Received October 2018; | Revised August 2019; | Accepted November 2019

DOI: xxx/xxxx

**ARTICLE TYPE**

# Distributed Autonomy and Trade-offs in Online Multi-object *k*-coverage

Lukas Esterle*[1] | Peter R. Lewis[2]

[1]Department of Engineering, DIGIT, Aarhus University, Aarhus, Denmark
[2]Aston Lab for Intelligent Collectives Engineering (ALICE), Aston University, Birmingham, UK

**Correspondence**
*Corresponding author, Email: lukas.esterle@eng.au.dk

**Summary**

In this paper, we explore the *online multi-object k-coverage* problem in visual sensor networks. This problem combines $k$-coverage and the Cooperative Multi-robot Observation of Multiple Moving Targets problem, and thereby captures key features of rapidly deployed camera networks, including redundancy and team-based tracking of evasive or unpredictable targets. The benefits of using mobile cameras are demonstrated and we explore the balance of autonomy between cameras generating new subgoals, and those responders able to fulfill them. We show that higher performance against global goals is achieved when decisions are delegated to potential responders who treat subgoals as optional, rather than as obligations that override existing goals without question. This is since responders have up-to-date knowledge of their own state and progress towards goals where they are situated, which is typically old or incomplete at locations remote from them. Examining the extent to which approaches over- or under-provision coverage, we find that being well-suited for achieving 1-coverage does not imply good performance at $k$-coverage. Depending on the structure of the environment, the problems of 1-coverage and $k$-coverage are not necessarily aligned: that there is often a trade-off to be made between standard coverage maximisation, and achieving $k$-coverage.

**KEYWORDS:**
distributed k-coverage, CMOMMT, mobile smart cameras, dynamic reconfiguration, distributed coordination

## 1 | INTRODUCTION

The ability of smart cameras to pre-process visual information on site is often exploited, however what is less well understood is how to take advantage of a major benefit of smart cameras: to cooperate effectively and efficiently in networks without central coordination. Recent technological advances enable us to move beyond simple static or PTZ (pan/tilt/zoom) cameras and introduce mobility, enabling them to change their physical location. Here cameras are either worn by humans or mounted on robotic platforms. In combination with dropping prices for cameras, (for example, body worn cameras can be purchased for less than $50), this allows for very large scale mobile smart camera networks. In these large networks of mobile smart cameras, decentralised coordination and distributed autonomy may be desirable, or even a necessity, as cameras can relocate (or be relocated) to areas where communication with a central component might either not be possible, or come with significant delays or bottlenecks, for example due to lack of mobile network signal. Nevertheless, mobile smart cameras allow for i) rapid deployment in unknown environments without existing infrastructure, and ii) flexible behaviour in the face of unforeseen and rapidly unfolding situations. Instead, we assume that cameras can communicate with each other through direct message passing, over an ad-hoc network.

Achieving the above vision in concrete ways will require the tackling of a number of open challenges. In this paper we focus on the challenge of how to achieve high levels of $k$-coverage of a number of target objects, measured online over time, when both cameras and objects can move. For sensor networks in general, $k$-coverage is achieved when a monitored region is covered by at least $k$ sensors [1], and is used as a measure of redundancy in one-shot coverage optimisation problems. When measured over time as targets move, this gives rise to an online version of $k$-coverage, where it is generally not possible to ensure all targets are $k$-covered on an ongoing basis (e.g., as objects disperse or evade sensors, or as sensors fail), and instead we are faced with the objective of maximising the number of targets for which the network achieves $k$-coverage, over time. We call this the problem of *online multi-object k-coverage* [2].

This problem extends the cooperative multi-robot observation of multiple moving targets problem (CMOMMT) [3,4] to consider: (i) the case when the set of objects to cover changes over time, (ii) directional rather than omnidirectional cameras, and most importantly (iii) how to achieve $k$-coverage in this context [1].

In our previous work [2], we introduced this problem, and presented a preliminary study that explored how the behaviour of various simple camera control algorithms performed. However, several important questions arose as a result of that study. Among these, it was first observed that different approaches relied on varying amounts of time and network overhead to achieve their respective solutions. Second, further analysis identified the presence of over-provisioning in some cases: in achieving at least $k$ coverage, it may be that we often end up having many times more than $k$ cameras unnecessarily covering an object. Both of these issues correspond to a potential efficiency trade-off and waste of resources.

While the high-level research question tackled in this paper remains the same as in our initial study [2], in this article we explore a number of new specific questions that arise, and provide deeper insight.

This leads to four sets of new experiments and corresponding results in this article, which are presented alongside our original base result [2]. These new contributions are:

1. A new set of experiments to compare the performance of mobility against cameras able to re-orient their field of view, as an example of a common non-trivial camera network set-up;
2. A new set of experiments to quantify the number of messages sent by each approach, in order to provide insight into the cost-benefit trade-off associated with each approach;
3. A new and deeper analysis of the extent to which the approaches over or under provision cameras, in order to achieve the required level of online multi-object $k$-coverage;
4. A new analysis of the time taken to $k$-cover an object, after it has been identified by the network as *interesting*.

Experimental results concerning additional communication strategies, which were present in the initial study but are not relevant to the above contributions, are not re-presented here. These strategies were, in general, less effective than those explored further in this article. They include techniques that use a learnt vision graph [5] to determine where to send requests. While this has previously been shown to be effective in supporting robustness to changes over time [6] in a resource-efficient way [7], it proved to be less effective at determining how to direct requests for help in a mobile-camera context [2]. Conversely, the family of approaches that use such a graph to determine instead how to *respond* to requests, do prove effective in some cases. Thus, a deeper analysis of these is included in this article, revealing that they trade off time taken to reach high $k$-coverage for a lower communication overhead.

In summary, the key contributions of this article are i) to provide new insight into the overheads (in terms of time, communication, and wastefulness) associated with effective approaches for achieving online multi-object $k$-coverage; and ii) comparing mobility-based approaches against pan-tilt-zoom, showing where static PTZ cameras are sufficient, and where mobility adds value. The experimental simulation study presented in this paper demonstrates that the addition of mobility and multi-camera coordination improve online multi-object $k$-coverage by a network of mobile cameras, in a variety of different scenarios. In other cases, the problems of 1-coverage and $k$-coverage are sufficiently aligned that simpler approaches (e.g., using PTZ cameras) can be adequate. We use archetypal scenarios to illustrate these cases.

Further, we consider the most effective place in the network for decisions to be made concerning how to best achieve a local sub-goal. Specifically, we explore decentralised coordination approaches where decision-making is made by cameras that *request* the help of other nearby cameras, or by those that are candidates to *respond*. While the network as a whole knows the current state of $k$-coverage and how well-placed individual cameras are to change their behaviours to improve it, this knowledge is distributed, and not available in any single place where a decision based on it could be made.

Hence, we analyse our results through the lens of distributed goal-awareness, asking questions concerning where the balance of autonomy should lie between nodes generating and issuing new subgoals, and those able to fulfill them. Our results show that when coordinating $k$-coverage in a distributed way, higher performance against global goals is achieved when decisions are delegated to potential responders who treat subgoals as optional, rather than when new goals are treated as obligations that override existing goals without question. This is since responders have up-to-date knowledge of their own state and the state of progress towards goals where they are situated, which is typically out-of-date or incomplete at locations remote from them. Put simply, the responding camera is the best place to decide which object to cover.

The implication of these results is that the design of effective coordination mechanisms, for contexts where goals change rapidly, should be based on decisions being made most locally to where they will be enacted: accounting for distributed goal-awareness through local decision making leads to higher overall goal satisfaction.

## 2 | PROBLEM STATEMENT

In this work we investigate the problem of achieving and maintaining desired levels of $k$-coverage of multiple mobile objects, over a period of time. This is the problem of *online multi-objective k-coverage*[2]. This problem extends the cooperative multi-robot observation of multiple moving targets problem (CMOMMT)[3,4] to consider: (i) the case when the set of objects to cover changes over time, (ii) directional rather than omnidirectional cameras, and most importantly (iii) how to achieve $k$-coverage in this context[1]. Parker and Emmons established[3] the NP-hard nature of the CMOMMT problem, to which *online multi-object k-coverage* adds additional complexity. The problem of *online multi-objective k-coverage* can be stated as follows.

Consider a set of $n$ mobile cameras $C = \{c_1, c_2, ..., c_n\}$ and a set of objects $O = \{o_1, o_2, ..., o_m\}$. In the 2D version of the problem considered in this article, each object and each camera $i$ has a 2-dimensional location denoted by their absolute positions in the coordinate system $x_i, y_i$, we also refer to this as $\vec{x}_i = (x_i, y_i)$. We use $\vec{x}_i(t)$ to refer to the position of camera/object $i$ at time step $t$. One could extend this to the 3D case, using similar notation. Each camera $c_i$ can move with a velocity $\vec{s}_i$ and rotate with an angular velocity $u_i$. Furthermore, cameras communicate via message passing.

$P \subseteq O$ denotes a subset of all known objects, where objects *of interest* are denoted $p_1, p_2, ..., p_l \in P$. Objects in $O$ can become *interesting* and *uninteresting* to the network at any time. One can think of this process being driven by an operator or a specific camera identifying targets that matter, thus rendering an object *interesting* (in $P$) or *uninteresting* (in $O$ but not in $P$). Importantly, this means that $P$ and $O$ are subject to change over time. Therefore, $O$ and $P$ at each point in time $t$ may be denoted as $O_t$ and $P_t$.

Each camera $c_i$ has its own field of view (FoV) $f_i$, which we here model as a cone and define by a range $r_i$, an angle $\omega_i$ defining the viewing orientation relative to a fixed reference point, and an angle $\beta_i$ defining the width on either side of $\omega_i$. The range of a camera is limited by the distance at which an object can be detected and identified on-board the camera. Therefore a camera's state is defined as $c_i = \langle \vec{x}_i, \vec{s}_i, u_i, \omega_i, r_i, \beta_i \rangle$. This defines a snapshot at a particular point in time and can be further indexed by $t$ to represent the camera's state over time. Specifically, the discrete-time behaviour of a camera $c_i$ can be defined as

$$\begin{aligned} \vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{s}_i(t) \\ \omega_i(t+1) &= \omega_i(t) + u_i(t) \end{aligned} \tag{1}$$

The velocities $\vec{s}_i$ and $u_i$ are controlled by an internal agent at each time $t$ with the aim of achieving the current objective (e.g., follow object, move to cover object, move back to original location).

An object $o_a$ is then *covered* at a given time $t$, if the object is geometrically within $f_i$:

$$cov(o_a, f_i, t) = \begin{cases} 1, \text{ if } d_{i,a} \le r_i \ \& \ |\alpha_{i,a}| \le |\beta_i| \\ 0, \text{ otherwise,} \end{cases}$$

where $d_{i,a}$ is the Euclidean distance and $\alpha_{i,a}$ is the angle between the object $o_a$ and the camera $c_i$. This is illustrated in Figure 1.

Further, an object $o_a$ is $k$-covered, for a given value of $k$, at time $t$, according to the following:

$$kcov(o_a, k, t) = \begin{cases} 1, \ if \ \sum_{i=1}^{n} cov(o_a, f_i, t) \ge k \\ 0, \ otherwhise. \end{cases}$$

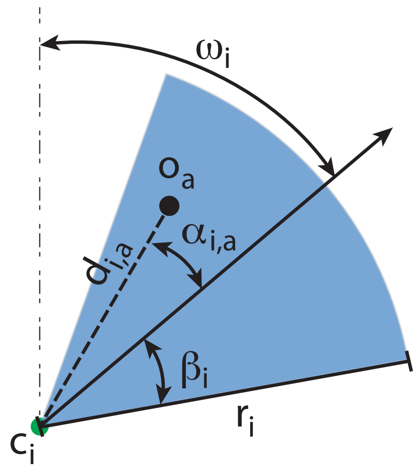



**FIGURE 1** Illustration of an object in a camera's FoV. The FoV is illustrated in blue with a range of $r_i$, an orientation of $\omega_i$, and an angle $\beta_i$ on both sides of $\omega_i$. The object has an angle $\alpha_{i,a}$ to the camera's orientation and a distance $d_{i,a}$.

In the scenarios studied in this paper, each object has a 5% chance of becoming *of interest* at any time step, and then remains *of interest* for a random number of time steps drawn from the uniform distribution [5, 100].

For a given value of $k$, provided by an operator and known to all cameras, the aim is to maximise the *online multi-object $k$-coverage* over time:

$$\sum_{t=1}^{T}\sum_{a=1}^{m} kcov(o_a, k, t)$$

where $T$ represents a finite time horizon of interest. For comparability, in this paper these values are presented as a fraction of the theoretical upper limit on this value in each scenario.

We can therefore define a normalised metric of how well online multi-object $k$-coverage is achieved as:

$$OMC_k = \sum_{t=1}^{T} \frac{\sum_{a=1}^{m} kcov(o_a, k, t)}{|P_t|} \tag{2}$$

for a given value of $k$.

The $OMC_k$ metric captures the central aspect of solution quality in the online multi-object $k$-coverage problem, in a comparable way. Namely, what proportion of a dynamic set of targets have been $k$-covered, over time, for a specific $k$ value. The structure of the metric is similar to that of an episodic reinforcement learning task.

However, the $OMC_k$ metric does not capture other important features of the behaviour of the system, including crucially, the associated overhead. In our analyses, we are also interested to explore how achieving online multi-object $k$-coverage trades off against these factors. Overhead includes number of messages required to coordinate, as well as the overall movement required by cameras, both of which correspond to increased energy usage. Thus, while $OMC_k$ is the primary metric used throughout this study, we analyse this in the context of additional metrics recording the number of messages sent and physical movement required. These complement the $OMC_k$ results and are presented in Section 6.

Finally while $OMC_k$ provides a valuable headline figure, it does not give insight into the distribution of achieved $k$ values by different coordination approaches. Analysis of these distributions is important, since it tells us whether techniques achieve $k$-coverage by regularly over-provisioning, which can be seen as wasteful, or if they are effective at accurately targeting exactly $k$ cameras per object. Where $k$-coverage is not achieved, distributions also provide insight into whether the inability to provision $k$ cameras for each object is typical behaviour, or if a smaller number of objects are ignored, thus reducing the mean-based $OMC_k$ metric. Thus, Section 6 also presents distributional results.

## 3 | RELATED WORK

The presented work extends the Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) which was introduced as an NP-hard problem by Parker and Emmons[3]. Their approach to tackling it was to generate artificial forcefields for each object of interest which attracts individual robots to targets. Werger and Matarić[8] extend this towards W-CMOMMT (weighted CMOMMT) giving a weight to each target. They use BLE (Broadcast of Local Eligibility) to directly coordinate tasks among the robots. Jung and Sukhatme[9] use learnt densities of sensors and targets to steer individual robots to insufficiently covered areas. This essentially leads to higher coverage of the available targets. Kollin and Carpin[10] perform target loss prediction to decide on when to call for help. They use broadcasting in order to ensure continuous 1-coverage of different objects. However, their main concern is to maximise overall coverage rather than covering objects with multiple sensors at once.

Covering objects with multiple sensors has received quite some attention as $k$-coverage in sensor networks. The idea is to have redundant measurements of a specific location[1]. This redundancy allows for resource-constrained sensor networks to cover areas better[11,12] or for sleep-cycles that prolong network lifetime[13]. Liu et al.[14] specifically define directional $k$-coverage for visual sensors and discuss the benefits of multiple perspectives on a target provided by $k$-coverage in camera networks. Although we do not explicitly consider the value of multiple perspectives in this study from a tracking perspective, this value can be enhanced by more deliberate coordination, such as triangulation[11]. Hefeeda and Bagheri[15] try to approximate optimal $k$-coverage in a network using a distributed approach within the network of sensors. Elhoseny et al.[16] introduced mobile nodes to cover known and stationary targets with $k$ sensors. In order to optimise the coverage, they use an evolutionary technique. Liu et al.[17] further analyse the benefits of moving sensors to detect and cover specific, but unknown, stationary points in the environment. Li and Kao[18] use Voronoi diagrams to estimate the minimum number of nodes and minimise sensor node travel required to achieve $k$-coverage.

Fusco and Gupta[19] explore the ability of a simple greedy approach to optimally place and orient directed sensors for $k$-coverage of static objects in the environment. Micheloni et al.[20] identify activity density maps to determine areas highly frequented by target objects, and use an expectation-maximization process to define optimal orientations of PTZ cameras. CMOMMT and coverage optimisation in camera networks have been researched quite intensively[21,22,4]. However, the problem of $k$-coverage with an unknown number of targets using mobile smart cameras has received little interest to date.

Many approaches to the control of sensor networks base decisions on consensus formation[23,24], for example through belief propagation. In that family of model-building approaches, questions of scalability are paramount[25,26], as consensus needs to be reached in near real-time, where possible. In our work, we do not add a consensus layer, instead exploring what can be achieved with a behaviour-based approach to cooperation[27], based on a *request-response* mechanism, in which the number of messages scales at worst linearly with the number of either targets and cameras.

In distributed systems such as this, a question arises concerning the various loci of control, typically operationalised through agents in a multi-agent system[28]. This has also been studied in systems where autonomy can be delegated to different people within an organisation. For example, it is tempting to consider that decisions in the military are made within strict chains of command. However, in practice, it is often far more effective for decisions to be made locally and quickly, when there is information disparity within different parts the system[29]. This means that authority is granted according to expertise, and hence control resides with the entity that has the most relevant and up-to-date knowledge with respect to the consequences[1] of the decision. For example in naval fleets, ships are often geographically highly dispersed. As a consequence, chains of command can not be executed sufficiently quickly due to information propagation delays. McCulloch[29] claims that in order to respond to rapidly changing situations, ship captains can disengage from the chain of command in order to respond faster and more effectively, given their more relevant knowledge. A second example is how different loci of control are used in the human central nervous system[30]. While control is hierarchical in nature, again we often see action initiated where relevant information is available first.

In systems with distributed autonomy, architecture can also be important. Balasuriya et al.[31] explore this in the context of Autonomous Underwater Vehicles (AUVs). AUVs inherently require a higher degree of autonomy of each individual node as communication bandwidth is drastically limited. In our work, we do not consider the overall architecture of the sensor network, rather letting this emerge from their interactions. However, since our results concern the distribution of decision making and loci of control, they could also be relevant in determining effective architectures based on the distribution of knowledge about the network. This is discussed in Section 7.

[1] Other brands of ethics are available.

Related to decision-making architecture is the question of spatial structure. In many areas of distributed robotics, an important task is the preservation of a multi-robot formation, as the robots move and are subject to potential failures[32]. Distributed consensus formation is also an approach employed here[33].

There have been a number of studies on the *online multi-object k coverage* problem since it was introduced[2]. CHAINMAIL[34] introduces a specific, gossiping-based task assignment algorithm for the online multi-object $k$-coverage problem. While the approach achieves good results, it relies heavily on the benevolence and cooperation of other agents in the environment. Other approaches investigate ideas around introducing networked self-awareness[35] to agents, allowing them to learn and understand the behaviour of other agents in their environment[36]. While the agents only use snapshot information, they can use this knowledge to make informed decisions on the behaviour of others and the potentially best action for themselves. As an alternative to a set of autonomous smart cameras controlled by individual software agents, coordinating them in teams has also been proposed. The main idea is to have a team observing the environment while the other team specifically follows objects. This reduces the clustering of agents in specific areas and improves the overall $k$-coverage. Static teams in combination with entropy signalling to attract and limit the number of observing agents to $k$ per object[37] allows good overall $k$-coverage. However, a question remains on how to select the number of team members and their location before deployment and without *a priori* knowledge about the scenario. Dynamic teams can overcome this limitation[38]. Here, each autonomous smart camera can decide during runtime if it is better to remain in its current team or change to the other in order to improve the overall, network-wide task of increasing $k$-coverage. The individual cameras rely on local information to make this decision during runtime.

## 4 | BASELINE APPROACHES AND RESULTS

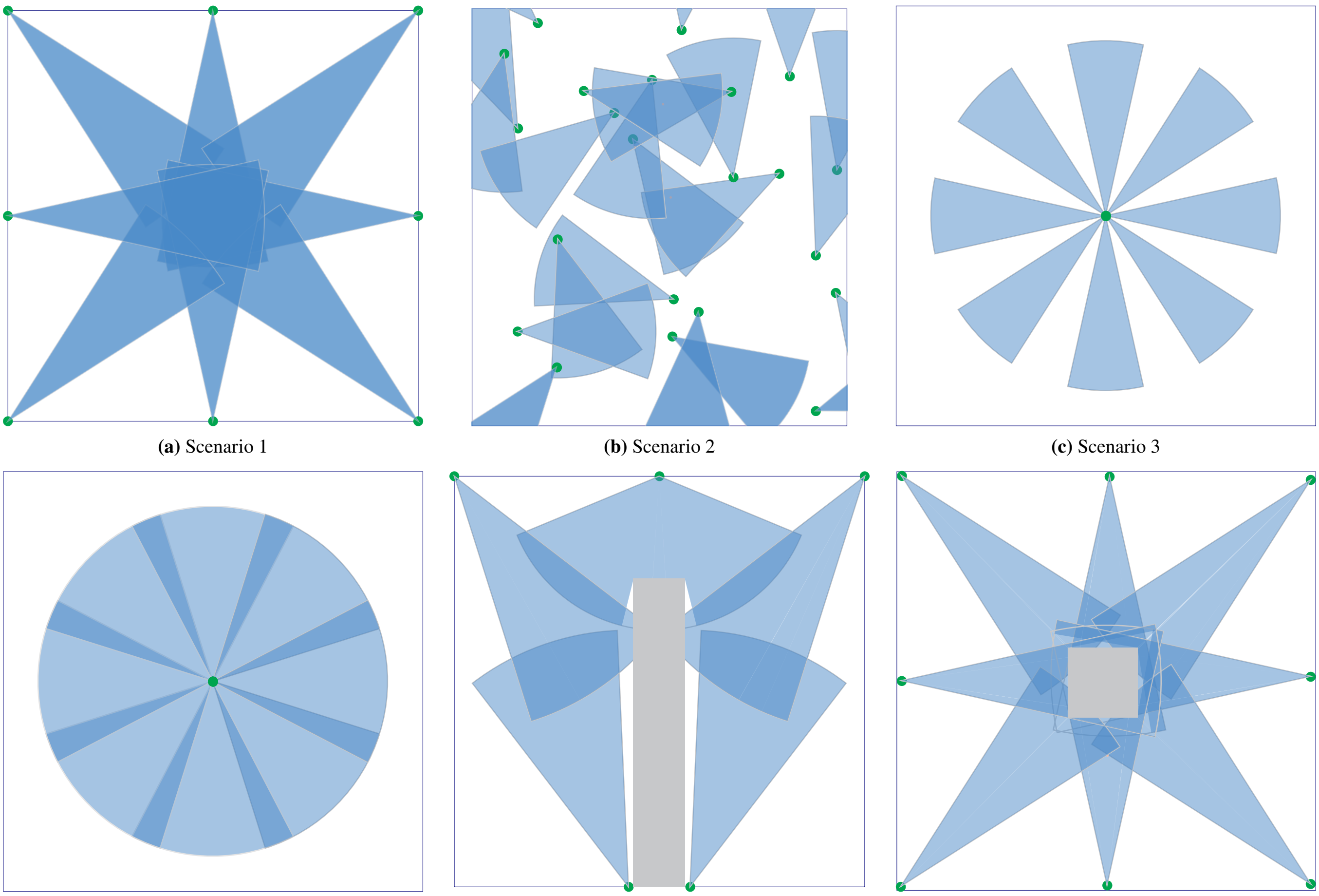
First, we are interested in establishing the levels of online multi-object $k$-coverage achievable by fully distributed autonomy, where there is no coordination between cameras. This forms a baseline for comparison against later approaches, where inter-camera communication is used as a basis for decision-making and online learning. There are two sources of change that the cameras must adapt to, over time. First, the set of objects to cover may change in its membership, and second, the physical positions of objects change.

To evaluate decentralised coordination approaches in the context of these dynamics, we constructed six qualitatively different scenarios in the CamSim[39] smart camera network simulator. These are depicted in Figure 2. These scenarios are designed to capture a number of different common features of smart camera deployments, including: wall-based installation in a room with no occlusions (Scenario 1); the same with occlusions (Scenario 5 and 6); randomised placement such as from an air drop or group of autonomous robots or people (Scenario 2); differently-oriented cameras installed on poles, turrets, or vehicles, without overlapping FOVs (Scenario 3) and with overlaps (Scenario 4). These features pose a range of challenges, which impact on the behaviour of coordination approaches in different ways. These are discussed as they arise through the discussion of results. Objects and cameras move in a straight line according to a random vector, bouncing back in a random fashion upon reaching the boundary. Cameras can move as fast as objects and can turn fast enough to keep passing objects within their FoV.

First we explore intuitive baseline decision-making approaches, in which behaviour is not coordinated through communication. Three of these were presented in our previous work[2]; here we also include PTZ, where cameras can change their orientation but not their position. We therefore present four intuitive baseline approaches for comparison.

1. *Static*: Each camera has an assigned fixed location and orientation (see Figure 2).
2. *Random movement*: Cameras have a starting location and move with a random vector. Their orientation changes by bouncing off from the boundary of the environment.
3. *Random and following*: As *random movement* but changes velocity in order to follow *interesting* objects.
4. *PTZ*: Each camera has a fixed location corresponding to its start position. However, each camera can change its orientation similarly[2] to a pan-tilt-zoom camera (PTZ) in order to provision an object.
5. *Centralised*: Due to the hardness of the problem, finding an exact global solution is infeasible within the decision time required for a camera (typically less than a couple of seconds). However, it is helpful to compare the behaviour of decentralised coordination schemes against a feasible and reasonable centralised heuristic. Therefore, we also include results for

[2] We do not model zooms in this study.



**(a)** Scenario 1 **(b)** Scenario 2 **(c)** Scenario 3

**(d)** Scenario 4 **(e)** Scenario 5 **(f)** Scenario 6

**FIGURE 2** Evaluated scenarios. Green dots represent cameras and blue cones their respective FoVs. Gray blocks illustrate opaque walls/areas. The studied scenarios are designed to capture a number of different common features of smart camera deployments, specifically: wall-based installation in a room with no occlusions (Scenario 1); the same with occlusions (Scenario 5 and 6); randomised placement such as from an air drop or group of autonomous robots or people (Scenario 2); differently-oriented cameras installed on poles, turrets, or vehicles, without overlapping FOVs (Scenario 3) and with overlaps (Scenario 4).

a 'greedy' centralised approach, in which we simply assign the closest $k$ cameras to each *interesting* object in each time step. Here, closest relates to the objects' distance from to the centre of the FOV of a camera. This means that a camera will track an object within its FOV that is further away than an object that is not within the FOV but close-by (e.g. right behind the camera). Since decisions are offloaded to a central decision-maker, this requires each camera to send its state to the central node at each time step, and receive instructions in return.

Figure 3 shows results for these four approaches. All graphs show mean results over 30 independent runs, while error bars represent one standard deviation. $T = 1000$ time steps, and $k = 3$. All results have been normalised by the maximum number of objects detected to be *interesting* by the network throughout the entire experiment in each scenario, respectively.

It is clear that using *static* typically achieves low online multi-object $k$-coverage over time, as expected, and only at all when there is an initial overlap in the camera layout. It is also clear that in almost all cases, a centralised approach outperforms the distributed approaches. Further, while *static* can achieve up to 50% 1-coverage, this is also highly dependent on the initial layout of the camera network. Pure *random movement* also performs poorly, since cameras do not follow important objects, thus $k$-coverage happens only by accident for limited periods of time. *Random and following* typically performs better at online multi-object $k$-coverage over time, among the moving approaches, as well as at achieving base 1-coverage (i.e. $k = 1$) of important objects. Depending on the scenario, this can reach up to 65% 1-coverage. Interestingly, *PTZ* performs best in achieving online multi-object $k$-coverage (where $k \geq 3$) over time, among all distributed approaches. This is due to the approach preserving the structure of the camera network layout, which is otherwise lost with the uncoordinated movement. In some scenarios, *PTZ* achieves up to 40% of the respective theoretical maximal value for online multi-object $k$-coverage over time. The *centralised* approach outperforms almost all distributed approaches when it comes to $k$-coverage. Nevertheless, the globally controlled cameras are often outperformed in 1-coverage as all cameras are focussed on the covering specific objects.

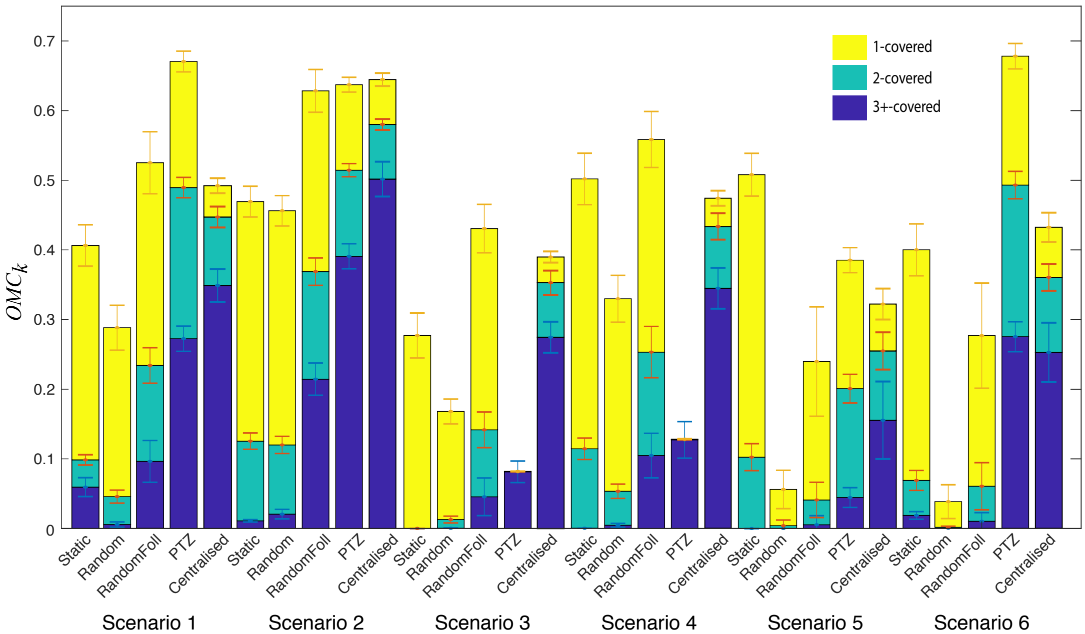



**FIGURE 3** Illustration of baseline approaches: *fixed locations*, *random movement*, *random movement and following*, and *PTZ* for all 6 scenarios. The top yellow bar represents 1-coverage, second green bar illustrates 2-coverage, and bottom blue bars represent objects being at least $k$-covered (here $k = 3$). Results have been normalised by the number of objects detected to be *interesting* by the network in each scenario, respectively.

## 5 | DECENTRALISED COORDINATION APPROACHES

We next investigate a number of decentralised inter-camera coordination schemes for online multi-object $k$-coverage. These are based on and explore the principle of behaviour-based cooperation[27], thus the agents controlling cameras do not maintain state or model each others' beliefs. The approaches studied here centre on the idea of cameras notifying others of *interesting* objects within their FoV, sending a "call for help". Upon receiving this, a camera decides how to react, based on its local information. Cameras use message passing to communicate. While we could consider simply broadcasting all information to all cameras, such a number of requests would be prohibitively expensive to process on-board resource-constrained cameras. The outline coordination algorithm is described in Algorithm 1, from which two important questions arise:

1. To avoid broadcasting to all other cameras, how should individual cameras focus their communication efforts (i.e., on the most relevant recipients)?

2. How should a camera receiving such a "call for help" react? I.e., when and where should it move, or how should it issue follow-on communications?

**Algorithm 1:** Query-based dynamic $k$-coverage algorithm

```
foreach c_i ∈ C at time t do
    foreach o_a ∈ P do
        if cov(o_a, f_i, t) then
            Notify other cameras as described in Section 5.1
            Adjust camera to cover object o_a (Equation 1)
        else
            if Received notification then
                React to notification (Section 5.2)
            else
                Baseline behaviour (Section 4)
            end
        end
    end
end
```

## 5.1 | Communication Strategies

In addition to a simple broadcast behaviour, we evaluate three strategies for the targeting of inter-camera messages: $k$-CLOSEST relies on the Euclidean distance between cameras, notifying only the $k-1$ closest cameras to the camera's own location. $k$-FURTHEST notifies the $k-1$ furthest cameras from the notifying camera. Finally, $k$-RANDOM does not rely on the distance between cameras, but communicates with a random set of cameras in the network.

Both $k$-FURTHEST and $k$-CLOSEST are implemented in CamSim such that they use global information about the location of individual cameras. More realistically, $k$-CLOSEST could be interpreted as using short-range wireless technology. However, it is not necessary to explore the implementation feasibility of this further, as the results reported in this paper show that, in any case, communicating purely based on distance can be improved upon using other methods, as described in the next section.

In our previous work[2], we also investigated the benefits of cameras learning neighbourhood relationships based on commonly observed objects. Each camera would locally create a pheromone inspired graph representing neighbourhood relationships[5]. However, it turned out that using this approach to focus communication has little benefit for the network.

## 5.2 | Response Strategies

When receiving to a request, the camera could simply oblige, and move to cover the object of interest. However, this might lead to over-provisioning on individual objects as well as losing objects currently covered by some cameras. In addition, this might lead to cameras constantly trying to cover different objects, getting stuck in-between both of them, as they always follow the most recent request. Therefore, in addition to this basic strategy (termed *Newest-Nearest*) we devised three additional response strategies:

1. *Newest-Nearest* (NN): The attempts to cover to the most recently requested object from another camera. If there are multiple requests, it will choose the nearest.

2. *Available* (AV): The camera will use the *Newest-Nearest* response strategy if and only if the camera is currently not occupied by covering/following a different object.

3. *Graph* (GR): The camera learns neighbourhood relationships based on commonly observed objects over time[2] and represents it locally in a graph structure. The camera uses this graph to make a decision which camera's request to follow. A camera will attempt to cover the object requested by the camera with the strongest link. As with *Available*, if the camera is occupied, it will instead continue to cover its current object.

4. *Received calls* (RE): In contrast to the other response strategies, this one considers currently covering cameras for a given object. Here, the camera will provision the object with the least requests as this corresponds to a small number of cameras currently observing this object. As before, this is only done, if the camera is currently available.

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | | Scenario 5 | | Scenario 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $k \geq 3$ | $k = 1$ | $k \geq 3$ | $k = 1$ | $k \geq 3$ | $k = 1$ | $k \geq 3$ | $k = 1$ | $k \geq 3$ | $k = 1$ | $k \geq 3$ | $k = 1$ |
| All Distributed | 0.7806 | 1.3622 | 0.7792 | 0.9885 | 0.7309 | 1.1040 | 0.7779 | 1.1790 | 1.0547 | 1.5758 | 1.0891 | 1.5681 |
| Coordinated Distributed | 0.7169 | 0.9343 | 0.6427 | 0.9594 | 0.7309 | 1.0576 | 0.7779 | 1.0946 | 1.0547 | 0.9116 | 0.8040 | 0.7667 |

**TABLE 1** Normalised performance comparison of best (coordinated) approach to the centralised approach in each scenario. Performance of 1 is the achieved performance of the centralised approach. Clearly, distributed approaches can be a good alternative to a centralised approach when 1-coverage is the only requirement. In $k$-coverage, alternative approaches achieve at least 70% of the performance of the centralised approach. If values are greater than 1, the (coordinated) distributed approach performs better than the centralised approach.

## 6 | RESULTS

We analyse the combination of the above-described communication strategies and response models using the scenarios illustrated in Figure 2. All experimental runs were repeated 30 times and the mean results and corresponding standard deviations are shown. Based on its performance with respect to $k$-coverage in the baseline experiments reported in Section 4, and its potential for full mobility, *random with following* was used as the default behaviour a camera reverts to when no notifications are received. Figure 4 shows the achieved network-wide normalised multi-object $k$-coverage, $OMC_k$ (see Equation 2), using the different approaches. In all cases, $k$ was set at 3, however for comparison, results are shown for how well the technique also achieved 1-coverage, 2-coverage, as well as $k(\geq 3)$-coverage.

When comparing against the results achieved when only using baseline behaviour, it is very clear that adding either communication or mobility, or both, to the baseline behaviour can increase online multi-object $k$-coverage. While *PTZ* is able to achieve good performance in terms of $k$-coverage, this is highly dependent on the scenario. However, throughout all experiments, our coordinated approaches perform better than any of the remaining non-coordinating approaches. Even counter-intuitive communication strategies, such as $k$-FURTHEST, achieve better results than no communication, indicating the benefit of communication per se. While only in a few cases, coordinated distributed approaches can outperform the centralised approach, we are particularly interested in how close we can get to the centralised approach while minimising communication efforts. This direct comparison is presented in Table 1 and normalised by the performance of the centralised approach (i.e. centralised is always 1). It is apparent that the centralised approach is often outperformed by baseline or coordinated distributed approaches when only 1-coverage is considered. However, coordinated distributed approaches still achieve almost always 90% of the performance of the centralised approach. We attribute this to the initial position of the cameras in these scenarios and the fact that they do not move for in the baseline approaches. It is also clear, that a centralised approach often performs better than any other approach when it comes to $k \geq 3$-coverage. However, distributed approaches still manage to achieve around 70% of the performance of the centralised approach. Interestingly, the centralised approach is clearly outperformed in scenario 5 and 6 for $k \geq 3$-coverage. We attribute this to the obstacles in the scenarios and the fact that the centralised approach would use the Euclidean distance to assign objects. This can lead to objects being assigned to cameras that are located on the other side of a wall and hence not directly accessible by the camera.

With respect to communication strategies, the performance of *k-Random* communication seems counter-intuitive at first. While this approach only communicates with random cameras, rather than those that might be close by, it still achieves very high $k$-coverage. This is due to the fact that *k-Random* communicates only with $k-1$ other cameras per time step, sampled randomly each time. Eventually, this leads to communications with all cameras. However, cameras are only requested gradually, leaving the rest of the cameras available to provision other calls. *Broadcast*, in comparison, communicates with all cameras at once, making them effectively unavailable to other calls. The *centralised* approach requires the cameras and the controller to exchange a single message. Here the camera updates the controller with its local information while the controller updates the camera on what actions to take. This results in a high overall communication effort within this approach.

It is important to note that the environment and the initial camera layout may enforce a constraint such that significant overlaps cannot be achieved without mobility. This is especially visible in Scenario 3. However, through mobility in terms of rotation, *PTZ* enables a good view of the entire area, leading to a high 1-coverage result in all performed experiments. However, if there are minimal overlaps, as in Scenario 3, 4 and 5, *PTZ* $k$-coverage results are also low. Conversely, the mobility approaches enable the cameras to achieve greater overlaps and thus a better $k$-coverage result, but this comes at the expense of loosing visibility of the broader scene and therefore a low 1-coverage result. Importantly, we are focussing on achieving $k$-coverage where $k > 1$. Often, as illustrated in scenarios 1 and 2, the $k$-coverage and 1-coverage problems are aligned in terms of how to solve them. Thus, an approach which performs well at 1-coverage also performs well at $k$-coverage but relies on the luck of the initial layout to create overlaps. Where the environment or the initial layout do not allow for significant overlaps, the problems of 1-coverage and $k$-coverage are not aligned, and a choice among the two of them is forced. Approaches using both mobility and coordination among the cameras can generate intentional overlaps and hence improve $k$-coverage at the expense of the number of detected objects and 1-coverage. In contrast, non-coordinated, mobility-based approaches (i.e. *random* and *random and following*) generate high 1-coverage, but only accidental overlaps and hence only low $k$-coverage where $k > 1$.

In Figure 5, we present the distribution of the achieved level of $k$ at time step 500, which enables us to observe over- and under-provisioning of cameras to objects. We focus here on three scenarios with qualitatively different characteristics to their results: Scenario 1, in which *PTZ* performs well and the problems of 1-coverage and $k$-coverage are aligned; scenario 3, in which *PTZ* performs poorly and the approaches that combine mobility and coordination achieve higher performance; and scenario 5, in which the problems of 1-coverage and $k$-coverage are not aligned, and a choice is forced between them. As we are looking for online multi-object $k$-coverage where $k = 3$, we highlighted those results with a red border. Interestingly, we can clearly see how immobile approaches barely over-provision objects. However, they also often under-provision them. Furthermore, we notice how the response model RE achieves very little under-provisioning with any communication strategy. Little surprising, the centralised approach achieves very good $k$-coverage in comparison to any other approach at a specific time step. It is apparent that the approach does little over- and under-provisioning when looking at this snapshot.

We next turn to the question of how long it takes cameras to respond to a request, and to actually cover a given object. Table 2 shows the mean times for this, by approach. It becomes apparent that for non-coordinated approaches, *PTZ* attends quickest, while *Random* and *Random following* require longest. We speculate this is due to the dispersion of the cameras across the scenario while PTZ clusters quickly and hence cover all visible objects, even though this is a low number, very quickly. This is especially visible in Scenario 3 and 4 where PTZ even outperforms any other approach that coordinates. For the coordination-based approaches, BC performs best in terms of fast provisioning of objects. Centralised, as a baseline approach, performs similarly well as other coordinated approaches but is still outperformed by BC. One might have expected CL to provision faster, however this only contacts the closest cameras, independently of their current occupancy. Conversely, BC contacts all cameras, those both currently provisioning objects and unoccupied cameras. In terms of duration until objects are provisioned, we can also observe that GR requires longest to attend and cover an object. We attribute this again to how cameras are requested.

We are also interested in the overhead associated with each approach, in terms of the number of messages sent. Table 3 we give an overview of exchanged messages across the entire network averaged over 30 runs. This table shows almost an opposite result to that in Table 2; we can clearly see that approaches using the *broadcast* advertising scheme exchange the most messages in the network. In contrast, *Furthest*, *Closest*, and *Random* require fewer messages to be exchanged.

Finally, Figure 6 shows the trade-off (again in terms of mean results) between higher movement and $OMC_k$, for $k = 3$. The benefit of *Available* (AV) as well as the drawback of higher movement when using *Received Calls* (RE) becomes apparent. Again, the importance of the response model over the actual communication strategy become evident. The response model *Available* usually requires only about 40% of the movement in comparison to *Received calls* while both of them achieve about the same average coverage (between 45% and 55%). The trade-off becomes more pronounced when the scenario contains obstacles, as these may lengthen the routes for cameras to travel, when moving towards objects to cover.

## 7 | DISCUSSION

In this paper we analysed a number of decentralised techniques for achieving online multi-object $k$-coverge in visual sensor networks, in particular with regard to questions of overhead and efficiency. The results presented show that the addition of mobility and multi-camera coordination can improve online multi-object $k$-coverage in a network of smart cameras, in a range of scenarios. In other cases, the problems of 1-coverage and $k$-coverage are suitably aligned that simpler approaches (e.g., using PTZ cameras), are sufficient. We use archetypal scenarios to illustrate these cases.

There are a number of interesting learning points to arise from this study. In this section, we discuss a particularly salient one relating to distributed autonomy and goal-awareness. One of the features of so-called *smart* systems, including smart camera networks, is that we delegate autonomy to the system, in expectation of achieving our high-level goals. A further feature of *distributed* smart systems is that this autonomy is typically also decomposed (into sub-decisions and sub-goals) and distributed about the entities that comprise the system as a whole.

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | | Scenario 5 | | Scenario 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Approach | $k \geq 3$ | $k = 2$ | $k \geq 3$ | $k = 2$ | $k \geq 3$ | $k = 2$ | $k \geq 3$ | $k = 2$ | $k \geq 3$ | $k = 2$ | $k \geq 3$ | $k = 2$ |
| Static-NN | 10.149 | 8.880 | 15.250 | 8.014 | - | - | - | 7.183 | - | 8.665 | 12.242 | 10.257 |
| PTZ-NN | 6.674 | 3.423 | 4.686 | 2.556 | 0.077 | 0.074 | 0.052 | 0.038 | 10.783 | 3.191 | 6.935 | 3.467 |
| Random-NN | 15.282 | 9.171 | 15.385 | 8.895 | - | 8.615 | 13.839 | 8.238 | 10.069 | 8.575 | 14.419 | 10.083 |
| RandomFoll-NN | 18.660 | 8.749 | 10.875 | 5.957 | 21.783 | 11.681 | 14.818 | 8.101 | 16.114 | 9.850 | 17.502 | 9.174 |
| Centralised | 3.189 | 1.552 | 2.447 | 1.392 | 3.616 | 1.918 | 3.039 | 1.428 | 5.395 | 2.863 | 3.864 | 2.298 |
| BC-AV | 2.679 | 1.689 | 1.761 | 1.117 | 2.337 | 1.475 | 1.316 | 0.881 | 3.634 | 2.579 | 3.781 | 2.462 |
| BC-NN | 2.790 | 1.803 | 1.695 | 1.110 | 1.819 | 1.092 | 1.385 | 0.935 | 4.002 | 3.025 | 3.266 | 2.200 |
| BC-GR | 3.613 | 2.607 | 4.551 | 3.159 | 6.484 | 5.142 | 3.690 | 2.953 | 7.736 | 4.496 | 8.562 | 5.737 |
| BC-RE | 2.235 | 1.353 | 1.353 | 0.869 | 1.856 | 1.241 | 1.409 | 0.920 | 3.807 | 2.580 | 3.045 | 2.000 |
| CL-AV | 4.667 | 3.159 | 4.097 | 2.252 | 5.151 | 3.122 | 4.216 | 2.243 | 7.748 | 4.094 | 5.058 | 2.990 |
| CL-NN | 4.209 | 2.297 | 3.137 | 1.768 | 4.297 | 2.267 | 2.875 | 1.559 | 6.191 | 4.177 | 4.954 | 2.759 |
| CL-GR | 6.248 | 3.985 | 5.162 | 3.178 | 9.654 | 6.521 | 4.807 | 3.499 | 13.063 | 6.477 | 11.757 | 7.138 |
| CL-RE | 4.113 | 2.088 | 3.360 | 1.787 | 4.172 | 2.435 | 3.345 | 1.835 | 6.697 | 3.721 | 4.628 | 2.710 |
| FU-AV | 4.698 | 2.964 | 4.503 | 2.826 | 4.572 | 2.919 | 3.061 | 1.936 | 5.606 | 3.625 | 4.880 | 2.978 |
| FU-NN | 4.407 | 2.756 | 2.940 | 1.914 | 3.607 | 2.138 | 2.212 | 1.132 | 5.540 | 2.935 | 4.283 | 2.730 |
| FU-GR | 11.805 | 6.799 | 10.930 | 5.900 | 19.598 | 10.930 | 13.191 | 7.830 | 15.386 | 9.284 | 16.065 | 8.700 |
| FU-RE | 3.711 | 2.055 | 3.608 | 2.099 | 3.374 | 2.057 | 2.806 | 1.696 | 5.657 | 3.705 | 5.018 | 2.661 |
| RA-AV | 4.129 | 2.510 | 2.658 | 1.553 | 3.553 | 2.732 | 1.943 | 1.201 | 4.303 | 3.121 | 4.150 | 2.756 |
| RA-NN | 3.428 | 2.115 | 2.104 | 1.192 | 2.422 | 1.664 | 1.999 | 1.187 | 4.294 | 2.878 | 3.427 | 1.978 |
| RA-GR | 4.533 | 3.065 | 5.254 | 3.293 | 7.521 | 6.726 | 4.404 | 3.612 | 10.043 | 5.271 | 9.753 | 6.537 |
| RA-RE | 3.021 | 1.780 | 2.528 | 1.533 | 2.474 | 1.601 | 1.937 | 1.107 | 4.675 | 2.575 | 3.440 | 2.175 |

**TABLE 2** Length of time until an object is provisioned by $k = 2$ and $k \geq 3$ cameras after it has been detected the first time. Average times have been taken over 30 experiments and across all individual objects.

This raises two important lines of questioning, as consequences. First, where should sub-decisions be made, and why? A natural justification for a particular distribution of autonomous decision making is to stipulate that decisions should be made where the decisions are most likely to be most effective at leading to the achievement of global goals. But it is not always possible to guarantee, or even determine this, since local decision making is subject to the constraints of varying local knowledge. This leads to the second line of questioning: what information is present and available at different loci of control within the system, and further, what information *should* be present, in order to enhance our ability to do well at the first question?

These issues show up in the system analysed here. Indeed, the observation above, that the network-wide goals are more effectively met when decisions are made by responding nodes, rather than request-issuing nodes, is an example. This can be analysed by considering the extent to which local camera nodes are aware of the impact of decisions relating to which goal to pursue.

Figure 7 provides a simple illustration of this issue. Here, camera $c_2$ is the recipient of a request to assist $c_1$ in 2-covering the object $p_1$. However, it is currently already occupied covering $p_2$, an object of which $c_1$ is unaware. In this instance, if $c_2$ were to treat the request from $c_1$ as an instruction, without considering its knowledge of its existing state and goal satisfaction, then, in terms of the system as a whole, the decision would have been made at a location where incomplete information led to a poor outcome. However, if $c_2$ instead makes the decision, based on the information available to it at its location, then it is able to take account of i) the request, and hence probability that there is an object that is currently not $k$-covered, near $c_1$, and ii) that it is already currently providing coverage of $p_2$. It would then be able to reason that in meeting $c_1$'s request, it would both sacrifice coverage of $p_2$, and also spend some period of time moving between the objects (here, rotating), during which time neither object would be $k$-covered. Thus, the result of $c_2$ treating the request as an obligation, rather than as additional information on which to base its decision, would be to reduce system-wide online multi-object $k$-coverage. While, in this article, we do not consider approaches that explicitly capture reasoning based on this idea of goal-awareness [40,38,36], we do compare behaviours that capture the effect of each of these two types of approach to distributed decision making.

This also highlights the tension between behaviour-based and cognitive model-based approaches to coordination, especially where resource and time constraints are important. A further line of questioning might therefore be: to what extent can such apparent awareness be a product of distributed behaviour-based coordination, and how can this be implemented in time-sensitive and resource-constrained systems? Given that cameras need to be able to make decisions based on rapidly changing states of themselves and their peers, and incorporate this information into their coordination behaviour, approaches to aggregating local knowledge, and building consensus about current goals, can only ever be partially effective at mitigating the issue illustrated in Figure 7, and discussed above.

| | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 | Scenario 5 | Scenario 6 |
|---|---|---|---|---|---|---|
| Static-NN | — | — | — | — | — | — |
| PTZ-NN | — | — | — | — | — | — |
| Random-NN | — | — | — | — | — | — |
| RandomFoll-NN | — | — | — | — | — | — |
| Centralised | 2.000 | 2.000 | 2.000 | 2.000 | 2.000 | 2.000 |
| BC-AV | 3.563 | 9.169 | 4.111 | 4.524 | 2.015 | 3.150 |
| BC-NN | 3.381 | 8.033 | 3.980 | 4.440 | 2.026 | 3.142 |
| BC-GR | 3.206 | 5.687 | 2.210 | 2.865 | 1.271 | 1.694 |
| BC-RE | 3.502 | 8.349 | 4.094 | 4.353 | 1.998 | 3.073 |
| CL-AV | 0.730 | 0.322 | 0.636 | 0.690 | 0.773 | 0.644 |
| CL-NN | 0.703 | 0.318 | 0.663 | 0.745 | 0.751 | 0.628 |
| CL-GR | 0.603 | 0.278 | 0.430 | 0.590 | 0.532 | 0.350 |
| CL-RE | 0.683 | 0.307 | 0.651 | 0.728 | 0.750 | 0.605 |
| FU-AV | 0.857 | 0.506 | 0.994 | 1.102 | 0.954 | 0.749 |
| FU-NN | 0.852 | 0.603 | 0.989 | 1.141 | 0.953 | 0.730 |
| FU-GR | 0.488 | 0.192 | 0.310 | 0.454 | 0.404 | 0.255 |
| FU-RE | 0.872 | 0.477 | 1.022 | 1.084 | 0.867 | 0.736 |
| RA-AV | 0.967 | 0.740 | 1.124 | 1.216 | 1.006 | 0.891 |
| RA-NN | 0.948 | 0.761 | 1.112 | 1.178 | 1.006 | 0.869 |
| RA-GR | 0.866 | 0.498 | 0.570 | 0.835 | 0.616 | 0.397 |
| RA-RE | 0.951 | 0.646 | 1.138 | 1.212 | 0.972 | 0.860 |

**TABLE 3** Mean number of exchanged messages over 1000 time steps and 30 experiments. Normalised by the number of identified object and the number of cameras in the scenario. Centralised is consistently 2, since at each time step, each camera sends its state to the central node, and receives instructions in response.

Regardless of which approach is taken, an implication of the results in this paper is that the design of effective coordination mechanisms, for contexts where goals change rapidly, should be based on decisions being made most locally to where they will be enacted: accounting for distributed goal awareness through local decision making leads to higher overall goal satisfaction. This becomes apparent when analysing the computational complexity of our approaches which results in the worst case in $\mathcal{O}(n)$ on each camera for communicating every detected object where $n$ is the number of available cameras. Each responding camera has to make decision based on potentially all other cameras which again results in a linear worst case computational complexity of $\mathcal{O}(n)$ where $n$ is the number of cameras. Thus the computational complexity of all the approaches studied here is low, and therefore the selection of the employed approach is not affected by the computational complexity.

In the particular instances of the online multi-object $k$-coverage problem studied here, (sub-)goal changes are highly rapid, invalidating the need of a camera to cover a given object. It will be interesting to explore this general result in other problems, where changes in goals have different characterisations. In principle, the speed and magnitude with which locally decomposed goals change, will motivate a greater need to make decisions locally to where those changes can be observed. Conversely, where the speed or magnitude of goal change is low, the penalty associated with making decisions based on out-of-date information may be lower.

## ACKNOWLEDGMENTS

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. Huang Chi-Fu, Tseng Yu-Chee. The coverage problem in a wireless sensor network. *Mobile Networks and Applications.* 2005;10(4):519–528.

2. Esterle Lukas, Lewis Peter R. Online Multi-object k-coverage with Mobile Smart Cameras. In: Proceedings of the International Conference on Distributed Smart Cameras:107–112; 2017.

3. Parker L. E., Emmons B. A.. Cooperative multi-robot observation of multiple moving targets. In: Proceedings of the International Conference on Robotics and Automation, vol. 3: :2082-2089 vol.3; 1997.

4. Khan A., Rinner B., Cavallaro A.. Cooperative Robots to Observe Moving Targets: Review. *IEEE Transactions on Cybernetics.* 2018;48(1):187–198.

5. Esterle Lukas, Lewis Peter R, Yao Xin, Rinner Bernhard. Socio-economic vision graph generation and handover in distributed smart camera networks. *ACM Transactions on Sensor Networks.* 2014;10(2):20.

6. Esterle L., Rinner B., Lewis P. R., Yao X.. Improved adaptivity and robustness in decentralised multi-camera networks. In: Proceedings of the International Conference on Distributed Smart Cameras:1-6; 2012.

7. Rinner B., Dieber B., Esterle L., Lewis P. R., Yao X.. Resource-aware configuration in smart camera networks. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshops:58-65; 2012.

8. Werger Barry Brian, Matarić Maja J.. From insect to Internet: Situated control for networked robot teams. *Annals of Mathematics and Artificial Intelligence.* 2001;31(1):173–197.

9. Jung Boyoon, Sukhatme Gaurav S. Cooperative multi-robot target tracking. In: 2006 (pp. 81–90).

10. Kolling Andreas, Carpin Stefano. Cooperative observation of multiple moving targets: an algorithm and its formalization. *The International Journal of Robotics Research.* 2007;26(9):935–953.

11. Eickstedt D. P., Benjamin M. R.. Cooperative Target Tracking in a Distributed Autonomous Sensor Network. In: OCEANS 2006:1-6; 2006.

12. Navarro-Serment L. E., Dolan J. M., Khosla P. K.. Optimal sensor placement for cooperative distributed vision. In: Proceedings of the International Conference on Robotics and Automation, vol. 1: :939-944; 2004.

13. Kumar Santosh, Lai Ten H, Balogh József. On k-coverage in a mostly sleeping sensor network. In: Proceedings of the International Conference on Mobile Computing and Networking:144–158; 2004.

14. Liu L., Ma H., Zhang X.. On Directional K-Coverage Analysis of Randomly Deployed Camera Sensor Networks. In: Proceedings of the International Conference on Communications:2707-2711; 2008.

15. Hefeeda M., Bagheri M.. Randomized k-Coverage Algorithms For Dense Sensor Networks. In: Proceedings of the International Conference on Computer Communications:2376-2380; 2007.

16. Elhoseny Mohamed, Tharwat Alaa, Yuan Xiaohui, Hassanien Aboul Ella. Optimizing K-coverage of mobile WSNs. *Expert Systems with Applications.* 2018;92:142 - 153.

17. Liu B., Dousse O., Nain P., Towsley D.. Dynamic Coverage of Mobile Sensor Networks. *Transactions on Parallel and Distributed Systems.* 2013;24(2):301-311.

18. Li J. S., Kao H. C.. Distributed K-coverage self-location estimation scheme based on Voronoi diagram. *IET Communications.* 2010;4(2):167-177.

19. Fusco G., Gupta H.. Selection and Orientation of Directional Sensors for Coverage Maximization. In: Proceedings of the International Conference on Sensor, Mesh and Ad Hoc Communications and Networks:1-9; 2009.

20. Micheloni C., Rinner B., Foresti G. L.. Video Analysis in Pan-Tilt-Zoom Camera Networks. *Signal Processing Magazine.* 2010;27(5):78-90.

21. SanMiguel J. C., Micheloni C., Shoop K., Foresti G. L., Cavallaro A.. Self-Reconfigurable Smart Camera Networks. *Computer.* 2014;47(5):67-73.

22. Piciarelli C., Esterle L., Khan A., Rinner B., Foresti G. L.. Dynamic Reconfiguration in Camera Networks: A Short Survey. *IEEE Transactions on Circuits and Systems for Video Technology.* 2016;26(5):965-977.

23. Ren W., Beard R. W., Atkins E. M.. Information consensus in multivehicle cooperative control. *IEEE Control Systems.* 2007;27(2):71-82.

24. Xiao L., Boyd S., Lall S.. A scheme for robust distributed sensor fusion based on average consensus. In: Proceedings of the International Symposium on Information Processing in Sensor Networks:63-70; 2005.

25. Meyer F., Braca P., Willett P., Hlawatsch F.. A Scalable Algorithm for Tracking an Unknown Number of Targets Using Multiple Sensors. *IEEE Transactions on Signal Processing.* 2017;65(13):3478-3493.

26. Fantacci C., Papi F.. Scalable Multisensor Multitarget Tracking Using the Marginalized $delta$-GLMB Density. *IEEE Signal Processing Letters.* 2016;23(6):863-867.

27. Werger Barry Brian. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence.* 1999;110(2):293 - 320.

28. Wooldridge Michael J.. *Introduction to Multiagent Systems.* John Wiley and Sons Inc.; 2001.

29. McCulloch Warren S. *Why the mind is in the head.* 1965.

30. Iberall Thea, Arbib Michael A.. *Schemas for the Control of Hand movements: An Essay on Cortical Localization.* 1989.

31. Balasuriya A., Schmidt H., Benjamin M.. Nested Distributed Autonomy Architecture for Undersea Sensor Networks. In: Proceedings of OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean:1-5; 2008.

32. Burns Alyxander, Schulze Bernd, St. John Audrey. Persistent Multi-robot Formations with Redundancy:133–146. Springer International Publishing 2018.

33. Saldaña D., Prorok A., Campos M.F.M., Kumar V.. Triangular Networks for Resilient Formations. In: Proceedings of the International Symposium Distributed Autonomous Robotic SystemsSpringer; 2016.

34. Esterle L.. CHAINMAIL: Distributed Coordination for Multi-task k-Assignment Using Autonomous Mobile IoT Devices. In: Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS):85-92; 2018.

35. Esterle L., Brown J. N.. Levels of Networked Self-Awareness. In: Proceedings of the International Workshops on Foundations and Applications of Self* Systems (FAS*W):237-238; 2018.

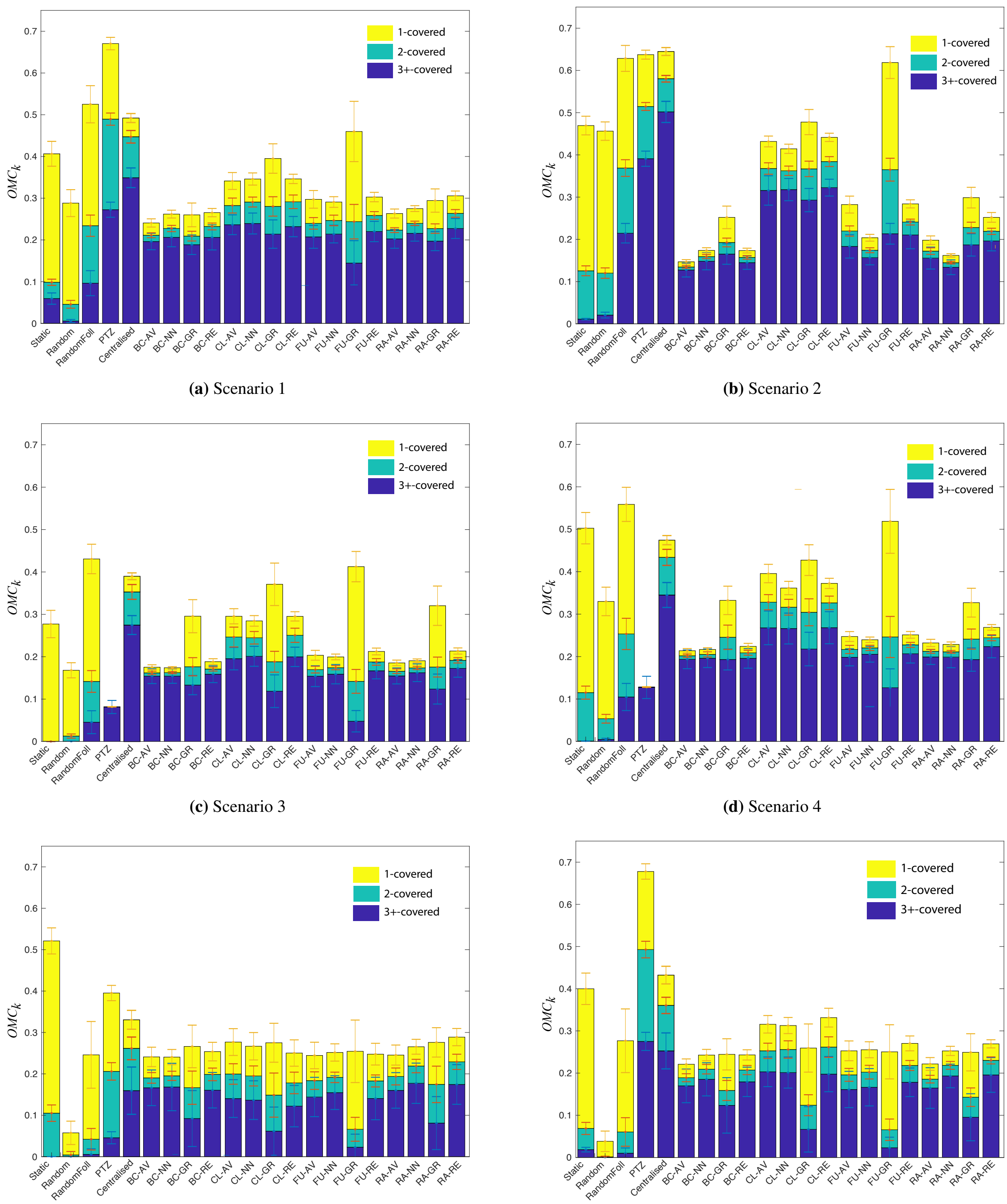
36. Esterle L., Rinner B.. An Architecture for Self -Aware IOT Applications. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP):6588-6592; 2018.

37. King David W., Esterle Lukas, Peterson Gilbert L.. Entropy-Based Team Self-Organization with Signal Suppression. *Proceedings of the Conference on Artificial Life.* 2019;(31):145-152.

38. Esterle L.. Goal-Aware Team Affiliation in Collectives of Autonomous Robots. In: Proceedings of the International Conference on Self-Adaptive and Self-Organizing Systems:90-99; 2018.

39. Esterle L., Lewis P. R., Caine H., Yao X., Rinner B.. CamSim: A Distributed Smart Camera Network Simulator. In: Proceedings of the Seventh International Conference on Self-Adaptive and Self-Organizing Systems Workshops:19-20; 2013.

40. Lewis Peter R., Chandra Arjun, Faniyi Funmilade, et al. Architectural Aspects of Self-Aware and Self-Expressive Systems: From Psychology to Engineering. *Computer.* 2015;48(8):62–70.




**(a)** Scenario 1

**(b)** Scenario 2

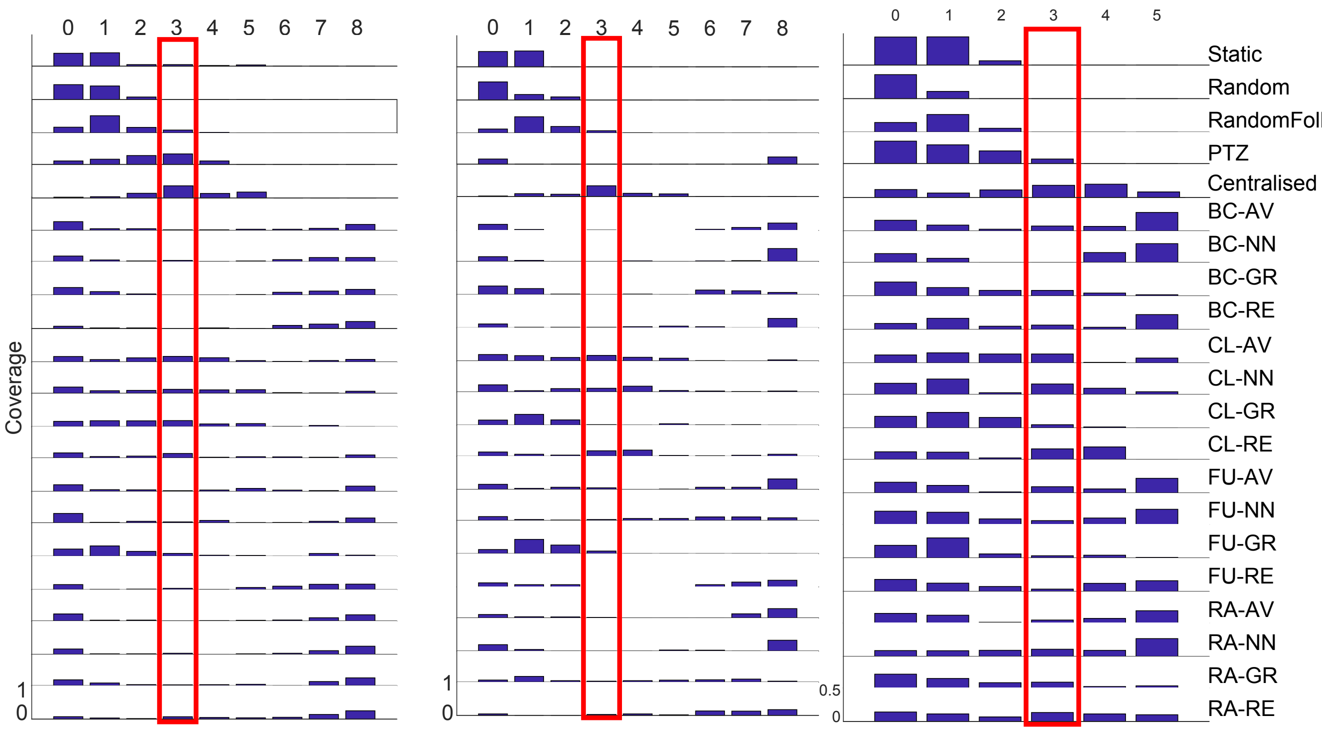
**(c)** Scenario 3

**(d)** Scenario 4

**(e)** Scenario 5

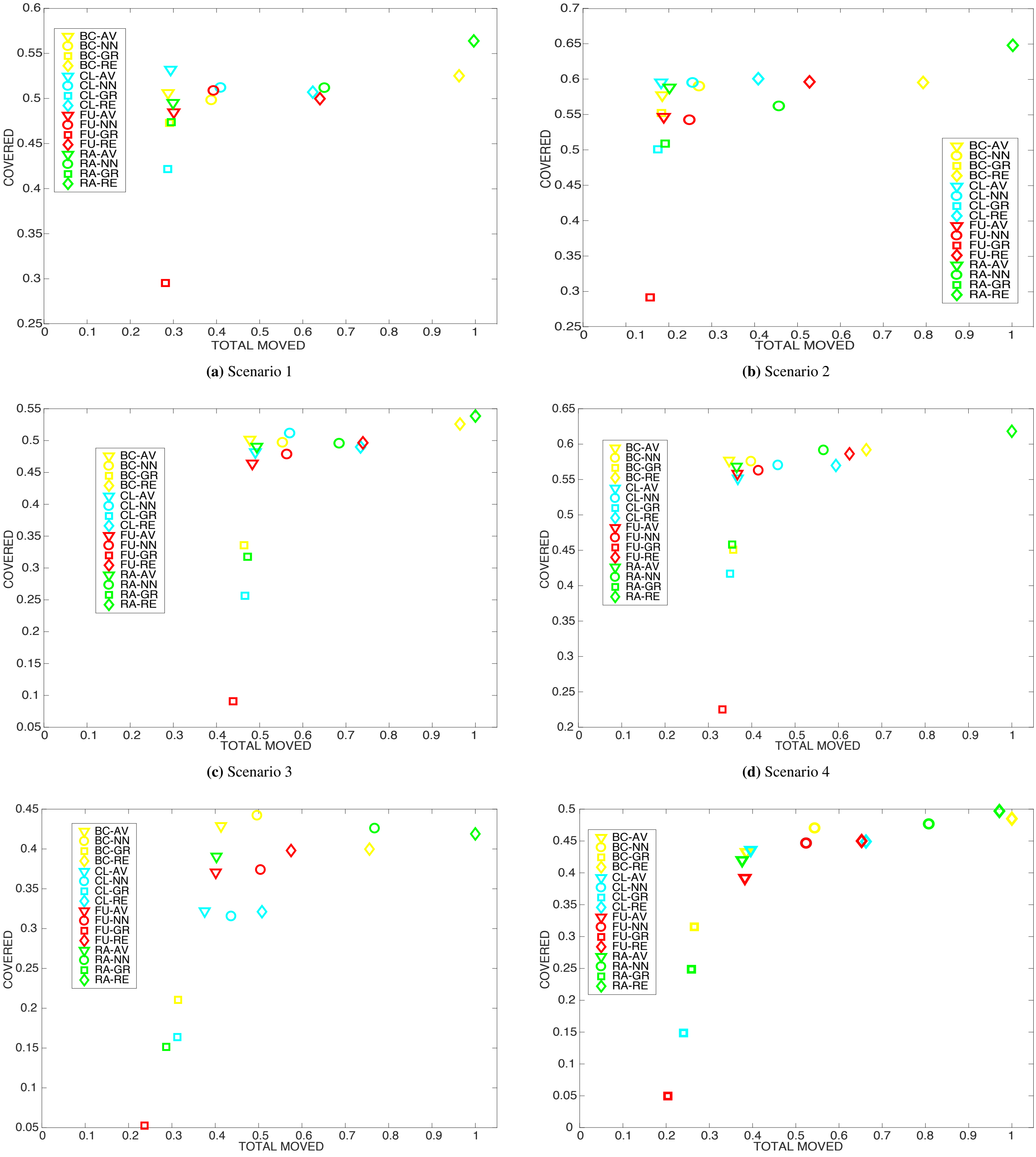
**(f)** Scenario 6

**FIGURE 4** Comparison of the different approaches in terms of $k$-coverage with default behaviour *random movement with following*. The top yellow bar represents mean 1-coverage, second green bar illustrates mean 2-coverage, and bottom blue coverage represents mean $k$+-coverage and corresponding standard deviations over 30 runs. Static represents *Fixed lcoations*, Random is for *Random movement*, RandFoll is for *Random and following* baseline behaviour. Communication strategies are BC for Broadcast, CL for $k$-CLOSEST, FU for $k$-FURTHEST, RA for $k$-RANDOM. Response models are AV for *Available*, NN for *Newest-Nearest*, GR for *Graph*, and RE for *Received calls*.




**(a)** Scenario 1 **(b)** Scenario 3 **(c)** Scenario 5

**FIGURE 5** Illustration of over- and under-provisioning of the different studied approaches at a single point in time (time step 500). The red border highlights the results where $k = 3$ cameras cover objects. Baseline approaches often under-provision and rarely over-provision, while coordinated approaches often over-provision but as a result, rarely miss objects altogether. The *centralised* approach provides very good $k$ coverage with little over and under-provisioning in all three scenarios. Static represents *Fixed lcoations*, Random is for *Random movement*, RandFoll is for *Random and following* baseline behaviour. Communication strategies are BC for Broadcast, BE for $k$-BEST, CL for $k$-CLOSEST, FU for $k$-FURTHEST, and RA for $k$-RANDOM. Response models are AV for *Available*, NN for *Newest-Nearest*, GR for *Graph*, and RE for *Received calls*.




**(a)** Scenario 1 **(b)** Scenario 2 **(c)** Scenario 3 **(d)** Scenario 4 **(e)** Scenario 5 **(f)** Scenario 6

**FIGURE 6** Trade-off of movement vs. achieved proportional coverage of previously seen objects for Scenario 1, 2 and 6

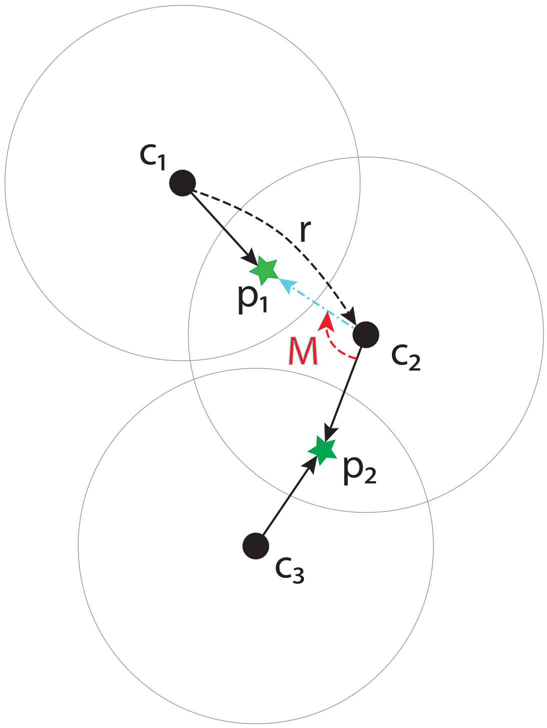



**FIGURE 7** Illustration of how a distributed awareness of the system's goals and current state can impact coverage and movement. Black spots indicate cameras, green stars illustrate objects of interest, grey circles indicate potential sensing range, black solid lines indicate the cameras and the currently provisioned objects. The black dashed line *r* represents a request from one camera to another. The dashed red line indicates potential movement in order for the requested camera to provision the other object, indicated with a dashed blue arrow.

## AUTHOR BIOGRAPHY

**Lukas Esterle** is an Assistant Professor in Electrical and Computer Engineering at Aarhus University, Denmark. Lukas holds a Masters degree in Computer Science and a Dr.-techn. in Electrical Engineering from Alpen-Adria University Klagenfurt. He was a postdoctoral researcher at the University of Technology in Vienna and a Marie Skłodowska Curie Fellow at Aston University, Birmingham, UK funded by the European Commission. His research interests are autonomous and self-aware systems, Computational and Artificial Intelligence, multi-agent and cyber-physical systems, and nature-inspired approaches to solve complex problems. Lukas co-authored several books on Self-aware computing systems and made substantial contributions to several self-aware computing applications around visual sensor networks and multi-robot systems. Lukas is a member of the Aarhus University Centre for Digitalisation, Big Data and Data Analytics (DIGIT).

**Peter R. Lewis** is a Senior Lecturer in Computer Science at Aston University, in Birmingham, UK. With a background in computational intelligence, Peter has made significant contributions to the field of self-aware computing, including the foundational book Self-aware Computing Systems: An Engineering Approach, in 2016. More broadly, his research is often inspired by biological, social and psychological processes, and advances our understanding about how to build intelligent systems that are embedded in society, learning and adapting on an ongoing basis. Through ongoing industrial research collaborations, his work has been applied in areas such as smart camera networks, interactive music devices, avionics, manufacturing, and cloud computing. He obtained his PhD at the University of Birmingham, and is a member of the Aston Lab for Intelligent Collectives Engineering (ALICE).

**How to cite this article:** Esterle L., Lewis P.R. (2018), Distributed Autonomy and Trade-offs in Online Multi-object $k$-coverage, *Computational Intelligence*, *XXXX*.