at $t-1$.

$$p\left(\mathbf{x}_t^{ik} \mid \mathbf{y}_{1:t-1}^{ik}\right) = \int p\left(\mathbf{x}_{t-1}^{ik} \mid \mathbf{y}_{1:t-1}^{ik}\right) p\left(\mathbf{x}_t^{ik} \mid \mathbf{x}_{t-1}^{ik}\right) d\mathbf{x}_{t-1}^{ik} \tag{3}$$

- Measurement update: Upon receiving the measurement, the predicted state is updated.

$$p\left(\mathbf{x}_t^{ik} \mid \mathbf{y}_{1:t}^{ik}\right) = \frac{p\left(\mathbf{x}_t^{ik} \mid \mathbf{y}_{1:t-1}^{ik}\right) p\left(\mathbf{y}_t^{ik} \mid \mathbf{x}_t^{ik}\right)}{\int p\left(\mathbf{x}_t^{ik} \mid \mathbf{y}_{1:t-1}^{ik}\right) p\left(\mathbf{y}_t^{ik} \mid \mathbf{x}_t^{ik}\right) d\mathbf{x}_t^{ik}}. \tag{4}$$

Under the assumptions of linearity and additive Gaussian noise, the integrals in (3) and (4) can be solved analytically and result in the Kalman filter (KF). In contrast, if the system model is non-linear as in the case of camera tracking then these integrals are intractable. Often, non-linear state estimation methods such as the PF and CKF are used to approximate these integrals.

III. Proposed Method

Several non-linear state estimation methods such as the EKF, and PF can be used to improve the tracking process by estimating the object's state. Our intention is to achieve a better non-linear state estimation in terms of state accuracy and complexity. Hence, we propose to integrate the CKF for distributed object tracking in a VSN. Due to the adverse camera effects, further explained in Section III-B, a consensus algorithm can be used to increase the reliability of the object's state. Thus, the state of each object o_k is estimated based on the CKF of all cameras c_i observing the object o_k. The local estimation is exchanged between the cameras. Then a consensus algorithm is used to aggregate the local states to the global state. All other cameras in the VSN get the information about the reached consensus on the object o_k.

A. Cubature Kalman Filter (CKF)

The CKF provides an approximation to the Bayesian filter. Under the assumption that the process and the measurement equations are non-linear with additive Gaussian noise, the integrals in (3) and (4) become multi-dimensional Gaussian-weighted integrals of the form

$$I\left(f\right) = \int_{R^n} f\left(\mathbf{x}\right) \exp\left(-\mathbf{x}^T\mathbf{x}\right) d\mathbf{x}. \tag{5}$$

This type of integrals does not have a closed form solution. On the other hand, there are several numerical approximation methods such as monte-carlo approximations (PF), grid and lattice based methods, and unscented transformations (UKF) are available to solve them. The efficiency of a non-linear filter highly depends on the efficiency of the approximation method it uses.

The CKF uses third degree cubature rule to approximate the integral of form (5). This rule uses the spherical-radial transformation to change the variables from the Cartesian coordinate system to the Radial and Spherical coordinate system as: $\mathbf{x} = r\mathbf{z}$ with $\mathbf{z}^T\mathbf{z} = 1$ such that $\mathbf{x}^T\mathbf{x} = r^2$ for $r \in [0, \infty)$. Then, the integral (5) becomes

$$I\left(f\right) = \int_0^\infty \int_{U_n} f\left(r\mathbf{z}\right) r^{n-1} \exp\left(-r^2\right) d\sigma\left(\mathbf{z}\right) dr, \tag{6}$$

where U_n is the surface of the n-dimensional unit sphere and $d\sigma\left(.\right)$ is the unit spherical surface measure. This form of integrals can be numerically approximated by the spherical cubature rule as given by

$$I\left(f\right) \approx \frac{\sqrt{\pi^n}}{2n} \sum_{i=1}^{2n} f\left(\sqrt{\frac{n}{2}}\xi_i\right), \tag{7}$$

where n is the dimension of the vector $\mathbf{x}$, ξ_i is the i-th cubature point located at the intersection of the surface of the n-dimensional unit sphere and its axes, as shown in Figure 2.

This rule can be extended to solve the prediction and the posterior PDFs that are in the form of standard Gaussian with a mean μ and variance Σ.

$$\int_{R^n} f\left(\mathbf{x}\right) \mathcal{N}\left(\mathbf{x}; \mu, \Sigma\right) d\mathbf{x} \approx \frac{1}{2n} \sum_{i=1}^{2n} f\left(\sqrt{\Sigma}\xi_i + \mu\right). \tag{8}$$

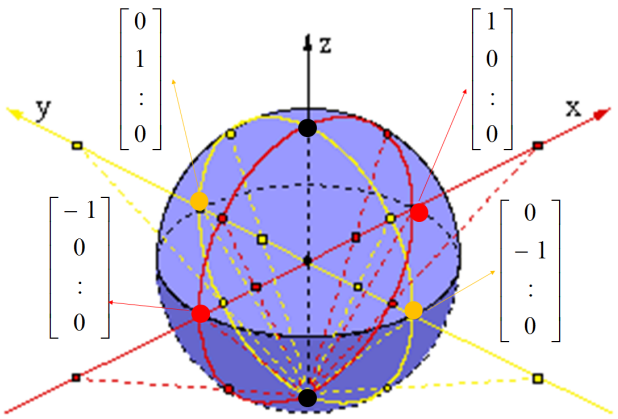

Fig. 2. Cubature points.

The simplified CKF algorithm is realized by the following steps:
To reduce the complexity in notation, $\mathbf{x}_t^{ik}$ is represented as $\mathbf{x}_t$. As a first step, initialize the error covariance $\mathbf{P}_{0|0}$, and the prior state $\mathbf{x}_{0|0}$. Under the assumptions that the state-space model is non-linear with additive Gaussian noise, the posterior PDF at $t-1$ is known and given by

$$p\left(\mathbf{x}_{t-1} \mid \mathbf{y}_{t-1}\right) \sim \mathcal{N}\left(\widehat{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}\right),$$

where $\widehat{\mathbf{x}}_{t-1|t-1}$ and $\mathbf{P}_{t-1|t-1}$ are the estimated state and covariance, respectively at time $t-1$ which can be used to predict the state at time t as described in (3). The predicted state is calculated as the expectation of $p\left(\mathbf{x}_t \mid \mathbf{y}_{t-1}\right)$ which is given by

$$\begin{aligned}\widehat{\mathbf{x}}_{t|t-1} &= E\left[p\left(\mathbf{x}_t \mid \mathbf{y}_{t-1}\right)\right] \\ &= \int p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) \mathcal{N}\left(\widehat{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}\right) d\mathbf{x}_{t-1}\end{aligned}$$

where $p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$ can be calculated from (1). To predict the state using the CKF, the square root of the error covariance $\mathbf{S}_{t-1|t-1}$ is calculated in such a way that

$$\mathbf{P}_{t-1|t-1} = \mathbf{S}_{t-1|t-1}\mathbf{S}_{t-1|t-1}^T.$$

In a further step, the cubature points $i = (1, 2, ..., 2n)$ are calculated as shown in (8) using $\widehat{\mathbf{x}}_{t-1|t-1}$ and $\mathbf{P}_{t-1|t-1}$ as

$$\mathbf{c}_{i,t-1|t-1} = \mathbf{S}_{t-1|t-1}\xi_i + \widehat{\mathbf{x}}_{t-1|t-1},$$

where ξ_i represents intersection points of the surface of the n-dimensional unit sphere and its n axes. These cubature points are propagated through the state transition function as

$$\mathbf{x}_{i,t|t-1}^* = f_t\left(\mathbf{c}_{i,t-1|t-1}\right).$$

The predicted state can be calculated by the weighted sum of the non-linearly transformed cubature points as

$$\widehat{\mathbf{x}}_{t|t-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{x}_{i,t|t-1}^*,$$

where the equal weight $\frac{1}{2n}$ is assumed to each cubature point. In the last step of the state prediction process, the error covariance is estimated by

$$\begin{aligned}\mathbf{P}_{t|t-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{x}_{i,t|t-1}^{*T}\mathbf{x}_{i,t|t-1}^* \\ - \widehat{\mathbf{x}}_{t|t-1}\widehat{\mathbf{x}}_{t|t-1}^T + \mathbf{Q}_t.\end{aligned}$$

The second part of the CKF is to correct the state prediction by using the measurement at time t. In the case of a non-linear measurement equation with additive Gaussian noise the predicted PDF is known as

$$p\left(\mathbf{x}_t \mid \mathbf{y}_{t-1}\right) \sim \mathcal{N}\left(\widehat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}\right).$$

In order to calculate the predicted measurement, the square root of the predicted error covariance $\mathbf{S}_{t|t-1}$ is calculated in such a way that

$$\mathbf{P}_{t|t-1} = \mathbf{S}_{t|t-1}\mathbf{S}_{t|t-1}^T.$$

Similar to the state prediction, the cubature points are calculated using the predicted state $\widehat{\mathbf{x}}_{t|t-1}$ and the square root of the predicted error covariance $\mathbf{S}_{t|t-1}$ as

$$\mathbf{c}_{i,t|t-1} = \mathbf{S}_{t|t-1}\xi_i + \widehat{\mathbf{x}}_{t|t-1},$$

where again ξ_i represents intersection points of the surface of the n-dimensional unit sphere and its axes. These cubature points are propagated through the measurement function as

$$\mathbf{y}_{i,t|t-1} = \mathbf{h}_t\left(\mathbf{c}_{i,t|t-1}\right).$$

The predicted measurement is derived by calculating the weighted sum of the transformed measurement cubature points as

$$\widehat{\mathbf{y}}_{t|t-1} = \frac{1}{2n}\sum_{i=1}^{2n}\boldsymbol{y}_{i,t|t-1}.$$

From this step onwards, the CKF follows the similar approach as the KF. The innovation covariance and the cross covariance matrices are calculated as

$$\begin{aligned}\mathbf{P}_{\mathbf{yy},t|t-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{y}_{i,t|t-1}\mathbf{y}_{i,t|t-1}^T \\ - \widehat{\mathbf{y}}_{t|t-1}\widehat{\mathbf{y}}_{t|t-1}^T + \mathbf{R}_t,\end{aligned}$$

$$\begin{aligned}\mathbf{P}_{\mathbf{xy},t|t-1} = \frac{1}{2n}\sum_{i=1}^{2n}\mathbf{x}_{i,t|t-1}^*\mathbf{y}_{i,t|t-1}^T \\ - \widehat{\mathbf{x}}_{t|t-1}\widehat{\mathbf{y}}_{t|t-1}^T.\end{aligned}$$

The Kalman gain is calculated by using the innovation covariance $\mathbf{P}_{\mathbf{yy},t|t-1}$ and the cross covariance $\mathbf{P}_{\mathbf{xy},t|t-1}$ matrices as

$$\mathbf{k}_t = \mathbf{P}_{\mathbf{xy},t|t-1}\mathbf{P}_{\mathbf{yy},t|t-1}^{-1}.$$

Finally, the predicted state can be updated using the received measurements , predicted measurements, and the Kalman gain.

$$\widehat{\mathbf{x}}_{t|t} = \widehat{\mathbf{x}}_{t|t-1} + \mathbf{k}_t\left(\mathbf{y}_t - \widehat{\mathbf{y}}_{t|t-1}\right).$$

At the end of the CKF algorithm, the error covariance is calculated in order to be reused in the next iteration step:

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{k}_t\mathbf{P}_{\mathbf{yy},t|t-1}\mathbf{k}_t^T.$$

Hence, the posterior PDF at t becomes

$$p\left(\mathbf{x}_t \mid \mathbf{y}_t\right) \approx \mathcal{N}\left(\widehat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}\right).$$

The CKF is used by each camera i where $i = 1, 2 \cdots, N$ to estimate the state of all objects $\mathbf{x}_t^{ik^i}$ in its FOV k^i where $k^i \subset k = 1, 2 \cdots, K$. Then, the cameras exchange the locally estimated states between themselves and run the average consensus algorithm to aggregate the global state.

B. Average Consensus Algorithm

The consensus describes an agreement over the state $\mathbf{x}_t^{ik}$ from cameras having the same object o_k in their FOV. Thus, they form a neighborhood, defined with

$$\mathcal{N}_i = \{j \in V : a_{ij} \neq 0\}, \tag{9}$$

where $V = 1, \ldots, N$ describes the vertices of the dynamic graph $G(t) = (V(t), E(t))$ together with $E(t) \subseteq V \times V$ as the edges. $\mathbf{A} = \{a_{ij}\}$ denotes the adjacency matrix of $G(t)$, describing the fact that camera c_i and camera c_j have overlapping FOVs. The number of neighbors of each camera c_i is formed by the degree of $G(t)$ with $d_i = |\mathcal{N}_i|$.

Forming a consensus for all neighbors is necessary to reduce deviations from the tracking output of the individual cameras c_i. These deviations result from different adverse effects, such as distance or orientation of the camera to the object or occlusions from static or moving obstacles.

Applying the consensus algorithm, the global state of an object o_k at time t at each camera c_i is then updated with

$$\mathbf{x}_t^k = \mathbf{x}_t^{ik} + \sum_{j=1}^{N} a_{ij}(\mathbf{x}_t^{jk} - \mathbf{x}_t^{ik}), \tag{10}$$

where $a_{ij} \in \{0, 1\}$ stands for the weights of the communication link [8]. In our approach the weights are identical to the entries of the adjacency matrix $\mathbf{A}$. Theoretically, a consensus is achieved by all neighbors if the information state difference converges to 0 when time moves to infinity

$$i, j = 1, \ldots N, |\mathbf{x}_t^{jk} - \mathbf{x}_t^{ik}| \to 0, \text{as } t \to \infty \tag{11}$$

Practically, it is reached up to a predefined threshold within a given number of iterations. Comparing the individual states to the joint state, the latter is much more optimized in terms of closeness to the real object's state.

IV. Simulation Results

In this section, we discuss the efficiency and complexity of the CKF based distributed object tracking in comparison with the EKF and PF based distributed object tracking. The overall efficiency of the proposed distributed object tracking depends both on the efficiency of the local tracking algorithm employed at each camera and the consensus algorithm used to aggregate the locally estimated states. The first part of this section discusses the efficiency and complexity of the local object tracker at each camera. Although, we do not consider different consensus algorithms, the second part discusses the efficiency of distributed object tracking based on the proposed consensus algorithm III-B with different local object tracking methods such as the CKF, PF, and EKF.

The overall simulation considers a VSN with six cameras having an overlapping FOV. All of the cameras can partially observe a xy-plane where, $x \in [0, 500]$ and $y \in [0, 500]$. The motion of the objects is modeled by a constant velocity with Gaussian distributed acceleration. The high variance of the acceleration can model the uncertainties of the object motion in real time. In the simulation, the considered process noise covariance is Q=diag$\{1, 1\}$. Each camera i has its own homography function h^i. Since we assume the fixed cameras, the homography of a camera does not change with time t and object k. The uncertainties in the homography are modeled by the additive Gaussian noise with variance R^i.

The simulation results show that the CKF and PF based object tracking features significant improvement over the EKF based method. Figure 3 illustrates a time snapshot of the estimated x and y coordinates for a single object for the CKF, EKF, and PF based tracking methods compared to the ground truth. The EKF and CKF assumes perfect covariance knowledge. The sequential important resampling PF (SIRPF) with number of particles N=1000 is used in the PF based object tracking method. In addition, this figure also shows the corresponding errors. Although, the PF based method achieves a slight improvement over the CKF based method in terms of error, it is computationally more complex.

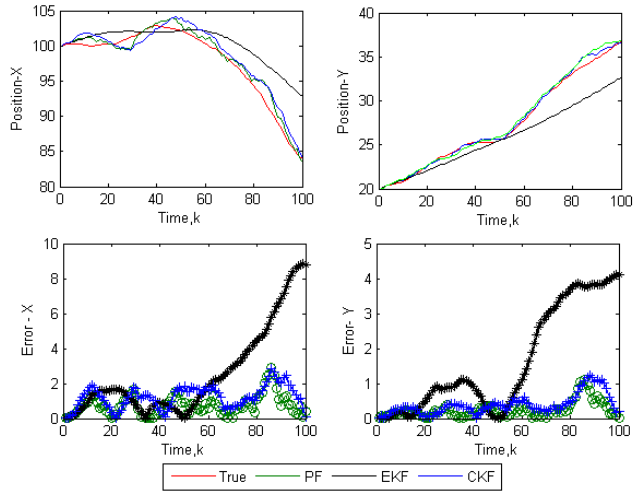

Fig. 3. Comparison of the tracking accuracy of the object tracking algorithms based on the EKF, PF, and CKF.

Table I compares the average MSE of the three methods and their complexity. The avergae MSE is averaged over thousand simulation runs as each simulation run tracks the object for 100 time steps. The complexity is illustrated in terms of the execution time for the single simulation run (number of particles for the PF is considered as 1000). This table shows that on the considered platform, the PF is 50 times more complex than the CKF. At the same time, the EKF is significantly less complex than the CKF, but the CKF features a considerably higher accuracy compared to the EKF.

In the second scenario, the efficiency of the three filters is compared in the distributed object tracking proposed in III-B. Each of the six cameras tracks all three objects in their FOV locally and then run the consensus algorithm. In the simulation run, each object is tracked by at least one camera and at

TABLE I. ACCURACY AND COMPLEXITY OF THE OBJECT TRACKING METHODS BASED ON THE EKF, PF, AND CKF.

Method	MSE	Execution time(sec)
EKF	4.39	0.0114
CKF	2.28	0.0899
PF (N=1000)	2.32	4.3534

most three cameras. The number of cameras that can track an object is based on the position and the motion of the object. The initial state and the noise conditions are assumed to be known for each camera. If an object enters camera's FOV for the first time at t, the aggregated global state at $t-1$ is considered as the initial state for the local object tracker. The communication links between the cameras with an overlapping FOV are assumed to be perfect.

Figure 4 illustrates overall efficiency of the distributed tracking methods based on the EKF, CKF, and PF. This figure shows the MSE of the position of all objects for 30 simulation runs. Each simulation run tracks the objects for 100 time steps. This figure clearly shows that the PF based distributed object tracking outperforms the CKF and EKF based methods. At the same time, the CKF based method outperforms the EKF based method considerably.

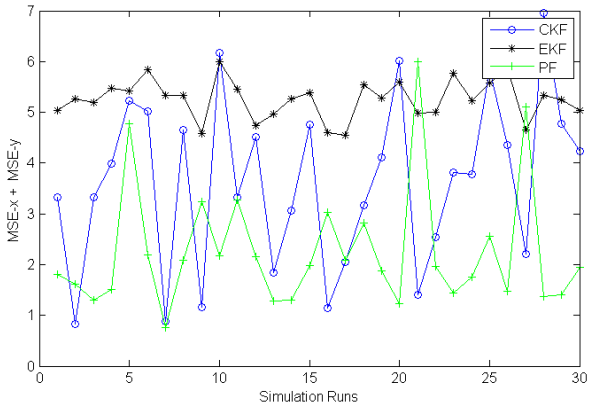

Fig. 4. MSE (both of x and y combined) comparison of the distributed object tracking using the EKF, PF and CKF.

Table II compares the MSE of the distributed object tracking based on the three methods. This table compares the tracking accuracy of a single object when the object is tracked by different number of cameras. Since the accuracy and the complexity are two important factors for the object tracking in a VSN consisting of embedded platforms, the CKF based distributed tracking can be a good trade-off between the EKF and PF.

TABLE II. MSE OF THE DISTRIBUTED OBJECT TRACKING METHODS BASED ON THE EKF, PF, AND CKF FOR DIFFERENT NUMBER OF CAMERAS C.

Method	C=1	C=2	C=3
EKF	4.39	3.97	3.46
CKF	2.89	2.09	1.81
PF (N=1000)	2.28	1.59	1.27

V. CONCLUSION

In this paper, a distributed CKF based object tracking is proposed for a VSN with calibrated smart cameras. Each camera runs a CKF based tracking on all the objects in its FOV. Then all the cameras in the network exchange the locally estimated states among themselves. An average consensus algorithm is used to aggregate the local state to the global state. Hence, at the end of consensus, all the cameras are in knowledge of the resulting global state of all objects in the network. The consensus also improves the tracking efficiency by reducing the deviations caused by different adverse effects such as rapid movements.

In addition, we also compared the efficiency of the CKF based method to that of the EKF and PF based tracking methods with simulation based analysis. Our results show that the CKF based method outperforms the EKF based method in terms of the efficiency. The PF based method always outperforms the CKF and EKF based methods, but it is computationally very complex. Since the accuracy and the complexity are two important factors for the object tracking in a VSN consisting of embedded platforms, the CKF based distributed tracking can be a good trade-off between the EKF and PF.

ACKNOWLEDGMENT

This work was supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA 2010-0012.

REFERENCES

[1] Reza Olfati-Saber and Nils F. Sandell S., *Distributed tracking in sensor networks with limited sensing range*, IEEE American Control Conference, pp. 3157-3162, 2008.

[2] Bi Song and Chong Ding and Kamal, A.T. and Farrell, J.A. and Roy-Chowdhury, A.K., *Distributed Camera Networks*, IEEE Signal Processing Magazine, Vol. 28, pp. 20-31 2011.

[3] Chong Ding and Bi Song and Akshay A. Morye and Jay A. Farrell and Amit K. Roy Chowdhury, *Collaborative Sensing in a Distributed PTZ Camera Network*, IEEE Transactions on Image Processing, Vol. 21, pp. 3282-3295, 2012.

[4] Ondrej Hlinka, Ondrej Slučiak, Franz Hlawatsch, Petar M. Djurić, and Markus Rupp, *Distributed Gaussian particle filtering using likelihood consensus*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3756-3759, 2011.

[5] Ienkaran Arasaratnam and Simon Haykin, *Cubature Kalman Filters*, IEEE Transactions on Automatic Control, vol.54, pp. 1254-1269, 2009.

[6] Bernhard Rinner and Markus Quaritsch and Wolfgang Schriebl and Thomas Winkler and Wayne Wolf, *The Evolution from Single to Pervasive Smart Cameras*, ACM/IEEE International Conference on Distributed Smart Cameras, pp. 1-10, 2008.

[7] Steven M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, Vol.1, 1993.

[8] Wei Ren and Randal W. Beard and Ella M. Atkins, *Distributed Consensus in Multi-Vehicle Cooperative Control*, Springer-Verlag London Limited, 2008.