Centralised, Decentralised, and Self-organised Coverage Maximisation in Smart Camera Networks

Lukas Esterle
Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Birmingham, UK
l.esterle@aston.ac.uk

***Abstract*—When maximising the coverage of a camera network, current approaches rely on a central approach and rarely consider the decentralised or even self-organised potential. In this paper, we study the performance of decentralised and self-organised approaches in comparison to centralised ones in terms of geometric coverage maximisation. We present a decentralised and self-organised algorithm to maximise coverage in a camera network using a Particle Swarm Optimiser (PSO) and compare them to a centralised version of PSO. Additionally, we present a decentralised and self-organised version of ARES, a centralised approximation algorithm for optimal plans combining PSO, Importance Splitting, and an adaptive receding horizons at its core. We first show the benefits of ARES over using PSO as a single, centralised optimisation algorithm when used before deployment time. Second, since cameras are not able to change instantaneously, we investigate gradual adaptation of individual cameras during runtime. Third, we compare achieved geometrical coverage of our decentralised approximation algorithm against the centralised version of ARES. Finally, we study the benefits of a self-organised version of PSO and ARES, allowing the system to improve its coverage over time. This allows the system to deal with quickly unfolding situations.**

I. Introduction

In computing systems composed of a large number of individual autonomous systems, the benefits of decentralised coordination and control usually break down to removing a potential bottleneck in terms of communication and processing the data, removing a single point of failure/attack and increasing the robustness, adaptability, and flexibility of the system. However, using an approach that does not rely on a central component requires increased information exchange among the individual systems. At the same time, by having less information, we expect to have lower performance or quality in achieving the collective goal.

In this paper, we investigate the effects of different amount of available information, as well as the degree of cooperation within a network on the performance of reaching a common network-wide goal. We differentiate between centralised, decentralised, and self-organised coordination. In *centralised coordination*, a single system has information of the states of all entities in the network and this central coordinate defines how all systems in the network should behave. This is illustrated in Figure 1a. In a *decentralised coordination*, each system has knowledge about itself and the state of its geometrically close neighbours which can be achieved through message passing. Individual systems decide on their behaviour in collaboration with their neighbouring cameras in the network to arrive at the best solution. Different mechanisms can be used to arrive at a common, neighbourhood-wide solution, such as consensus, voting, or a simply greedy approach. This is illustrated in Figure 1b. Employing *self-organised coordination*, each system has again only knowledge about its neighbours. However, in contrast to a centralised or decentralised approach, using a self-organised approach each entity decides its behaviour on its own and does so during runtime. A snapshot of this is illustrated in Figure 1c.

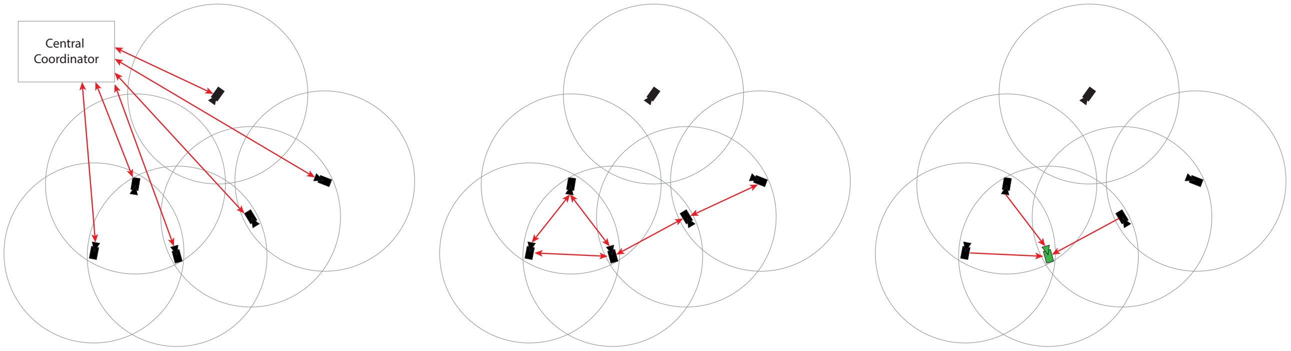
In this context, we investigate the well-studied problem of coverage maximisation in Pan-Tilt-Zoom (PTZ) camera networks. The main idea here is to adapt the orientation and focal length of each camera in order to maximize the geometrical coverage of the entire area using the individual fields of view (FOV). While we investigate maximum coverage, we are also interested in minimising the overlap between multiple FOVs.

While for smaller sized camera networks, a centralised approach is usually applied, we are interested in investigating approaches that are applicable to very large networks. This requires to switch from a centralised version towards a decentralised or even self-organised one.

Furthermore, cameras are not able to change their orientation and zoom instantaneous. Thus, an iterative approach that takes the gradual changes within the network into account is required. To achieve this, we employed the recently presented approach of ARES (Adaptive Receding-horizon Synthesis of Optimal Plans) [1]. This approach allows us to gradually apply changes to our individual cameras until they reach a very high coverage of the area.

Our contribution in this paper is threefold. First, we show the benefits of ARES in comparison to PSO in a centralised setting when considering global knowledge. Second, we present a distributed version of PSO and ARES and analyse their performance in terms of maximised coverage when only neighbourhood information is available. Third, we present a self-organised approach for coverage maximisation. This approach operates at runtime and is able to deal with quickly unfolding situations. Along these lines we are interested in investigating the following research questions:

1) Can we achieve efficient results in terms of coverage maximisation when limiting information exchange to individual neighbourhoods rather than collecting information about the entire network at a central controller?



(a) Centralised Coordination (b) Decentralised Coordination (c) Self-organised Coordination

Fig. 1. Illustration of the different coordination approaches investigated in this paper. A cameras' communication range is illustrated as a grey circle around each camera. The information exchange between different cameras or a central coordinator is illustrated as a red arrow. Figure 1c shows the self-organised coordination at a single time step, where the green camera is currently adapting its pose.

2) Is a self-organised approach able to achieve an efficient configuration over time where the performance is comparable to the resulting configuration of a centralised or decentralised approach?

This paper is structured as follows. In the next section, we define the problem formally. Afterwards, we discuss the related work. In Section IV we summarise the ARES approach and compare it to a simple PSO. Further, we show the benefits of ARES over this simple PSO. In Section V we extend ARES to operate in a decentralised fashion, where it is executed on individual cameras and only information of geometrically close neighbours is being considered. We investigate the benefits of a simple consensus and a greedy approach in this setting. In Section VI we consider online adaptation of our cameras using PSO and ARES. Here, each camera only considers local information in each step towards increased coverage of the observed area during runtime. We discuss our findings and conclude the paper in Section VII.

II. Problem Statement

In this paper we consider the problem of geometric coverage maximization of a given area A with a set of cameras $C = \{c_1, c_2, ...c_n\}$. Each of these cameras has a given position $\mathbf{x}_i = (x_i, y_i)$ and a FOV f_i defined by an angle ω_i, giving the orientation of the camera relative to a fixed position, its range r_i, and the angle of the FOV γ_i. This is illustrated in Figure 2.

However, we use the focal length fl_i to determine the angle γ_i as well as the range r_i of each camera c_i. To do this, we need to consider the size of the sensor s_i. We selected a standard size of 35mm for all cameras. Therefore, the angle of the FOV of a camera at time t is defined by

$$\gamma_i(t) = \arctan(\frac{s_i}{2 \cdot fl_i})$$

while at the same time the range of a camera is defined by

$$r_i(t) = \frac{fl_i}{m + fl_i}.$$

Here, m represents the minimum magnification required for a unit square to be detected by the camera. We set this value to $m = 0.3$ for all cameras. Based on the optical properties of standard cameras, we limit $fl_i = [20, 125]$.

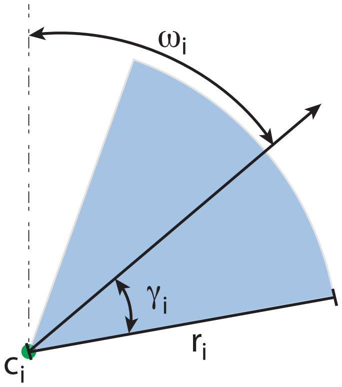


Fig. 2. Illustration of an object in a camera's FoV. The FoV is illustrated in blue with a range of r_i, an orientation of ω_i, and an angle γ_i on both sides of ω_i

Finally our cameras can change their orientation ω_i and their focal length fl_i. The change in focal length and orientation is given as

$$fl_i(t+1) = fl_i(t) + \mathrm{f}$$

and

$$\omega_i(t+1) = \omega_i(t) + \varpi$$

respectively, where f represents the change of the focal length and ϖ the change in orientation in degrees. We consider $\mathbf{k}(t) = \{\mathbf{c}_i(t)\}_{i=1}^n = \{\mathbf{x}_i(t), fl_i(t), \omega_i(t)\}_{i=1}^n$ to be a camera network configuration for a given set of cameras at time step t.

For a single camera, coverage is defined by the area of the circular sector representing the FOV f_i. We can now calculate the total coverage $\mathbf{C}$ of the entire network as follows

$$\mathbf{C}(t) = \bigcup_{i=1}^{n} f_i(t) \tag{1}$$

where the area $A \geq \mathbf{C}$.

For simply maximising the coverage of given area, maximising the focal length of each camera would be the dominant strategy. However, we are also interested in having as little unused areas in a FOV as possible. This unused area comes about when FOVs of cameras overlap. We calculate this network wide area as

$$\mathbf{U}(t) = \sum_{i=1}^{n} f_i(t) - \mathbf{C}. \tag{2}$$

We can now define a fitness for a configuration $\mathbf{k}$ for a given set of cameras as

$$F(\mathbf{k}(t)) = \alpha \cdot \mathbf{C}(t) - (1-\alpha) \cdot \mathbf{U}(t)$$

where the parameter α allows us to tune the fitness function according to the network-wide goals, i.e. maximising the coverage or minimizing the unused area. Ultimately, the goal is to find appropriate adaptation values for each fl and ω for each individual camera in the network over time that eventually leads to a maximised $F(\mathbf{k})$.

III. Related Work

In the presented work, we investigate the performance of decentralised and self-organised algorithms against a corresponding centralised version. We use the case study of maximising coverage in a distributed smart camera network using PTZ cameras.

The benefits and drawbacks of decentralised as well as self-organised systems have been discussed in a variety of different application areas ([2], [3], [4], [5], [6], [7]). Decentralised and self-organising systems are composed of autonomous individuals, able to specialise based on their current circumstances and their environment. They operate independent and hence can access their own resources as needed. Due to their individuality, they make decisions locally and do not rely on a central controller. This not only introduces more robustness to the system but also higher flexibility and adaptability in the face of dynamics in the system, such as added or failing individuals. Each individual responds directly to its environment, making an interaction with a central controller obsolete. However, there are also drawbacks an individual entity might for example end up performing the same operations in parallel. This results in a waste of resources, as operations might be performed unnecessarily when considered from a system-wide level. Furthermore, the constant exchange of information leads to a communication overhead. Not having a central controller also makes it more complicated to enforce system-wide changes (e.g. system strategies or preferences). Finally, since none of the individual entities has knowledge about the state of the entire system, solutions can only be found within the reach of the individual system.

We investigate the potential of decentralised, and self-organised coordination in comparison to centralised coordination in the context of coverage maximisation in distributed PTZ camera networks. Camera network coverage, also known as the *Art Gallery problem*, has been researched quite intensively ([8], [9], [10]).

Fusco and Gupta [11] consider the coverage of an area with a minimum number of cameras. They define specific points in this area and require each point to be covered by a predefined number of cameras at any time. In addition to applying greedy algorithm using a central controller they also propose a distributed version of the greedy algorithm. A distributed W-Learning approach is employed by each camera in the proposed mechanism by Rudolph et al. [12]. The W-Learner exchanges information about current states with neighbouring cameras to learn the optimal placement over time. In [13], we investigated the self-organised coverage in 360° cameras. There we were interested in the trade-off between covered objects moving around in the environment and the overlap between cameras. In contrast, we do not consider covering objects in this work, but are purely interested in geometric coverage of an area. Konda et al. [14] present a planning algorithm, maximising coverage for the entire network based on visual quality assessment. For this centralised approach they employ a Particle Swarm Optimizer. Yap and Yen [15] introduce time-awareness in PTZ camera networks. Their approach considers an area to be covered if a PTZ camera can cover it within a given time-constraint. In an iterative approach, they ensure that defined locations are being covered by at least one camera within the given time constraint. Altahir et al. [16] present a dynamic programming approach to maximise the coverage in a given network. In each iteration, they add another camera from the given network to the considered network and optimise them based on the previous solution. Their final solution needs to back-track all previous solutions.

IV. Centralised Coordination

When we use centralised coordination of a network or a collective, each individual either needs to be monitored by a central coordination unit or is required to constantly report back to this coordinator. This leads to a large overhead in communication. Furthermore, this introduces a single point of failure and a bottleneck when the size of the network increases. While the benefits of decentralised and distributed coordination are well known, we are interested in the potential performance of a centralised coordination of PTZ cameras for maximising their geometric coverage. We use this centralised approach as a benchmark for our decentralised technique and self-organizing mechanism.

In our centralised approach, the controller knows about all cameras and their current state. This corresponds to a controller with *global* knowledge. Furthermore, all individual cameras behave in a cooperative manner and perform the actions as defined by the central controller.

For a centralised approach to maximise coverage with PTZ cameras, we use two different *Optimisation Algorithms*, namely the well-known Particle Swarm Optimiser (PSO) [17] and ARES [1]—a novel approximation algorithm to calculate optimal control paths. In the following, we will briefly outline the idea of ARES.

A. ARES

ARES is a recently developed approximation algorithm to calculate optimal control paths for dynamic systems. At its core, ARES uses PSO to optimise each step towards a final configuration but it is extended using three important concepts:

1) Importance Splitting: Importance splitting is a technique to increase the probability of reaching rare events [18]. The idea is to decompose the rare event into a sequence of levels, where each level gets the system closer to the rare event. In order to increase the probability to reach the rare event, importance splitting replicates states and re-samples them if the state does not reach the next level towards a rare event. States not reaching the next level are discarded, keeping the number of concurrent states constant.
2) Adaptive Receding Horizons: When using limited adaptation, an optimiser might require to look beyond the next state. With an receding horizon, the optimiser can anticipate the actions of cameras in future steps and incorporate this knowledge in its control decision [19]. In ARES, this length of receding horizon is usually a single time step but can be adapted if the next step does not reach the next level.
3) Adaptive Particles: PSO uses a large but fixed number of particles in its optimisation process. However, ARES only uses PSO to find the best configuration to reach the next level and does so with multiple copies. Therefore, ARES can employ PSO with a low number of particles. Only if no solution reaching the next level has been found, ARES increases the number of particles slightly.

In short, the original ARES algorithm operates as follows. First, it calculates three fitnesses: (i) the current fitness of the system, (ii) the goal fitness it would like to achieve given the number of levels it would like to pursue, and finally (iii) the fitness required to arrive at the next level towards the goal fitness. Second, it generates copies of its initial configuration. Each of these configurations employs an individual PSO trying to find the best adaptation values to transit to the next configuration based on the given fitness function. If any of these next configurations has the fitness of the next level, these copies are re-used and randomly sampled to replace those configurations that did not reach the required fitness for the next level. Afterwards, the required fitness of the next level is calculated if the goal fitness has not been reached yet. This describes a single, successful step of ARES. If none of the copies has the fitness of the next level, ARES increases the receding horizon, enabling it to investigate potential configurations further in the future, beyond a single adaptation step. If this still does not lead to a state with sufficient fitness, ARES also increases the number of particles, effectively increasing the chance of finding better results. If the fitness satisfies the next level while applying a higher receding horizon, only the configuration after the first step of ARES is applied to the system. This allows ARES, on one hand to explore potentially better solutions in the following iterations and on the other hand, to apply a solution which is within the movement capabilities of the system. If the final level has been reached, or the number of particles and the receding horizon are exceeded, ARES stops.

Given we do not know the optimal solution for a given initial configuration, we do not specify a number of levels we would like to achieve *a priori*, which contrasts the approach by Lukina et al. [1]. Instead, we use the initially achieved performance as a baseline and try to improve on it over time. We therefore initialise the first level with the best value among all copies $\ell_\lambda = \widehat{F}(\mathbf{k}(1))$ with $\lambda = 1$ and each subsequent level as follows:

$$\ell_{\lambda+1} = \widehat{F}(\mathbf{k}(t)) \cdot \left(1 + \frac{1}{\lambda+1}\right), \tag{3}$$

where λ represents the current level of ARES and is increased by 1 with every new level achieved.

B. Results of Centralised Coordination

To evaluate the performance of centralised, decentralised, and self-organised coverage maximisation we use the camera network simulation tool CamSim [20]. We designed four qualitatively different scenarios for experimentation as shown in Figure 3.

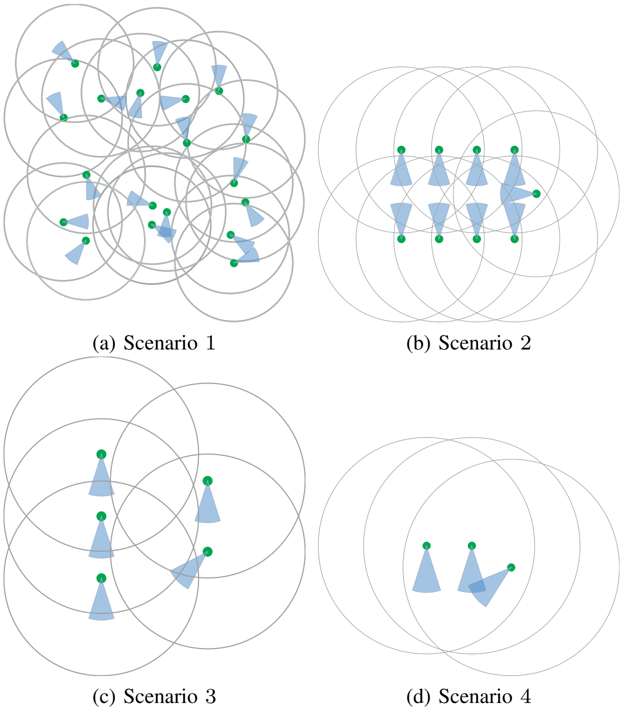

(a) Scenario 1
(b) Scenario 2
(c) Scenario 3
(d) Scenario 4

Fig. 3. Evaluation Scenarios: Green spots indicate the location of the camera. Grey circles indicate the potential range of the FOV and the range of communication. A blue cone shows the initial orientation and angle of the respective camera.

In our first set of experiments, we do not consider a time component to be of relevance. This means a camera is able to adopt any of its potential values without any delay. This would mean that a camera is able to turn by 180 degree without time delay or can change from its minimum focal length to the maximum focal length instantaneously. While this would not be possible during runtime, this behaviour would correspond to *a priori* and offline coverage maximisation similar to the approach presented by Fusco and Gupta [11]. We call these experiments without time constraint *optimal* as each camera c_i is able to select the optimal value from the potential values for fl_i and ω_i

In our second set of experiments, we introduce a previously discussed time constraint. Having such a time constraint allows each camera only limited adaptation. This might lead to a camera reaching a local optimum which it might not be able to overcome using traditional PSO. In such situations, ARES is able to improve the network-wide outcome by using a longer receding horizon. In time constraint experiments we limit the adaptation of the focal length and the orientation to $\mathrm{f} = [-10, +10]$ millimetre and $\varpi = [-5, +5]$ degree. We refer to these experiments as *step*, as each camera c_i can only adapt fl_i and ω_i within a limited range.

All experiments have been repeated 30 times and the average result is shown. Results are normalised by the maximum achieved coverage value across all studied α-values in a single experiment. Figure 4 shows the benefits of ARES in comparison to using just PSO in all four experiments. We employ PSO with 100 particles and 100 iterations to maximise coverage in the given area. ARES is executed with initially 10 particles and a maximum receding horizon of 5. The initial receding horizon of 1 is increased by 1 if no satisfactory solution could be found to reach the next level. We only increase the number of particles once by 20 if no satisfactory solution could be found. If a state reaches the next level, ARES resets the receding horizon and the number of particles to its initial value. ARES copies the initial state 10 times and keeps this number constant by re-sampling unsuccessful states. While we do have a lower number of particles per copy, the total number of particles in ARES equals the number of particles in PSO in its initial form. Nevertheless, ARES is able to outperform PSO in almost all cases. Only when we disregard the coverage in the fitness, PSO achieves better coverage within the network. Furthermore, the low number of cameras and hence low possible overlap in Scenario 4 results in equally good performance of PSO and ARES.

Furthermore, Figure 4 shows a comparison of achieved coverage of PSO and ARES when introducing a time constraint (i.e. only allowing limited adaptation in each step). When running PSO in this setting, PSO is executed multiple times and each result is applied intermediately. This corresponds to running ARES with a maximum receding horizon of 1, a single copy of the initial state, and a fixed number of particles (i.e. 75). In this case, we iterate PSO 100 times, where in each iteration the particles try to find the optimum solution. Due to the step-wise adaptation of each camera and the limited receding horizon of ARES, certain solutions might not be found as the *Optimisation Algorithm*, which deems this path infeasible. Throughout all experiments, ARES performs significantly better than PSO, when only applying step-wise adaptation with $\alpha \neq 0.00$. Even with this limitation, ARES is able to achieve more than 70% of the best solution (i.e optimal ARES), in many cases even more than 90% with $\alpha \neq 0.00$. Furthermore, in cases where $\alpha \neq 0.00$ ARES achieves double the coverage in comparison to step-wise adapting PSO.

While these results achieve high, network-wide coverage, we are interested in the performance of decentralised and self-organised approaches and the corresponding benefits they bring about such a system. In the next section, we introduce a decentralised version of ARES for this problem and analyse its performance against the performance of PSO in a decentralised setting as well as the previously achieved performances of the corresponding centralised approaches.

V. Decentralised Coordination

While the centralised version of ARES is currently applied to all cameras equally, one can easily imagine a distributed version of ARES as follows: ARES is executed with z copies of the current configuration of the system. This configuration represents the states of all cameras in the network. In the central version, each step in ARES is executed z times (i.e. PSO) in parallel. In a distributed version, where each camera could calculate the next configuration of the entire network, each camera c_i would execute ARES with only $\lceil \frac{z}{n} \rceil$ copies of the current configuration and share intermediate results at specified times. If $n > z$, only a subset of the cameras has to perform ARES or, alternatively, z could be increased, effectively increasing the probability of ARES finding a state with a fitness satisfying the next level.

However, such a distributed approach still requires information about the current configuration of the entire network. This means an enormous information exchange among the cameras in order to achieve this. In order to reduce this overall information exchange and to make the approach more robust and adaptive, we look into decentralised versions of ARES and PSO.

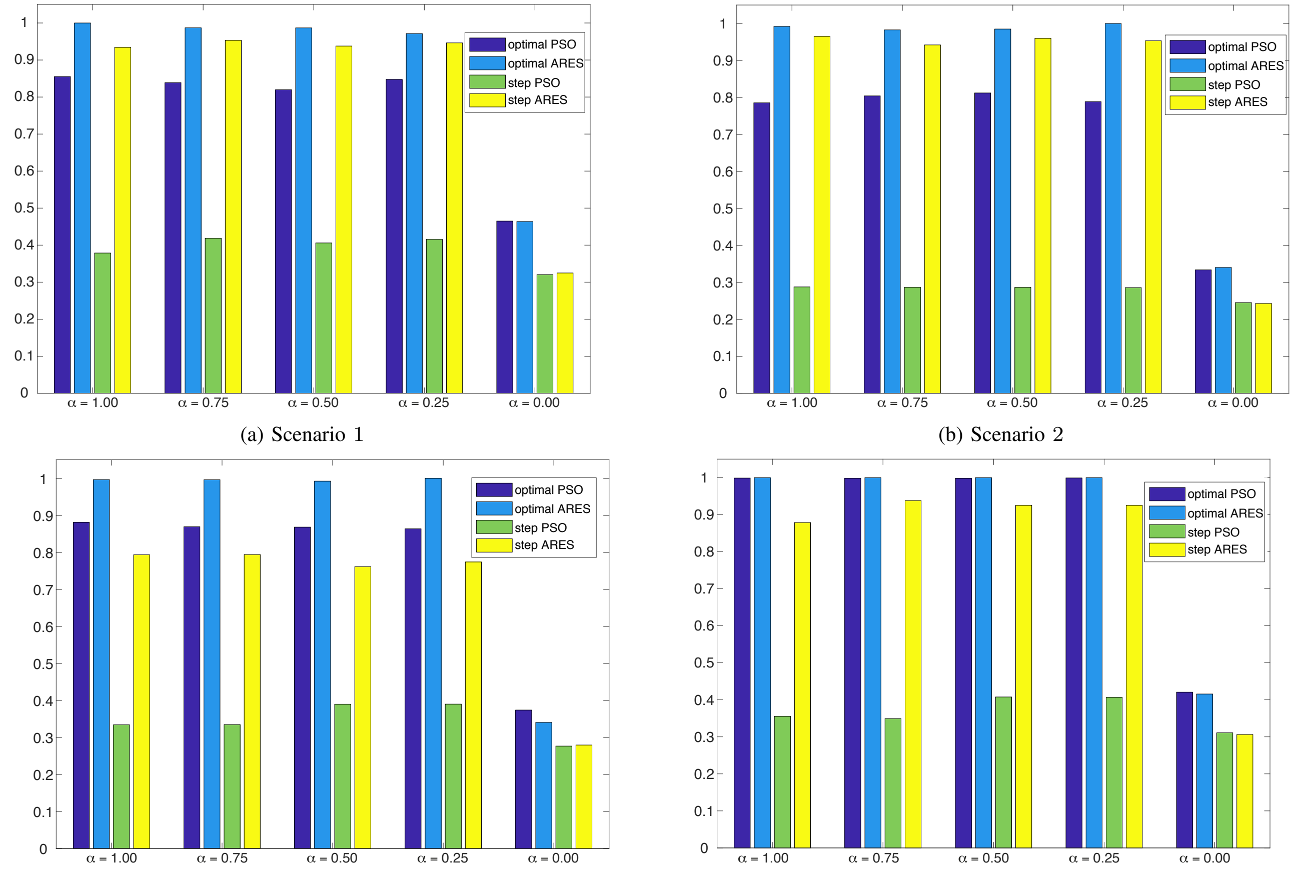
We limit our decentralised approaches to those requiring minimal information exchange among the cameras. Here, each camera performs the *Optimisation Algorithm* locally, using only available information of neighbouring cameras. We developed two different approaches on handling decentralised coordination: an ignorant decentralised approach and a consensus finding decentralised approach.

A. Ignorant behaviour

In this approach, each camera assumes it already found best configuration for itself as well as its neighbours. This means a camera will also assume that another camera arrives at the same conclusion and is ignorant to potential solutions of other cameras. Therefore, the results of the *Optimisation Algorithm* of the other cameras is not incorporated in their own outcomes. The behaviour of each individual camera is given in Algorithm 1.

B. Consensus finding behaviour

When cameras found their best solution using an *Optimisation Algorithm*, they communicate this solution to the



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

Fig. 4. Comparison of achieved normalised mean coverage using the centralised *optimal* and *step* PSO and ARES with different α values.

Algorithm 1: Ignorant Decentralised Coverage Maximisation

```
foreach $c_i \in C$ at time $t$ do
    Exchange camera states with neighbours $cn$ within
      range $r_i$.
    Perform Optimization algorithm (PSO or ARES)
      using own state and neighbouring cameras as
      configuration $\mathbf{k}_n$.
    Apply outcome to $c_i$.
end
```

neighbouring cameras that have been part of this solution. At the same time, they also acquire the best solutions from other cameras. Using this information, they use a simple average value consensus ([21], [22]) is given by

$$cons(O(t)) = \frac{\sum_{j=1}^{l} o_j}{l}. \tag{4}$$

An average consensus of decentralised optimisation algorithm using the collection of solutions $O(t)$ of neighbouring cameras at time t as input. Algorithm 2 represents the behaviour of each camera in the network. It is important to note that consensus is only made by the individual camera based on the solutions of surrounding cameras. This does not mean, that neighbouring cameras will arrive at the same solution. This can even be highly unlikely, if two neighbouring cameras c_i and c_j have disjunct sets of neighbouring cameras where $cn_i \backslash cn_j \neq \emptyset$. As results affect neighbouring cameras, these results may be propagated through the system leading to potentially beneficial but also disadvantageous configurations of the entire network.

Algorithm 2: Consensus-finding Decentralised Coverage Maximisation

```
foreach $c_i \in C$ at time $t$ do
    Exchange camera states with neighbours
      $N_C = n_1, \ldots n_l$ within range $r_i$.
    Perform *Optimisation Algorithm* (PSO or ARES)
      using own state and neighbouring cameras as
      configuration $\mathbf{k}_n$.
    Send outcome $o_i = (fl_j(t+1), \omega_j(t+1))$ of
      *Optimisation Algorithm* to neighbour $n_j$
    Receive outcome $O(t) = \{o_j\}_{j=1}^{l}$ from neighbours
      $N_C$ respectively.
    Apply apply $cons(O(t))$ as of Equation 4.
end
```

C. Results of Decentralised Coordination

For evaluation, we used the same experiments as in Section IV allowing us to compare the results directly. Again, all experiments have been performed 30 times and the average has been normalised by the best achieved coverage within each individual experiment across all performed α-values. Figure 5 shows the results for all four experiments. In contrast to the centralised version, ARES is only executed with 3 copies on each camera. In most cases, ARES is able to outperform PSO when using consensus finding as well as an ignorant decentralised approach. Interestingly, for Scenario 2, ignorant ARES performs particularly poor. We speculate, that this is related to the high density of cameras leading to a potentially high probability of overlaps. In addition, cameras are initially arranged in in such a manner, that they run into a local minimum very quickly. We further speculate that consensus ARES is able to overcome an initial minima but is not able to overcome further arising minima, where an ignorant ARES navigates itself into a minima it is not able to escape from anymore. However, the actual impact of initial configurations on the outcome of the overall approach remains an open research question.

As discussed before, consensus-finding decentralised ARES performs inconsistently. In all cases, consensus-finding decentralised ARES outperforms any PSO version by $10 - 30\%$, depending on the scenario and $\alpha > 0.00$. In some cases, especially in Scenario 2, consensus-finding decentralised ARES is even able to outperform ignorant decentralised ARES by almost 20%. In all other scenarios, our ignorant decentralised ARES approach is able to achieve at least 50%, in some cases even up to 85%.

VI. Self-organised Coordination

In the third approach, we assume each camera only has knowledge about the latest state of other cameras within a given communication range, as in with the decentralised approach. This communication range corresponds again to the maximum range a camera might be able to see if the focal length is increased to maximum. However, in this approach, the cameras do not exchange information about their latest adaptation but may only assume the behaviour of other cameras in their vicinity. Each camera performs the optimization locally. A camera may observe the change of others and adapt correspondingly. Importantly, and in contrast to the centralised and decentralised approaches, the adaptation is not performed concurrently by all cameras in the network but rather a single camera performs the adaptation individually at any time.

Due to the constant indirect interaction between the cameras, which is based on the observation of other cameras' state and evaluation of their own state, this results in a self-organized coverage maximisation over time. A short outline of this approach is given in Algorithm 3.

The self-organising coverage maximisation algorithm performs a single optimisation per camera in each time step. Over time, all n cameras are performing an optimisation step. This is done until time T is reached where $T \gg n$ allowing all cameras to perform sufficient adaptations.

Algorithm 3: Self-organised Coverage Maximisation

```
for $t = 1$ to $T$ do
    Select next camera $c_i$ where $i = t$ MOD $n$.
    Collect states of neighbouring cameras $cn$.
    Perform *Optimisation Algorithm* (PSO or ARES).
      Apply outcome to $c_i$.
end
```

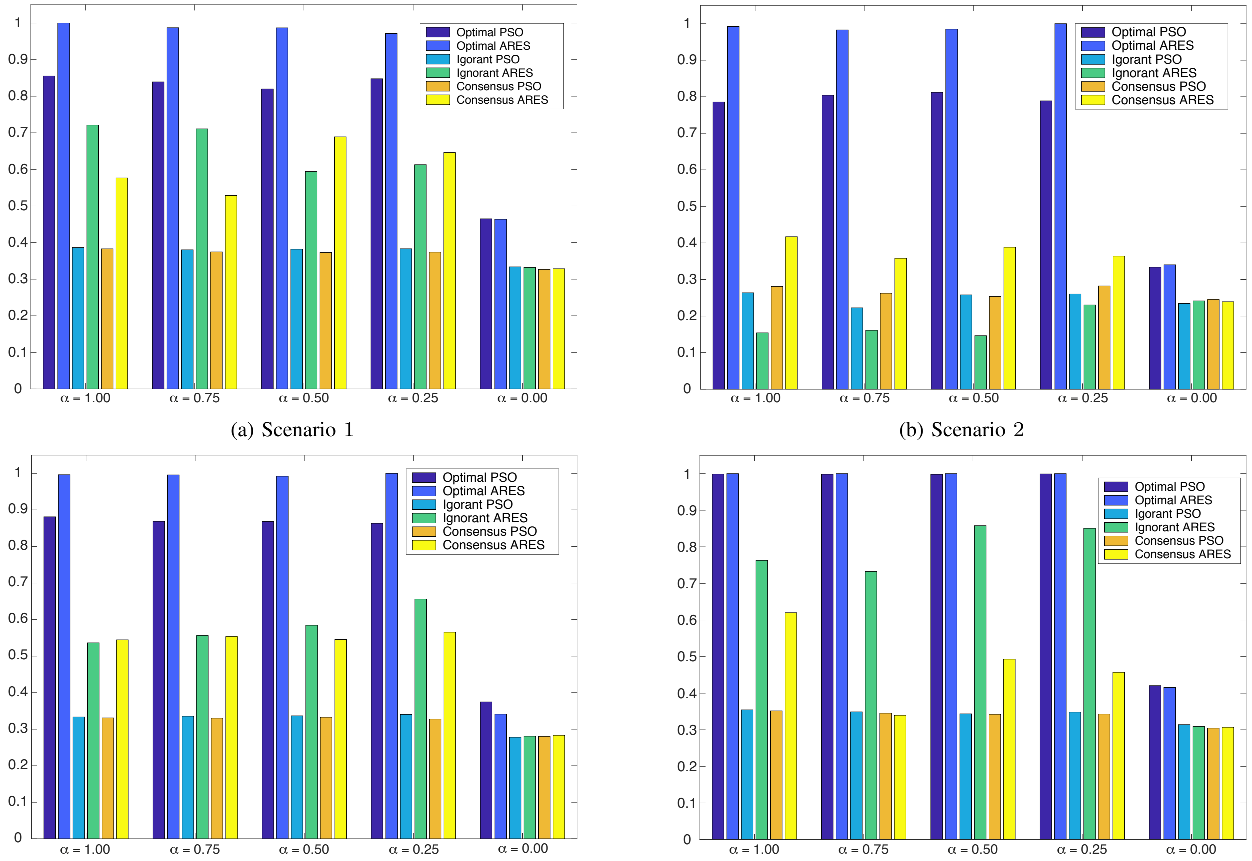
Since cameras operate only on local information and without direct interaction and coordination with other cameras, an individual camera might change its orientation and focal length towards an outcome with a local minima. Additionally, one may analyse the network structure *a priori* and adapt individual cameras in each disconnected graphs (e.g. left and right column in Scenario 3) concurrently. However, the benefits of such an approach remain an open question.

A. Results of Self-organised Coordination

For our self-organising coordination approach, we revert to our initial four scenarios. The average results of the final iteration, normalised by the best outcome per scenario across all α-values, is illustrated in Figure 6. Again, as in the decentralised case, we perform ARES with only 3 copies of the given configuration of the neighbourhood. We iterated through all cameras 30 times when using ARES, while we iterate through the entire network 100 times when using PSO. Furthermore, we present a comparison of the trade-off between the achieved coverage and the amount of unused area of the decentralised and self-organised coverage maximisation approach in Figure 7. The unused area is calculated as given in Equation 2. Coverage and unused area is normalised by the maximum value across all used α-values in a single scenario.

Interestingly, in many situations, the self-organising approaches perform better than their decentralised counterpart. Except for Scenario 2, self-organising approaches reach between $50\% - 75\%$ of the optimal coverage maximisation approach when $\alpha > 0.00$. In a similar fashion, ARES performs about 5% better than PSO. However, the higher processing effort of ARES might make this benefit neglectable.

When looking at the trade-off between coverage and unused area, we can notice that the individual approaches cluster. This suggests, that the α-value has a lower impact on the outcome than the employed approach to identify the maximum coverage *per se*. This theory is supported by the fact that the coverage performance of individual approaches does not vary much when changing the α-value, except when $\alpha = 0.00$. We also note that the self-organising approaches do not as well in Scenario 4, where only three cameras are utilised. In this particular case, a simple decentralised approach outperforms our self-organising approach. We attributed this to the fact, that all cameras are within communication range and hence are aware of each others states. This is equivalent to having



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

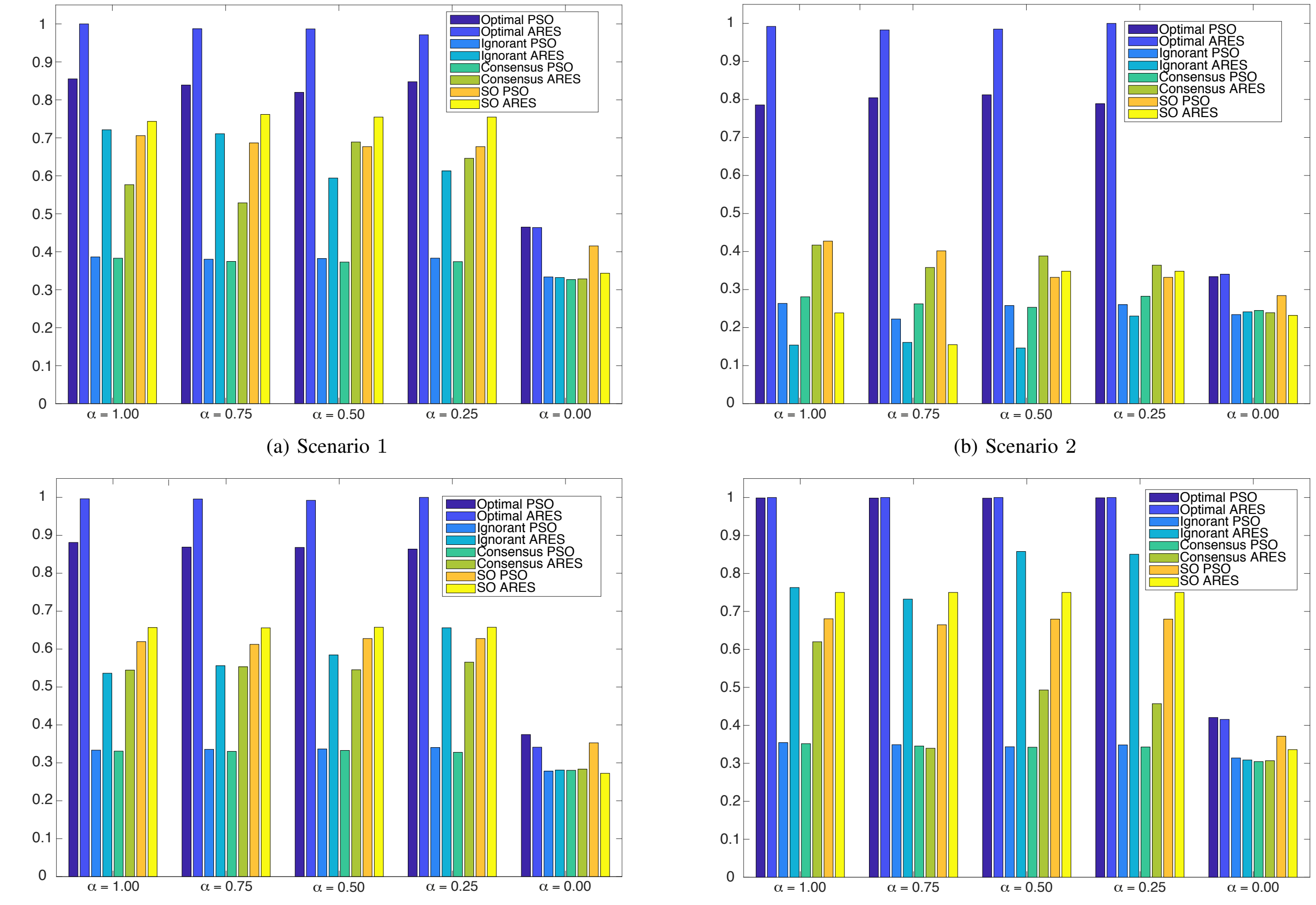
Fig. 5. Comparison of the mean coverage achieved using the centralised *optimal* and decentralised *ignorant* and *consensus* PSO and ARES with different α-values over 30 runs.

global knowledge. This lets us speculate, that there is a relation between the size of the network, the communication range of the individual cameras, and the performance of the respective algorithm. However, this remains an open question for future research.

VII. Discussion and Future Work

In this paper we investigate the performance of centralised, decentralised, and self-organised approaches on coverage maximisation in distributed smart camera networks. In addition to the available centralised approaches, we also present a decentralised as well as a self-organised approach employing ARES and PSO. We showed the benefits of ARES over PSO in a centralised setting. Furthermore, we were able to illustrate that our self-organised approach with limited information is able to outperform a decentralised approach over time. While neither the decentralised nor the self-organised approach is able to achieve a coverage as high as any of the optimal, centralised approaches, they are able to achieve good coverage in the absence of a global controller. With the different approaches presented in this paper, an operator is able to choose the most appropriate one for the given scenario and network. Furthermore, a selection based on given constraints can be made. Such constraints can be based on the general strategy, e.g. maximum coverage or maximum overlap between cameras, as well as based on the available resources, e.g. dynamic runtime adaptation or centralised optimisation.

However, there are benefits and drawbacks for each individual approach. The decentralised and self-organised approaches highly depend on exchanging information about their current state with their neighbours and hence depend on the communication range of the individual camera. If the communication range is equal to the visual range, cameras can interfere with each other without knowing it. Only a communication range which is at least the combined visual range of both potentially interfering cameras will allow them to be aware of an arising interference. Furthermore, while decentralised and self-organised approaches remove the potential bottleneck of a central controller, they introduce an overhead in communication as each camera needs to send their information to multiple other cameras instead of a single system. Finally, self-organising approaches operate over time allowing the system to gradually progress from its initial configuration towards a better one. This corresponds with our findings on dynamic runtime adaptation of behaviour ([23]). While the self-organising approach can be used *a priori*, its strength lies in the runtime adaptation of the network. This allows for dealing with quickly unfolding situations in dynamic environments. In contrast,



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

Fig. 6. Comparison of *optimal* PSO and ARES with different α-values against decentralised *ignorant* PSO/ARES, decentralised *consensus* PSO/ARES, and self-organising (SO) PSO/ARES in terms of achieved coverage normalised by the best result. Mean values over 30 runs are shown.

decentralised and centralised approaches would require the cameras to notice changes in the environment and trigger a corresponding adaptation when necessary.

Our research so far has shown the potential of self-organised coverage optimisation. However, there are different avenues that can be followed for further research. First, this paper focussed on investigating the benefits of decentralised and self-organised approaches in comparison to centralised ones. However, our experimental research showed that the initial configuration as well as the size and density of the network appears to have an impact on the overall performance of the presented approaches. A parametrisation of these factors is on our roadmap for future research allowing us better insight in the potential effects of self-organising approaches on other distributed systems. Second, in our experiments, we did not consider any obstacles so far. Adding opaque elements which occlude and effectively limit the FOV of individual cameras will introduce an additional challenge to the problem. We would expect this to show the benefits of ARES even more pronounced in comparison to PSO. Third, in our presented experiments, our cameras can change their orientations and focal lengths. In order to increase the potential dynamic of the network, we will look into mobile smart cameras able to change their position altogether. Furthermore, we plan to investigate the impact of changes in the network such as failing or added cameras. Finally, our investigation focussed on two optimization approaches. In future work we will investigate further optimization techniques such as ADMM [24] or distributed CSP [25].

ACKNOWLEDGMENT

This work was supported by the SOLOMON project (Self-Organisation and Learning Online in Mobile Observation Networks) funded by the European Union H2020 Programme under grant agreement no 705020.

REFERENCES

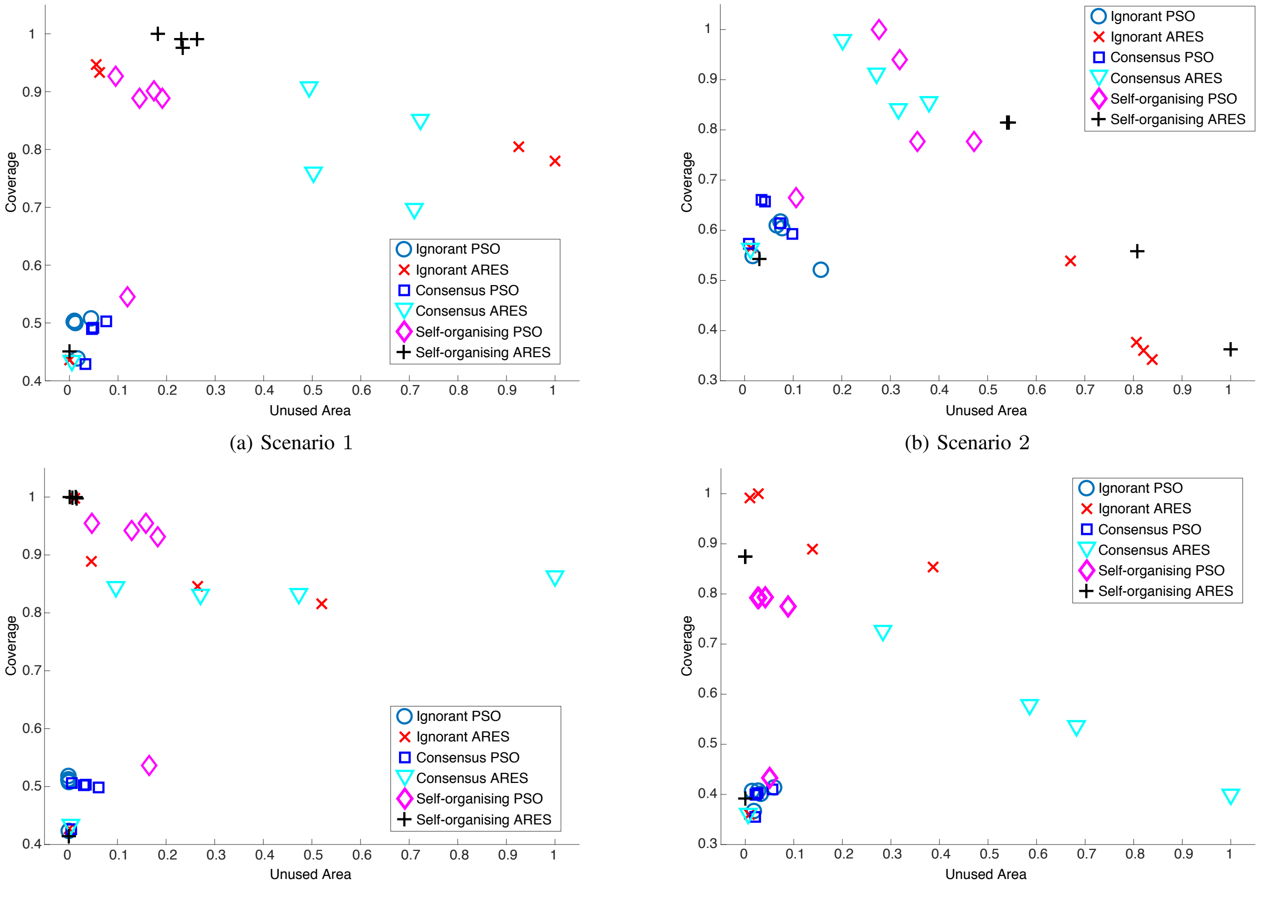
[1] A. Lukina, L. Esterle, C. Hirsch, E. Bartocci, J. Yang, A. Tiwari, S. A. Smolka, and R. Grosu, *ARES: Adaptive Receding-Horizon Synthesis of Optimal Plans*. Springer Berlin Heidelberg, 2017, pp. 286–302.

[2] D. A. Truesdale, "Advantages and disadvantages of decentralized computing services," in *Proc. of the Conf. on User Services*. ACM, 1982, pp. 169–173.

[3] P. Jarzabkowski, "Centralised or decentralised? strategic implications of resource allocation models," *Higher Education Quarterly*, vol. 56, no. 1, pp. 5–32, 2002.

[4] C. Müller-Schloer, "Organic computing: On the feasibility of controlled emergence," in *Proc. of the Int. Conf. on Hardware/Software Codesign and System Synthesis*. ACM, 2004, pp. 2–5.

[5] J. Liu and K. Tsui, "Toward nature-inspired computing," *Communications of the ACM*, vol. 49, no. 10, pp. 59–64, Oct. 2006.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

Fig. 7. Comparison of decentralised and self-organising approaches with different α-values.

[6] H. Schmeck, C. Müller-Schloer, E. Çakar, M. Mnif, and U. Richter, “Adaptivity and self-organization in organic computing systems,” *ACM Trans. on Autonomic and Adaptive Systems*, vol. 5, no. 3, pp. 10:1–10:32, 2010.

[7] A. Omicini and F. Zambonelli, “Challenges of decentralized coordination in large-scale ubicomp systems,” in *Proc. of the Int. Joint Conf. on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1315–1320.

[8] S. Soro and W. Heinzelman, “A survey of visual sensor networks,” *Advances in Multimedia*, vol. 2009, 2009.

[9] C. Piciarelli, L. Esterle, A. Khan, B. Rinner, and G. L. Foresti, “Dynamic reconfiguration in camera networks: A short survey,” *Trans. on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 965–977, 2016.

[10] J. C. SanMiguel, C. Micheloni, K. Shoop, G. L. Foresti, and A. Cavallaro, “Self-reconfigurable smart camera networks,” *Computer*, vol. 47, no. 5, pp. 67–73, 2014.

[11] G. Fusco and H. Gupta, “Selection and orientation of directional sensors for coverage maximization,” in *Proc. of the Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1–9.

[12] S. Rudolph, S. Edenhofer, S. Tomforde, and J. Hähner, “Reinforcement learning for coverage optimization through ptz camera alignment in highly dynamic environments,” in *Proc. of the Int. Conf. on Distributed Smart Cameras*, 2014, pp. 19:1–19:6.

[13] L. Esterle, B. Rinner, and P. R. Lewis, “Self-organising zooms for decentralised redundancy management in visual sensor networks,” in *Proc. of the Int. Conf. on Self-Adaptive and Self-Organizing Systems*, 2015, pp. 41–50.

[14] K. R. Konda, N. Conci, and F. D. Natale, “Global coverage maximization in ptz-camera networks based on visual quality assessment,” *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6317–6332, 2016.

[15] F. G. H. Yap and H.-H. Yen, “Novel visual sensor coverage and deployment in time aware ptz wireless visual sensor networks,” *MDPI Sensors Journal*, vol. 17, no. 1, 2017.

[16] A. A. Altahir, V. S. Asirvadam, N. H. B. Hamid, P. Sebastian, N. B. Saad, R. B. Ibrahim, and S. C. Dass, “Optimizing visual surveillance sensor coverage using dynamic programming,” *IEEE Sensors Journal*, vol. 17, no. 11, pp. 3398–3405, 2017.

[17] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proc. of the Int. Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[18] H. Kahn and T. E. Harris, “Estimation of particle transmission by random sampling,” *National Bureau of Standards applied mathematics series*, vol. 12, pp. 27–30, 1951.

[19] C. Chen and L. Shaw, “On receding horizon feedback control,” *Automatica*, vol. 18, no. 3, pp. 349 – 352, 1982.

[20] L. Esterle, P. R. Lewis, H. Caine, X. Yao, and B. Rinner, “Camsim: A distributed smart camera network simulator,” in *Proc. of the Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops*. IEEE, 2013, pp. 19–20.

[21] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[22] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, vol. 53, no. 1, pp. 65 – 78, 2004.

[23] P. R. Lewis, L. Esterle, A. Chandra, B. Rinner, J. Torresen, and X. Yao, “Static, dynamic, and adaptive heterogeneity in distributed smart camera networks,” *ACM Trans. on Autonomic and Adaptive Systems*, vol. 10, no. 2, pp. 8:1–8:30, 2015.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[25] K. R. Duffy, C. Bordenave, and D. J. Leith, “Decentralized constraint satisfaction,” *IEEE/ACM Trans. on Networking*, vol. 21, no. 4, pp. 1298–1308, Aug 2013.