Sensors & Transducers, Vol. 191, Issue 8, August 2015, pp. 28-39

Sensors & Transducers

http://www.sensorsportal.com

Resource-Aware Dynamic Clustering Utilizing State Estimation in Visual Sensor Networks

Melanie SCHRANZ and Bernhard RINNER
Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
E-mail: melanie.schranz@aau.at, bernhard.rinner@aau.at

Received: 22 July 2015 /Accepted: 24 August 2015 /Published: 31 August 2015

Abstract: Generally, resource-awareness plays a key role in wireless sensor networks due the limited capabilities in processing, storage and communication. In this paper we present a resource-aware cooperative state estimation facilitated by a dynamic cluster-based protocol in a visual sensor network (VSN). The VSN consists of smart cameras, which process and analyze the captured data locally. We apply a state estimation algorithm to improve the tracking results of the cameras. To design a lightweight protocol, the final aggregation of the observations and state estimation are only performed by the cluster head. Our protocol is based on a market-based approach in which the cluster head is elected based on the available resources and a visibility parameter of the object gained by the cluster members. We show in simulations that our approach reduces the costs for state estimation and communication as compared to a fully distributed approach. As resource-awareness is the focus of the clusterbased protocol we can accept a slight degradation of the accuracy on the object's state estimation by a standard deviation of about 1.48 length units to the available ground truth. *Copyright © 2015 IFSA Publishing, S. L.*

Keywords: Visual sensor network, Resource distribution, Wireless communication, Negotiation theory, Decentralized processing.

1. Introduction

In wireless sensor networks (WSNs), cooperative control and distributed processing opened up a wide research field. It is a very popular topic, e.g., in mobile robots, unmanned vehicles, automated highway systems, industrial process or environmental monitoring [1]. WSNs are constituted of spatially distributed sensor nodes to retrieve information from the environment and react on it. The individual sensor nodes in such a network communicate wirelessly and their actions are also autonomous with respect to the received information. Furthermore, the individual sensor nodes in a WSN are able to learn from its environment especially through exchanging locally retrieved information among themselves. A typical characteristic of sensor nodes used in an ad-hoc WSN are the limited resources. They are usually battery powered, have a bounded communication range and limited onboard processing and storage capabilities.

Throughout this paper, we consider only networks consisting of visual sensors communicating wirelessly. Visual sensor networks (VSNs) consist of autonomous low-power image sensors with storage and communication capabilities as well as a processing unit on board [2]. Thus, they have the ability to analyze and process the data locally. A typical task of a VSN is to identify and track objects for surveillance and identification applications.

The object is usually described by a state, including the position, the velocity or other characteristics of the object. A VSN with overlapping field of views (FOVs) and thus multiple observations of the same target simultaneously asks for aggregating them to a joint, improved observation. To compute a global state, the individual observations are exchanged among the cameras and aggregated locally. These aggregated observations serve as input to state estimation algorithms. A typical approach for state estimation in a VSN is to forward the observations to a central unit. This unit aggregates the observations and performs a global state estimation algorithm. Another possibility in state estimation is to use a fully distributed approach. Each camera exchanges its observations with the other cameras in the VSN and performs global state estimation. In this paper we propose a lightweight resource-aware cluster-based protocol. In a modified market-based approach—proposed by [3]—we elect a cluster head responsible for state estimation incorporating the observations from its cluster members. The role of the cluster head is handed over to a cluster member if the available local resources decrease. Comparing the cluster-based to the centralized approach, scalability is increased and there is no longer the risk of a single point of failure. Further, in resource-aware VSNs the fully distributed approach stresses the camera's capabilities in communication and processing due to the high amount of messages to be exchanged. Further, their observations are processed simultaneously and thus leading to redundant results on each camera.

This paper provides two scientific contributions with a focus on resource-awareness: (i) Utilized by the market-based approach we perform dynamic cluster management including cluster head election and handover. (ii) With evaluations in a simulation environment we show the advantages in terms of resource awareness of the proposed cluster-based protocol over the fully distributed approach. A state estimation algorithm is incorporated for both approaches.

The paper is organized as follows: Section 2 presents the related work to state estimation and clustering methods in VSNs. In Section 3 we define the system model for the cluster-based approach. Further, Section 4 describes the underlying market-based approach as well as the state-space model. Section 5 shows the cluster-based protocol and the incorporation of the state estimation algorithm. In Section 6 we evaluate the proposed protocol and discuss the simulation results. Finally, Section 7 concludes the paper and gives an outlook on future work.

2. Related Work

In our approach we focus on related work concerning cooperative state estimation and clustering methods in VSNs. Cooperative state estimation is a well-known research topic in VSNs to optimize an object state. There already exist approaches for fully distributed systems having an underlying linear state-space model [4]. Several authors in [5], [6] and [7] propose the distributed Kalman-Consensus Filter (KCF) for distributed state estimation in camera networks. For a non-linear state space model, there exist other filters like the Extended Kalman Filter, the Particle Filter or the newly approached Cubature Kalman Filter. In [8] we made a comparison between these three filters for distributed state estimation in VSNs. The best trade-off in terms of computational complexity and estimation accuracy when modeling non-linear states is achieved with the Cubature Kalman Filter.

Nevertheless, in a VSN the limited resources of the cameras need to be managed accordingly. One approach to reduce the participating nodes and thus save resources is clustering. The literature describes two main strategies for clustering: (i) In a static cluster the nodes are assigned offline to a specific cluster and do not change over the network's lifetime [9], [10]. (ii) In a dynamic approach clustering is triggered by arising events in the network as in [11], [12], [13] and [14]. In [13] and [14] they use the term *grouping* instead of *clustering*. Nevertheless, their task is to form clusters having a qualifying parameter. In [14] this qualifying parameter describes the extrinsic parameters of a PTZ-camera to examine the cameras coverage over the object of interest. Thus, they focus on distribute tracking performance among the cameras. Further, in [11] and [14] it is necessary to exchange various messages among the cluster members, e.g. to log-in/logoff from the cluster. Also the coverage problem plays a role in VSNs for air space surveillance as in [15] and [16], although the clustering process is directed via a central unit. Especially for resource management of the nodes in VSNs there are several ideas: In [17] they propose a handoff algorithm with adaptive resource management that automatically and dynamically allocates resources to objects with different priority ranks. Their resource management approach is to decrease the frame rate. Similarly it is done in [18], focusing on coverage as well. In [19] a cluster head selects cluster members to deliver tracking responsibilities. Further, in [20] they propose HEED (hybrid, energy-efficient distributed clustering approach) for sensor networks. They select the cluster head based on the residual energy of the node as well as neighbor proximity. Nevertheless, the termination of the clustering approach is dependent on the number of neighbors. A similar approach is realized in [21]. Nevertheless, the communication overhead produced by this clustering protocol is quite high and its usage for battery-powered devices questionable.

In contrast to the existing research directions, our objective is to establish a resource-aware approach for smart cameras in VSNs. We adapt a market based approach proposed in [3] to design a dynamic cluster-based protocol focusing on available resources and a visibility parameter in order to elect a single camera for state estimation. Contrary to [6], we reduce the overhead for communication by designing a lightweight cluster-based protocol with a minimal number of messages to be exchanged and thus, spare a node's resources. Additionally, to the work in [22], this article covers the relaxation of the assumption on object re-identification.

3. System Model

In this paper we consider a VSN of a fixed set of calibrated smart cameras $c_i \in C$ as illustrated in Fig. 1. The task of the VSN is to monitor the given environment and thus to identify and track one or more specific objects $o_k \in O$. We assume a perfect object re-identification. Thus, each $c_i \in C$ is aware of the object's global identifier. As these cameras are calibrated, they are able to calculate the object's position on the ground plane by applying a homography on the object's image plane coordinates. The object position is referred to as observation. Since the cameras in Fig. 1 have overlapping FOVs, they have the ability to track a specific object ok simultaneously. This enables cooperative work in the VSN. Cooperation is achieved by exchanging their individual observations and processing them accordingly.

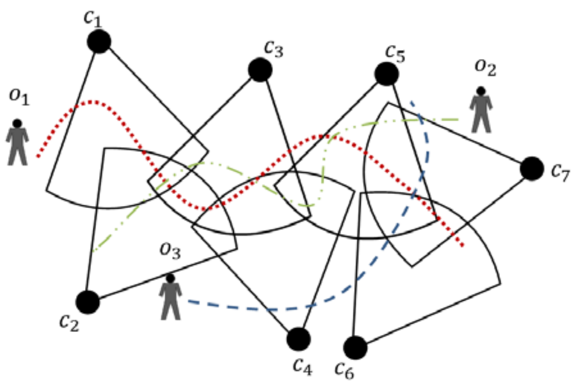


Fig. 1. VSN with spatially distributed smart cameras performing multiple object tracking.

The objective of this paper is to present a resourceaware protocol for cooperative state estimation in a VSN by forming dynamic clusters—one per object in the scene. A cluster is a subset of all cameras in the network $C^k \subset C$, whereby a camera c_i^k is a cluster member of the cluster C^k, if the camera has the object ok in its FOV. Thus, this cluster is given as

$$C^k := \{c_i^k \in C | c_i \in C \wedge o_k \text{ in FOV}\} \quad (1)$$

and c_h^k represents the cluster head of C^k.

The dynamics of the clustering is illustrated in Fig. 2. The individual figures show subsequent time slots of a specific cluster C^k. The camera marked with an x is the cluster head c_h^k with the responsibility to estimate the state for the specific object o_k. All cameras with gray colored FOVs indicate the presence of the object in their FOV. Cameras without an x, but with a colored FOV, denote the cluster members.

4. Resource-Aware State Estimation

In our approach the cluster head is responsible for the collection of the object's observations from all cluster members and to perform cooperative state estimation on them. For its election we propose a market-based approach based on the work of [3]. In this approach the tracking responsibility for a specific object is autonomously distributed among the cameras in the network. This market-based approach is used to elect a single camera as cluster head out of all cluster members. Contrary to the approach in [3], all cluster members continuously track the objects in their FOV.

4.1. Market-based Dynamic Clustering

Within the market-based approach we have two different interacting components: a camera owning the object and cameras bidding for the object. In our case the owner is the cluster head $c_h^k \in C^k$ with the responsibility for auction initiation. The bidders are the cluster members $c_i^k \in C^k$ and have the task to bid for an object.

The primary step of the market-based approach is to initiate an auction by the owner for a specific object o_k. The auction initiation is necessary to elect the owner for the object o_k in the next round. The possibilities for auction initiation are described in Section 4.1.2. Subsequently, the cluster members c_i^k track the object in their FOV. With a set of parameters they bid with a utility α_i^k for the object at the owner's side. The composition of the utility α_i^k is discussed in Section 4.1.1.

In market-based clustering each camera (cluster head c_h^k as well as cluster member c_i^k) tries to maximize its local utility $\mathcal{A}_i$ which is given by

$$\mathcal{A}_i = \sum_{k \in O} \alpha_i^k - p + r. \quad (2)$$

The parameter p describes all payments made and the parameter r all received payments in this iteration. According to the Vickrey auction mechanism [23], the state estimation responsibility is transferred to the highest bidder, but at the price by the 2nd highest bidder. This strategy imposes to bid truthful valuations from the camera side instead of speculations. If $\mathcal{A}_i$ can be increased by selling o_k, the owner chooses the highest bidder to be the next owner of the object o_k and thus to become the next cluster head c_h^k. The estimation process and thus, the dynamic cluster "follow" the object's trajectory through the network, as illustrated in Fig. 2.

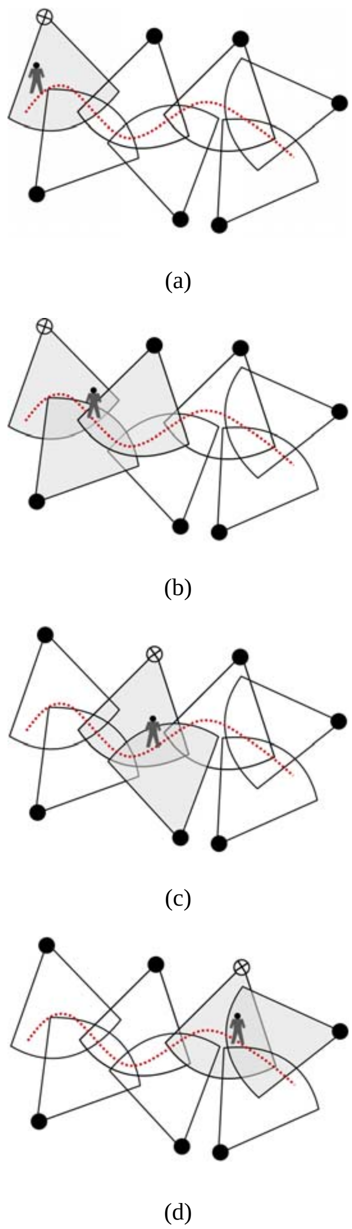


Fig. 2. Dynamic clustering in a VSN. Fig. 2a to Fig. 2d show the clustering in specific time steps.

4.1.1. Utility Definition

The utility α_i^k is used as value to bid for an object o_k in the FOV, if an auction is initiated by the current cluster head c_h^k. The utility is an election criterion that can be defined with different parameters. In our approach, the utility is based on two parameters: the available resources on the camera and the confidence in the tracking performance.

Storage, communication and processing power are the most critical resources in VSNs. Especially in VSNs with heterogeneous camera systems the individual distribution of resources can express how many tasks can be fulfilled by a specific camera. The available resources are indicated with $R_{total;i}$, normalized in the range $0 \leq R_{total;i} \leq 1$, and can describe any resources the designer perceives to pay attention to. As already mentioned, we typically pay attention to exchanging, processing and storing the observations retrieved by the visual sensor. The task of sensing is ignored in the resource model, due to its continuous execution. The resources totally available are described with

$$\begin{aligned} R_{total,i} &= \sum \lambda r \\ &= \lambda_0 * \sum [r_{WL}^k] + \lambda_1 * r_{E,i} \\ &\quad + \lambda_2 * r_{MEM,i} + \lambda_3 * r_{COMM,i} \end{aligned} \tag{3}$$

The parameter $\lambda = [\lambda_0; ... ;\lambda_3]$ with $\sum \lambda_r = 1$ indicates the weights of the resources we pay attention to. The individual resources are denoted with i) r_{WL}^k as the workload for each object o_k in terms of processing power, ii) $r_{E,i}$ as the total energy available on the node, iii) $r_{MEM,i}$ as the total memory available on the node and iv) $r_{COMM,i}$ as the amount of communication performed. Each parameter is a normalized value between 0 and 1. For the cluster-based protocol presented in Section 5 and the corresponding evaluation in Section 6 we use the parameter $r_{E,i}$, solely. Therefore, we set the parameter $\lambda_1 = 1$, all others are 0. Thus, we only consider the total energy available on each camera c_i as we focus on battery powered smart cameras.

The other parameter used for calculating the utility is the local confidence in the tracking performance of the camera on the object o_k. The confidence, denoted with ζ_i^k, can be obtained in various ways. One approach is to derive the confidence out of the matched features when comparing the tracked object to a given model. Another possibility is to use ζ_i^k as a visibility parameter described as a binary value [0;1]. A 1 indicates that the object ok can be detected in the FOV of camera c_i. With a 0 we express that the object is not detected. Thus, ζ_i^k can be considered as membership function being a part of the cluster or not. The utility α_i^k is then defined as

$$\alpha_i^k = \zeta_i^k * R_{total,i} \tag{4}$$

The utility α_i^k is only positive, and thus a valid bid, if the object o_k is visible to camera c_i.

4.1.2. Auction Initiation

The cluster head initiates an auction for the election of the cluster head for the next time step. The election is necessary, to select a camera with sufficient resources and confidence of the tracking performance. If the cluster head *sells* the responsibility for object o_k, we hand over the object ID together with the actual state as initial state for the processing by the new cluster head. If the cluster head can maximize its own utility by *keeping* o_k it remains the cluster head for the next time step as well.

Selecting a proper time for the handover is essential to limit the communication overhead produced by the market-based approach itself. As can be seen in Figs. 2a to 2d a new cluster head is elected after an auction initiation. We identified three possibilities, when an auction can be initiated: (i) α_h^k == 0: The utility of c_h^k is equal to zero—the worst case with no available resources at all or the object is no longer in the camera's FOV. (ii) $\alpha_h^k < \alpha_{thr}$: The utility of c_h^k is smaller than a given threshold. (iii) We continuously initiate an auction at regular intervals to examine $\alpha_h^k < \alpha_i^k$, hence, the utility of c_h^k is smaller than the utility of a cluster member c_i^k.

For the cluster-based protocol presented in Section 5 and the corresponding evaluation in Section 6 we apply the auction initiation point ii.

4.2. State-Space Model

The objective of a VSN is to detect and track objects. As we assume overlapping FOVs in the VSN we can perform cooperative state estimation on the object's state. In our approach we choose a continuous state, describing the position and the velocity of the object moving in the VSN. Equation 5 describes the state s consisting of position (x,y) and velocity ($\dot{x}, \dot{y}$) of an object o_k determined by camera c_i at time step t.

$$s_i^k(t) = [x_i^k(t), y_i^k(t), \dot{x}_i^k(t), \dot{y}_i^k(t)] \qquad (5)$$

The state is modeled in a linear state-space model. As an approach for cooperative state estimation Song et al. [6] designed a Kalman Consensus model for fully distributed processing in VSNs. Their approach serves as reference system in Section 6. Furthermore, we apply the Kalman Consensus Filter of [6] in the cluster-based protocol.

5. The Cluster-Based Protocol

In our cluster-based protocol (cp. Algorithm 1) the camera can take on either of the following two roles for each object in its FOV.

Cluster Head c_h^k. The cluster head is an elected camera in the VSN. It has the task of collecting the observations and the bids from the cluster members. Further, it performs a state estimation algorithm and initiates an auction to trigger cluster head election if necessary.

Cluster Member c_i^k. First, a cluster member waits for a defined timeout to receive a request for auction initiation. After receiving the request, the cluster member provides the cluster head its observation and bid to a corresponding object. If no request for auction initiation was received within the timeout, the camera assigns itself as cluster head. This procedure is denoted as the initialization phase of the clusterbased protocol.

Algorithm 1: The cluster-based protocol for a resource-aware state estimation.

```
ObjectDetection();
for ∀ o_k in FOV do
    role (k)=role (k̃);
    GetObservation();
    switch (role (k)) do
        case (c_h^k):
            InitiateAuction();
            ReceiveInformation();
            PerformStateEstimation();

            if (α_i^k > α_h^k) then
                Handover (s_i^k);
                role (k̃)=c_i^k;
        case (c_i^k):
            WaitTimeout();
            if (ReceiveRequest()) then
                SendInformation();
            else
                role (k̃)=c_h^k;
            if (ReceiveHandover()) then
                role (k̃)=c_h^k;
```

For each iteration of the cluster-based protocol in Algorithm 1, the initial task for all cameras in the VSN is the detection of the objects. As already mentioned, we assume a global identifier for each object known by the cameras in the VSN. With `ObjectDetection()` we can identify all objects o_k in the FOV of a camera. The first task for each camera is to take over the role in `role(`$\tilde{k}$`)` of the camera from the previous time step of the specific object o_k. For all detected objects o_k the camera retrieves its local observations z_i^k in Observation() and calculate the corresponding utility α_i^k.

If the camera is a cluster head c_h^k it initiates an auction by `InitiateAuction()`. Thereafter, it receives the observations z_i^k — in our case the object position on ground plane — as well as the utlity α_i^k from the cluster members with `ReceiveInformation()`. In the next routine it performs state estimation to optimize the object's state in `PerformStateEstimation()` by integrating the received observations from the cluster members. Now if one of the received utilities α_i^k is smaller than the local utility of the cluster head α_h^k, the cluster head performs a handover and transmits the current estimated state s_i^k using `Handover(` s_i^k `)` to the new cluster head c_i^k. Thereby, it assigns itself as cluster member.

In case, the camera is a cluster member c_i^k, it first waits for a defined timeout in `WaitTimeout()`. If c_i^k is able to receive a request from the cluster head in `ReceiveRequest()`, it transfers the object's observation z_i^k as well as the corresponding utility α_i^k to the cluster head in `SendInformation()`. On the other hand, if c_i^k has not received a request, it assigns itself as cluster head c_h^k Further, if c_i^k receives the message `ReceiveHandover()` it assigns itself to the cluster head c_h^k and adopts its tasks in the next time step. With the self-nomination it is possible to assign multiple cluster heads for a single object. Nevertheless, in the next iteration of the algorithm the auction initiation process elects a cluster head through the exchanged utilities. Thus, after the first bidding process the issue on multiple cluster heads is resolved. In this process, each self-nominated cluster head initiates an auction, in which the cluster head with the highest utility keeps its role.

5.1. Additional Settings to the Cluster-Based Protocol

To keep the dynamic cluster head allocation as lightweight as possible, cluster members do not know each other, only the cluster head is in knowledge of them. Joining a cluster is straightforward. If a camera detects an object, it waits for a predefined timeout to receive a message for auction initiation by an already existing cluster head. Since it is able to detect and identify the object, it has also information about its state and the related utility. Leaving the cluster is only possible, if the camera is not able to detect the object in its FOV. Nevertheless, if this is the case, the camera shows simply no reaction on messages for auction initiation. Thus, the cluster head would not receive any further information related to the object by this camera. Further, a camera failure or a camera adding to the network would not disturb the process of the clustering protocol.

5.2. State Estimation

In this work we apply the Kalman Consensus Filter (KCF) proposed by [6] as state estimator. The major steps are summarized in Algorithm 2. In the information form of the Kalman Filter, prediction and update are done in one step,

$$\begin{aligned} s_i^k(t+1) =& A_i(t) s_i^k(t) \\ &+ K_i^k(t)[z_i^k(t+1) - H_i(t) s_i^k(t)] \end{aligned} \tag{6}$$

The observations of the cameras are indicated with $z_i^k(t+1)$ identically described as the state in Equation 5 with position ($x_i^k(t), y_i^k(t)$) and velocity ($\dot{x}_i^k(t), \dot{y}_i^k(t)$). Further, A_i is denoted as the state change for each time step t and H_i referred to as the observation matrix, which maps the true state space into the observed space. The Kalman gain K_i^k defines how much the difference between the previous estimation and the actual measurement influences the actual estimation. Algorithm 2 summarizes the main steps in KCF state estimation. As input for the state estimation in `PerformStateEstimation()` serves the current observation $z_i^k(t+1)$, the state $s_i^k(t)$ from the last time step t and the corresponding covariance matrix $P_i^k(t)$. First, the information matrix and the information vector are built in `BuildInformation()`. The information vector u_i in Equation 7 is a statistical generalization of the observation, whereas the information matrix U_i in Equation 8 builds the covariance matrix expressing the uncertainty in the estimated values of the system state.

Algorithm 2: `PerformStateEstimation()`

input : current observations $z_i^k(t+1)$ from all participants (c_i^k and c_h^k); state $s_i^k(t)$ and covariance matrix $P_i^k(t)$ from the last time step t.

output: state $s_i^k(t+1)$ and covariance matrix $P_i^k(t+1)$ update.

```
BuildInformation();
StateEstimation();
StateUpdate();
```

$$u_i^k = \sum_i H_i^T R_i^{-1} z_i^k \tag{7}$$

$$U_i^k = \sum_i H_i^T R_i^{-1} H_i \tag{8}$$

Within `StateEstimation()` the state is estimated as described in Equation 6. The state s_i^k as well as the error covariance P_i are updated in `StateUpdate()` with

$$
\begin{aligned}
P_i^k(t+1) &= A_i(t)M_i^k(t)A_i(t)^T + B_i(t)Q_i(t)B_i(t)^T \\
s_i^k(t+1) &= A_i(t)s_i^k(t)
\end{aligned}
\tag{9}
$$

where $M_i^k(t) = (P_i^k(t)^{-1} + U_i^k)^{-1}$. Finally we return P_i^k and s_i^k as inputs for the iteration in the next time step.

5.3. Relaxing the Assumption on Object Re-Identification

In the underlying approach a perfect object reidentification is assumed. As it is well-known, the task of object re-identification is a tremendous challenge in image processing in terms of computation cost and re-identification accuracy. Re-identifying the same object, especially in multi-object tracking scenarios, with a classification accuracy of 100% is extremely difficult, if not impossible to realize. Typical examples are that the target objects get lost due to overlap with other moving or static objects, changes in illumination, incidences of light or extremely fast moving targets. The error in re-identification also directly infects the credibility of the underlying observations on the object. To overcome and smooth errors generated through the loss of objects in reidentification, we focus on compensating the errors made in vision, especially in re-identification.

By utilizing mathematics, object re-identification can be described as a static classification issue. The classification is formed related to set theory expressing a class by describing an arbitrary attribute $\mathcal{A}$ with

$$\forall y : (y \in o | \mathcal{A}(o) \Longleftrightarrow \mathcal{A}(y)), \tag{10}$$

where y can only take two labels for object o_k [24]. Thus it forms a binary problem whether the object o_k has the attribute of being correctly identified—positive—or not—negative. This gives us two attributes $\mathcal{A} = [\mathcal{A}_1, \mathcal{A}_2]$. Thus, mapping this statement on sets, the object o_k can belong only to one class with

$$A \neq B \Longleftrightarrow \forall o_k : (o_k \in A \Longrightarrow x \notin B), \tag{11}$$

where *A* and *B* are two sets or classes, respectively. If an object o_k is identified, it belongs to class A with $\{o_k|\mathcal{A}_1\} \in A$. If it is not identified, it belongs to class B with $\{o_k|\mathcal{A}_2\} \in B$. Being positively or negatively identified are also known as *true positives* and *true negatives*. The downside includes also *false positives* — wrongly identified objects —and *false negatives* — wrongly not-identified objects. These types are summarized in Table 1. Within the literature, this 2×2 matrix is known as contingency table comparing two nominal variables—the actual class and the decisions of the classificator.

The major consequence of a false negative or a false positive classification — or rather identification — of an object are illustrated in Fig. 3 and Fig. 4. The figures show two objects o_1 and o_2 to be identified by cameras c_1, c_2 and c_3. Further, the rectangles beside these cameras indicate a list of objects locally identified. Firstly, Fig. 3a shows the correct identification of object o_1 by camera c_1. Following the proposed cluster-based protocol in Fig. 3b, camera c_3 would receive an auction initiation from c_1. Unfortunately the object o_1 was *falsely* identified by c_3 as o_2. Thus, the camera c_3 doesn't react on the auction initiation, but rather builds a new cluster for the falsely identified object o_1. The risk exists that o_1's measurements get lost as this object is identified as o_1 and o_2 within the VSN. A match of their identifiers is no longer possible. A second failure could arise as illustrated in Fig. 3b. A second object, namely object o_2 moves into c_3's FOV during the auction initiation process with c1. In this case, object o_2 is identified by c_3 as o_1. Thus, c_3 is bidding for the *falsely* identified object o_1. If it owns the object in the next step o_1 comes up with wrong position information. Further, another camera in the VSN could be able to *positively* re-identify object o_2. Thus, several previous position information details get lost. Moreover, the object o_1 still exists in the VSN. This could lead to further collisions in position exchange when initiating an auction.

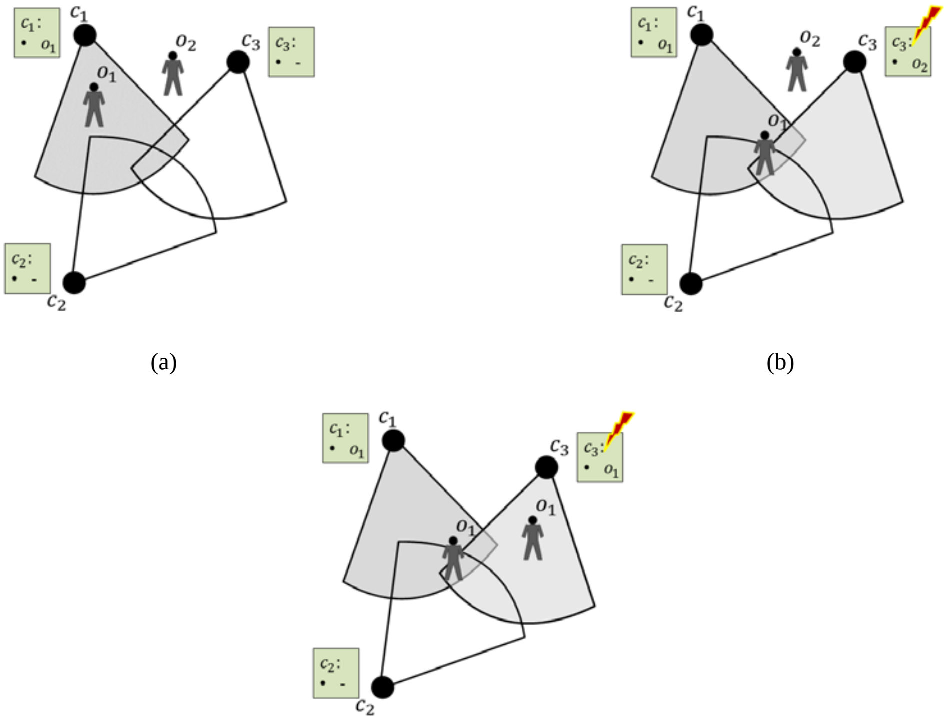
Secondly, Fig. 4 shows an example of false negative identification. As we can see in Fig. 4a the objects o_1 and o_2 are prepared to enter the observation area of the VSN. In Fig. 4b c_1 should have both objects in its FOV. Nevertheless, only object o_2 was identified.

5.3.1. Diminishing the Influence on False Positives

As described in the previous section, the object identification part is considered as classification problem. From Table 1 we can see that only 2 classifications produce problems: i) false positives, where the object is identified although its real ID is different, and ii) false negatives, where the object is not identified at all. Subsequently, we focus on diminishing false positives already one step before the identification process makes its classification.

Table 1. Classification in terms of object re-identification.

	Object's ID is x	**Object's ID is $\overline{x}$**
Object identified	True positive	False positive
Object not identified	False negative	True negative



(a) (b) (c)

Fig. 3. Illustration of false positive classification in the application of object re-identification.

(a) (b)

Fig. 4. Illustration of false negative classification in the application of object re-identification.

In the proposed cluster-based protocol, the cluster head initiates auctions at pre-defined intervals or as reaction to events. The message `InitiateAuction()`, introduced in Algorithm 1, is received by the cluster members as well as by other cameras in range. To overcome a false positive classification as illustrated in Fig. 3b and Fig. 3c we propose to hand over more information about the object. The message auction initiation contains the ID of the object o_k. We expand its payload with information on the object position on the ground truth.

$$\texttt{InitiateAuction()} = (ID_k; x; y) \qquad (12)$$

In case a camera receives the `InitiateAuction()` it has the object's ID together with its position locally available. This allows to compare the retrieved with the locally gained values in order to match the appropriate object. Nevertheless, this asks for the assumption that the cluster head has a true positive identification.

6. Simulation and Experimental Results

We evaluate the proposed resource-aware state estimation with a dynamic clustering approach by simulation studies. For these evaluations we use a new VSN-Simulator [25], a graphical simulator built in the game engine Unity3D. The reason for developing a new simulator beside the existing ones, as presented in [14], was to create a tool that i) is easy in installation, use and extension of the simulation environment, ii) can model multiple cameras, iii) having a simulator close to real-time performance (up to now all 2 seconds a measurement is made), and iv) getting a fancy looking and thus motivating environment with multiple GUI elements. The VSN-simulator provides 26 smart cameras set to 14 emulated office rooms. Fig. 5 shows a screenshot of the simulator, with 3 chosen camera views and buttons to interact with (add objects, delete objects, switch tracking of objects on/off and save the observations). In the simulation environment the cameras have overlapping FOVs and the object identification as well as all other processing task concerning clustering and state estimation runs locally on the cameras. The corresponding scripts to the processing tasks were written in C#. The tracking of the object is realized by the so-called raycast method provided by Unity3D. In simple terms, if a camera has a sustained sight on the object, it gets the object's coordinates.

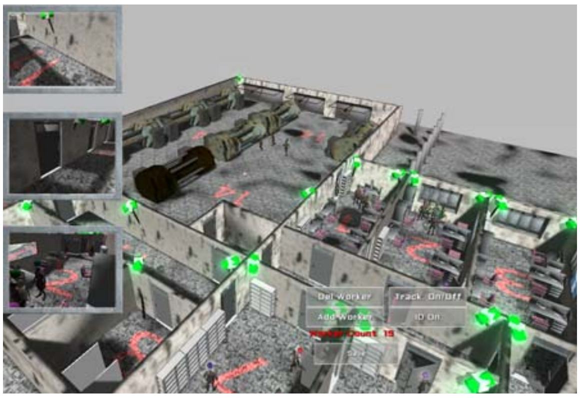


Fig. 5. A screenshot of the VSN-Simulator.

Fig. 6 shows the scenario for the underlying simulation results together with the trajectory the object is moving on. In our evaluation we consider a single room of the VSN-Simulator equipped with 9 cameras. The object is following pre-defined waypoints. The coordinates on the object's trajectory are further denoted as ground truth. Within a simulation environment, the cameras need individual observations from the object. Thus, each camera's tracking output is a random modification of the ground truth. In this evaluation we set the modification value randomly to a standard deviation of 3 length units.

6.1. Performance Measure

A first evaluation is referred to the accuracy of the object state gained by the proposed resource-aware dynamic clustering protocol. To compare the clusterbased protocol with the fully distributed approach of [6] both were implemented to the simulator using C#. For this evaluation we present the results for t = 1, ..., 52 measurement points on the object's trajectory of Fig. 6. On an average 5 of the 9 cameras have the object in their FOV. Thus, for the clusterbased protocol, we have 5 participants (consisting of cluster head and cluster members) on average as well.

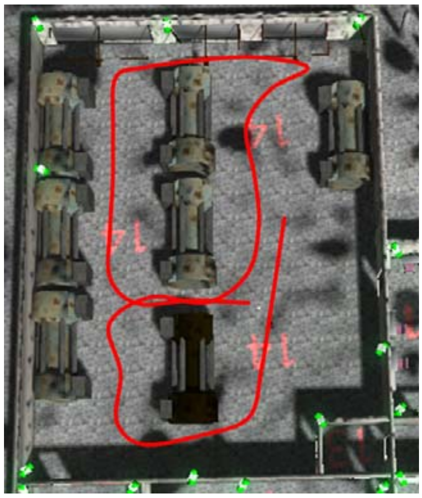


Fig. 6. A screenshot of the considered scenario in the VSN-Simulator including the object's trajectory.

Fig. 7 illustrates the estimated x and y coordinates for a single object, comparing the fully distributed approach with the cluster-based protocol to the camera output, a random modification of the ground truth of the observed object. The difference of the estimated object's state between the applied filters using the cluster-based protocol and the fully distributed approach is evaluated in Table 2.

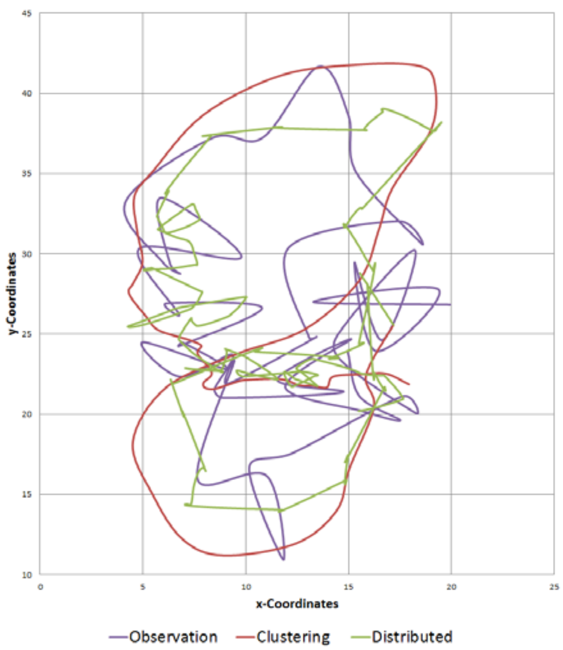


Fig. 7. Comparing the results of state estimation in the cluster-based protocol, the fully distributed approach and the camera observation.

Table 2. Comparison between the fully distributed and the clustering approach of the RMSE and the standard deviation s to the ground truth of the tracked object.

Method	RMSE		σ	
	x	y	x	y
Fully distributed	1.06	0.93	0.64	0.54
Clustering approach	1.71	1.88	1.24	1.08

The state accuracy computed in a distributed approach achieves a higher accuracy of the actual object's state. Comparing the root-mean-square errors (RMSE) and the standard deviation σ in x and y, the distributed approach was able to reduce the error to almost 50%. The reason is that in the distributed approach the cameras exchange full states and the error covariance matrices instead of single observations as with the clusterbased protocol.

6.2. Resource Measure

Further evaluations are related to the resource consumption of the VSN within the simulation environment. Therefore, we record the exchanged messages as well as the operations for state estimation applied for the cluster-based protocol and the fully distributed approach. Both of the following evaluations include the initialization phase of the cluster-based protocol.

In the simulator we have the following settings for this evaluation: The communication channel is wireless and thus, we exchange the messages by broadcasting. We record the exchange of the messages for both approaches in 1 office room of 9 cameras with overlapping FOVs (see Fig. 6). Further, in this evaluation we consider t =1, ..., 52 measurement points, as for the performance measure in Section 6.1. As before, we have on average of 5 of the 9 cameras having the object in their FOV.

6.2.1. Communication Effort

Table 3 describes the messages for both approaches with its content. As can be seen, the total payload of the fully distributed approach exceeds the one of the cluster-based approach. The message payload is based on the standard C data types short int (2 Bytes) and float (4 Bytes). Table 4 shows the average number of messages exchanged for 52 measurement points per camera. If we multiply the retrieved number of the individual message types with the payload in Bytes from Table 3, we get for the distributed approach 165.48 Bytes, for the cluster-based protocol only 14.14 Bytes on average per camera.

Table 3. Message types in the distributed approach and the cluster-based protocol together with content and total payload size in Bytes.

Fully Distributed approach		
Message Type	**Content**	**Payload (Bytes)**
`ExchangeInformation()`	u_i^k, U_i^k, s_i^k	80 Bytes
`ExchangeFinalState()`	s_i^k, P_i^k	72 Bytes
Cluster-based Protocol		
Message Type	**Content**	**Payload (Bytes)**
`InitiateAuction()`	c_h^k	2 Bytes
`SendInformation()`	z_i^k, α_i^k	8 Bytes
`Handover()`	s_i^k, P_i^k	48 Bytes

Table 4. Number of messages exchanged in the fully distributed and the clustering approach for 52 measurement points on average per camera.

Fully Distributed approach	
Message Type	**Avg. Number per c_i**
`ExchangeInformation()`	1.09
`ExchangeFinalState()`	1.09
Cluster-based Protocol	
Message Type	**Avg. Number per c_i**
`InitiateAuction()`	0.19
`SendInformation()`	0.94
`Handover()`	0.13

6.2.2. Computational Effort

In a further analysis, we focus on comparing the amount of operations. Therefore, we compare the number of additions and multiplications between the two approaches. The result is shown in Table 5 comparing the average number of additions and multiplications for 52 measurements per camera. From the simulation result of Table 5 we can see that the clustering approach needs a much lower number of operations on average. For the cluster-based protocol the number of operations N_o on average are given with

$$N_0^k(t) = t * (82.42 + 89.45) \quad (13)$$

with t as the measurement point index. The first number indicates the additions, the second the multiplications for state estimation and the clustering process. In the distributed approach each camera has the same number of operations to execute on average given by

$$N_0^k(t) = t * (399.02 + 356.77) \quad (14)$$

again, with t as the measurement point index. The first number indicates the number of additions, the second the number of multiplications. In the cluster-based protocol the state estimation is calculated only on the cluster head. In contrast, in the fully distributed approach the operations are executed on each participating camera. Thus, we can achieve an enormous reduction of processing and storage consumption in the cluster-based approach when comparing it to the fully distributed approach.

Table 5. Number of operations for state estimation in the fully distributed approach and the cluster-based protocol for 52 measurement points on average per camera.

Method	Additions	Multiplications
Fully distributed	399.02	356.77
Cluster-based	82.42	89.45

7. Conclusion and Future Work

In this paper we propose resource-aware state estimation with a cluster-based protocol for VSNs with limited capacities in storage, processing and communication. Our simulation results show that the achieved accuracy of the state estimation in the cluster-based protocol declines compared to fully distributed systems. Nevertheless, the achieved reduction of communication and storage consumption confirm that the cluster-based protocol is a highly applicable resourceaware approach for VSNs. Thus, a trade-off between accuracy and resource-awareness exists for object tracking applications in low-power systems. The next step is to integrate the validated approach into a VSN of real cameras. As a low-cost development platform we use the pandaboard[1] extended with a standard web cam. Furthermore, an approach is presented to relax the assumption on perfect object reidentification that is typically assumed in approaches dealing with coordination and control. This approach can be used to classify the object's state given by the tracker as output.

References

[1]. W. Ren and R. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, *IEEE Transactions on Automatic Control*, Vol. 50, No. 5, 2005, pp. 655– 661.

[2]. S. Soro and W. Heinzelman, A survey of visual sensor networks, *Advances in Multimedia*, Vol. 2009, 2009, p. 21.

[3]. L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, Socio-economic vision graph generation and handover in distributed smart camera networks, *Transactions on Sensor Networks*, Vol. 10, No. 2, 2014, pp. 20:1–20:24.

[4]. R. Olfati-Saber and N. Sandell, Distributed tracking in sensor networks with limited sensing range, in *Proceedings of the IEEE American Control Conference*, June 2008, pp. 3157–3162.

[1] http://pandaboard.org/

[5]. C. Ding, B. Song, A. Morye, J. Farrell, and A. Roy-Chowdhury, Collaborative sensing in a distributed ptz camera network, *IEEE Transactions on Image Processing*, Vol. 21, No. 7, July 2012, pp. 3282–95.

[6]. B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, Distributed camera networks, *IEEE Signal Processing Magazine*, Vol. 28, No. 3, May 2011, pp. 20–31.

[7]. C. Soto and A. Roy-Chowdhury, Distributed multi-target tracking in a self-configuring camera network, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 1486–1493.

[8]. V. Bhuvana, M. Schranz, M. Huemer, and B. Rinner, Distributed object tracking based on cubature Kalman filter, in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, November 2013, pp. 423–427.

[9]. S. K. Chaurasiya, T. Pal, and S. D. Bit, An enhanced energy-efficient protocol with static clustering for WSN, in *Proceedings of the International Conference on Information Networking*, 2011, pp. 58–63.

[10]. A. S. Zahmati, B. Abolhassani, A. B. S. Asghar, and A. S. Bakhtiari, An energy-efficient protocol with static clustering for wireless sensor networks, *International Journal of Electronics, Circuits and Systems*, Vol. 1, No. 2, 2007, pp. 135–138.

[11]. H. Medeiros, J. Park, and A. Kak, Distributed object tracking using a cluster-based kalman filter in wireless camera networks, *IEEE Journal of Selected Topics in Signal Processing*, Vol. 2, No. 4, Aug. 2008, pp. 448–463.

[12]. M. Taj and A. Cavallaro, Distributed and decentralized multi-camera tracking : a survey, *IEEE Signal Processing Magazine*, Vol. 28, No. 3, 2011, pp. 46–58.

[13]. J. Mallett, The role of groups in smart camera networks, Ph. D. dissertation, *Massachusetts Institute of Technology*, 2006.

[14]. F. Qureshi and D. Terzopoulos, Smart camera networks in virtual reality, in *Proceedings of the IEEE*, Vol. 96, No. 10, 2008, pp. 1640–1656.

[15]. M. Hooshmand, S. M. R. Soroushmehr, P. Khadivi, S. Samavi, and S. Shirani, Visual sensor network lifetime maximization by prioritized scheduling of nodes, *Journal of Network and Computer Applications*, Vol. 36, 2013, pp. 409– 419.

[16]. E. S. Torshizi and E. S. Ghahremanlu, Energy efficient sensor selection in visual sensor networks based on multi-objective optimization, International *Journal on Computational Sciences and Applications*, Vol. 3, 2013, pp. 37–46.

[17]. C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan, and M. Abidi, Camera handoff with adaptive resource management for multicamera multi-target surveillance, in *Proceedings of the 5th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2008, pp. 79–86.

[18]. B. Dieber, C. Micheloni, and B. Rinner, Resource-aware coverage and task assignment in visual sensor networks, *IEEE Transactions on Circuit and Systems for Video Technology*, Vol. 21, 2011, pp. 1424 – 1437.

[19]. E. Monari and K. Kroschel, Task-oriented object tracking in large distributed camera networks, in *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010, pp. 40–47.

[20]. O. Younis and S. Fahmy, Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *IEEE Transactions on Mobile Computing*, Vol. 3, No. 4, 2004, pp. 366– 379.

[21]. J. SanMiguel and A. Cavallaro, Cost-aware coalitions for collaborative tracking in resource constrained camera networks, *IEEE Sensors Journal*, 15, 5, 2014, p. 2657 - 2668.

[22]. M. Schranz and B. Rinner, Resource-aware state estimation in visual sensor networks with dynamic clustering, in *Proceedings of the 4th International Conference on Sensor Networks*, 2015, p. 10.

[23]. W. Vickrey, Counter speculation, auctions, and competitive sealed tenders, *The Journal of Finance*, Vol. 16, No. 1, 1961, pp. 8–37.

[24]. J. Barwise, Handbook of Mathematical Logic, *North Holland Publishing Company*, 1977, Vol. 90.

[25]. M. Schranz and B. Rinner, Demo: VSNsim - a simulator for control and coordination in visual sensor networks, in *Proceedings of the 8th ACM/IEEE International Conference on Distributed Smart Cameras*, 2014, p. 3.