

# Self-organising Zooms for Decentralised Redundancy Management in Visual Sensor Networks

Lukas Esterle and Bernhard Rinner  
Institute of Networked and Embedded Systems  
Alpen-Adria-Universität Klagenfurt  
LakesideLabs Klagenfurt  
Klagenfurt, Austria  
{firstname.lastname}@aau.at

Peter R. Lewis  
Aston Lab for Intelligent Collectives Engineering (ALICE)  
Systems Analytics Research Institute  
Aston University  
Birmingham, UK  
p.lewis@aston.ac.uk

**Abstract**—When visual sensor networks are composed of cameras which can adjust the zoom factor of their own lens, one must determine the optimal zoom levels for the cameras, for a given task. This gives rise to an important trade-off between the overlap of the different cameras’ fields of view, providing redundancy, and image quality. In an object tracking task, having multiple cameras observe the same area allows for quicker recovery, when a camera fails. In contrast having narrow zooms allow for a higher pixel count on regions of interest, leading to increased tracking confidence. In this paper we propose an approach for the self-organisation of redundancy in a distributed visual sensor network, based on decentralised multi-objective online learning using only local information to approximate the global state. We explore the impact of different zoom levels on these trade-offs, when tasking omnidirectional cameras, having perfect 360-degree view, with keeping track of a varying number of moving objects. We further show how employing decentralised reinforcement learning enables zoom configurations to be achieved dynamically at runtime according to an operator’s preference for maximising either the proportion of objects tracked, confidence associated with tracking, or redundancy in expectation of camera failure. We show that explicitly taking account of the level of overlap, even based only on local knowledge, improves resilience when cameras fail. Our results illustrate the trade-off between maintaining high confidence and object coverage, and maintaining redundancy, in anticipation of future failure. Our approach provides a fully tunable decentralised method for the self-organisation of redundancy in a changing environment, according to an operator’s preferences.

**Keywords**—visual sensor networks; redundancy management; decentralised learning; self-organisation; runtime trade-offs

## I. INTRODUCTION

Smart visual sensor networks are sensor networks comprised of *embedded smart cameras*, which each combine a visual sensor, a processing unit and a communication interface. Over the last fifteen years, embedded smart cameras have evolved from traditional cameras [25]. While processing capabilities are still limited, they allow to pre-process video data on board, transmitting only aggregated information, instead of plain images. Cameras run computer vision algorithms which are typically given surveillance tasks, such as identifying and tracking objects of interest within a scene. Many single smart cameras are now often connected in distributed smart camera networks [20].

In networks of smart cameras, the operator often wants to

focus the attention of the cameras on important areas, rather than observing the same space, as this may be seen as a waste of resources. Therefore, these cameras are typically deployed such that the overlap of their fields of view (FOVs) and hence the redundancy of the observed area is minimised [10]. However, overlapping FOVs can also be useful. In a network where cameras may fail, due to power or network issues, or due to malicious activity, these overlaps between FOVs provide redundancy when tracking objects. If a camera providing a view on a region of interest fails, then another camera also able to view that area may take over its tasks immediately, without needing to adjust its own zoom first. This exposes a trade-off between redundancy in anticipation of failure and image quality.

Calibrating FOVs manually is both time and resource intensive. Also, the topology of the network can change at runtime, requiring further adjustments. Therefore we propose an approach by which cameras autonomously manage the overlaps between their FOVs during runtime. In this context, we are interested in a network’s ability to manage the trade-off between redundancy and image quality, adapting zoom lengths as appropriate in a potentially changing environment.

While visual sensor networks can be used for a variety of tasks, object tracking is one of the most important performed tasks by modern smart cameras. In object tracking, a description of the object of interest is initially provided to the camera, which thereafter attempts to re-identify this object in consecutive frames of its own FOV. There are various tracking algorithms which can be used to locate objects in each frame. They provide so-called *tracking confidence* values, representing the probability of having successfully matched the given object description with a frame. Hence, a camera having a high confidence value for a given object, indicates a high probability of having successfully identified its presence.

In many cases, it is desirable that the management of a smart camera network is achieved in a decentralised fashion. This avoids bottlenecks associated with central nodes, facilitates scalability and provides resilience since there is no single point of failure for the network. To overcome the problem of how to achieve coordination of tracking responsibilities in a decentralised setting, Esterle et al. [7] presented a market-based approach based on self-interested cameras which exchange object tracking responsibilities using Vickrey auctions. In this

approach, an object is only tracked by a single camera within the network at any time. Furthermore, the tracking camera decides when to *hand over* the tracking responsibility of an object to another camera. In such a case, it initiates an auction to solicit the tracking responsibility. Other cameras receiving this auction invitation, try to detect the object within their FOV. A bid is generated based on the confidence of the tracker and the visibility of the object representing a quantifiable and comparable measurement. This use of auctions amounts to a query-based approach to the decentralised allocation of objects to cameras, with the inherent property of being able to adapt to a changing environment or network topology. However, this approach only considered cameras with fixed lenses, and hence fixed FOVs. Variable zoom lenses present an opportunity to also adapt the FOVs themselves, leading to a trade-off between coverage and image quality. This is an important trade-off to manage, since it balances the tension between high performance at the tracking task, driven by maximising image quality, and redundancy based on maximising overlapping FOVs.

The key contribution of this paper is a novel, decentralised and self-organising approach for smart cameras to determine their zoom levels in order to manage the redundancy-performance trade-off, in coordination with other cameras in the network. We firstly perform an extensive offline profiling of how varying zoom levels drives the trade-off between the confidences of the tracked objects, the proportion of tracked objects from those present in the environment, and the overlap between the different cameras. Secondly, we describe the approach in detail, which is based on decentralised reinforcement learning, able to be tuned according to an operator's preferences. This allows the network to achieve appropriate zoom levels dynamically at runtime, obtaining desired outcomes in the trade-off space. We show how to self-organise with redundancy, with a control parameter to determine the amount of redundancy expressed by the self-organisation. Thirdly, we allow cameras to be removed at runtime. We capture the trade-off between high confidence throughout the run, with built-in self-healing capabilities of Esterle et al's original approach (e.g. as explored in [6]), against a new anticipation of such uncertainties, brought about through deliberately overlapping FOVs. While this self-healing capabilities allow the network to recover, this anticipation capability increases the resilience of the network which in turn leads to lower impact when camera failure occurs. Nevertheless, the proposed approach is not limited to smart camera networks. In many networks one can adjust the sensing range, sensor resolution, or—for directional sensors—the orientation.

The rest of the paper is structured as follows. Section II discusses related work on resilience and coverage management in sensor networks. Section III introduces the proposed scenario and the problem as well as the different objectives and metrics for evaluation formally. Section IV discusses the offline analysis of the effect of zoom on the confidence of the tracking algorithm. Section V extensively profiles the trade-offs between the zoom level, the proportion of tracked objects and the achieved tracking performance. Furthermore, this section gives an overview of the experiments performed. We introduce online reinforcement learning for self-organised zoom management in Section VI. This enables us to trade off redundancy (i.e. overlap) and the maximisation of the

proportion of objects tracked, for high tracking confidence, in a decentralised manner. Section VII briefly discusses the anticipatory resilience properties of our approach in the case of network uncertainties. Section VIII concludes the paper and discusses open challenges and future work.

## II. RELATED WORK

While we are not primarily interested in camera placement and coverage optimisation, this is a very important and related task in multi-camera surveillance and has been approached using self-organising mechanisms. Dieber et al. [3] define static observation points and employ an evolutionary algorithm in order to determine the optimal coverage of these points. These observation points are selected by an operator indicating important areas to be observed. Schwager et al. [21] present a distributed approach in order to coordinate unmanned aerial vehicles for optimal coverage of a predefined area. They are able to cover the area optimally even in the presence of uncertainties. Morsley et al. [18] facilitate various evolutionary-like algorithms based on particle swarm optimisation to approach the camera placement problem. In multiple iterations they are able to find a satisfactory camera placement with optimal coverage of the defined area. Hoffman et al. [10] present a self-organising approach to camera pan-tilt-zoom configurations, however they are concerned with minimising overlap between FOVs, rather than seeing this as an opportunity to make use of redundancy. Further information on camera placement algorithms and coverage optimisation is given in [2], [15], [19]. However, our aim is not maximal coverage. Instead, we are interested in the autonomous management of the trade-off between the number tracked objects, their tracking confidence, and the redundancy associated with the covered area.

Self-healing and resilience are concepts dating back at least to the middle of the 20th century in electronic engineering and telecommunications. By the early 21st century, they became also important in software systems, in particular due the Autonomic Computing vision (e.g. [22]). Redundancy has long been a common approach to achieving resilience, and with the proliferation of complex, decentralised and dynamic computing systems, researchers have increasingly looked to self-organisation to provide such redundancy [26].

In “flat” systems, resilience in the presence of failing nodes can be virtually transparent, if nodes can act as perfect substitutes, providing equivalent functionality to the wider system. Many examples of this (e.g. [13]) lie in service-based computing, where many equivalent nodes compete, for example in a market mechanism, to provide an equivalent service. In more complex systems, such as those based on roles, many of the features of successful resilience, and which facilitate effective resilience and self-healing are architectural. For example, how should nodes in a network be connected, and how should they share work, such that in the event of node failure, others can take their place? The self-organisation of architecture is indeed the topic of ongoing research, and in many places has looked to biological systems for inspiration. Capodiecici and Hart [1], for example, demonstrate a wide range of architectures emerging from self-organisation behaviour of nodes, inspired by swarm chemistry.

Resilience to node failure is a key design factor for peer-to-peer overlay networks, where it is desirable that connectivity

be maintained even after node failure. For example, Massoulié et al. [14] propose a distributed algorithm for the formation of such overlay networks designed explicitly with resilience in mind. Similarly, Misra and Mandal [16] and Vlajic and Moniz [24] are concerned with resilience and self-healing in terms of connectivity in wireless sensor networks. Galzarano et al. [9] consider another aspect of resilience in sensor networks, namely that of tolerating cases when sensors provide faulty (i.e. corrupt) data. Du et al. [4] also consider resilience in sensor networks where nodes may fail, however they highlight that sensor coverage as well as connectivity are important in the sensor network context. They present a fuzzy logic-based technique with low resource requirements, for the distributed control of mobile sensors' locations, such that coverage is maintained, in a scenario where other sensors can fail.

### III. SCENARIO & PROBLEM DEFINITION

We are concerned with smart visual sensor networks where individual nodes can observe their local environment through an omnidirectional camera with 360-degree view, each fitted with an adjustable zoom lens. This results in a circular FOV where the radius corresponds to the current zoom level. Adjusting the zoom level of a camera lens changes its FOV. This in turn impacts on its ability to correctly track objects, since a wider zoom leads to fewer pixels being used for the region of interest. However, narrower zooms mean that objects may often be outside of the field of view, and hence are not trackable at all by that camera. In determining optimal zoom levels for a set of cameras, an important question is to what extent cameras' FOV should overlap, providing redundancy for objects covered by more than one camera. Such redundancy is beneficial if camera failure occurs, since objects are not lost to the network. However, having cameras maintain wide zooms reduces the pixel count on regions of interest, leading to decreased performance in computer vision tasks.

The considered network of uncalibrated cameras are not time synchronised. Furthermore, cameras in the network do not have any initial knowledge about their environment nor about the total number of objects or cameras available in the scenario. Cameras are able to learn about their local, observable space and interact with other cameras in the network through message passing. Nevertheless, their own location as well as the location of other cameras in the environment is unknown to all cameras. The network of cameras is primarily interested in tracking objects moving through this space. As tracking is a resource intensive task, each object is only tracked by a single camera at the time to limit the network-wide resource consumption. We employ the tracking coordination algorithm as described by Esterle et. al [7] to assign tracking responsibilities among the cameras in the network at runtime.

It is important to note that in this paper we are not trying to overcome the classical coverage maximisation problem, where cameras try to maximize their covered area while satisfying a certain quality criteria. Instead, this paper shows how cameras can, in a self-organised manner, focus their attention towards important areas during runtime and cover those sufficiently. This means at the network-wide level, we are interested in achieving solutions specified by an operator's preferences over a multi-objective space. The three global objectives are i) maximising the tracking **confidence** of the currently tracked

objects, ii) maximising the **proportion** of objects able to be tracked, as determined by the FOVs of the cameras in the network, and iii) maximising the **overlap** of the FOV to provide increased redundancy of covered areas. However, these three objectives are in tension with each other, primarily since (ii) and (iii) can be achieved by widening the FOV in different circumstances, and (i) can be achieved by narrowing the FOV around objects of interest to maximise the pixel count. These objectives are based on the following metrics respectively:

- **Confidence** of tracked objects: confidence values returned from the cameras' trackers, representing the probability of the tracker having the *correct* object, summed over all tracked objects.
- **Proportion** of available objects tracked: ratio of objects with a positive confidence value (i.e. tracked) compared to objects known (i.e. "owned" in the market-based handover approach used).
- **Overlap**: represents the geometric overlap of the circular FOVs of the individual cameras.

For network-wide performance evaluation we collect these global metrics centrally. However, due to the decentralised nature of the network, such global information is not known to individual cameras for the purposes of decision making. For the global confidence metric,  $confidence_G$ , we accumulate the local confidences achieved by the individual cameras. A camera's local confidence is calculated as the mean confidence over all objects it tracks, and is defined by the tracking algorithm chosen. In general, confidence is affected by two factors: the distance between the object and the camera and the zoom factor of the camera. The distance of the object to the camera affects the acquired pixels on the target. If the object is close, the camera can capture more details of the object and *vice versa*. In the simulation study carried out in Sections V-VII, we simply use the Euclidean distance between the camera and the object to determine the number of pixels acquired by the camera. The zoom factor on the other hand, affects the total number of acquired pixels. As the total number of pixels for an entire image acquired by the camera does not change due to changing the optical zoom level, using a narrow zoom level, the pixel count of the selected target region is relatively high in comparison to the covered area. When widening the zoom level, the number of pixels for the whole image stays the same, but decreases for the target region.

$$confidence_G = \sum_{i=1}^{|C|} confidence_L(c_i) \quad (1)$$

$$confidence_L(c_i) = \frac{\sum_{j \in o(c_i)} \wp(j) * \kappa(c_i)}{|o(c_i)|}$$

where  $o(c_i)$  is the set of owned objects of camera  $c_i$  and  $\wp(j)$  defines the confidence of the tracker for the specified target  $j$  while  $\kappa(c_i)$  defines the deviation of the tracking confidence due to the current zoom level of the camera. In simulation  $\wp(j)$  is defined by the inverse Euclidean distance between the object and the camera, normalized by the maximum possible zoom of a camera. In real tracking algorithms,  $\wp(j)$  would be the correlation between the model of the object and the identified image patch. For  $\kappa(c_i)$  we employ a simple linear approximation to reflect effects of zoom level on the performance of the tracker. We elaborate on this effect in Section IV. For



Fig. 1. One of the six examples of the same region of interest at four different zoom levels resulting in four images with different pixel densities. From left to right, the region of interest has 1600x800 pixels, 800x400 pixels, 160x120 pixels, and 80x60 pixels. As the pixel count lowers, the probability of positively detecting the searched model decreases. The red-black-red square in the leftmost image marks the template used for the detection evaluation.

the second metric,  $proportion_G$ , we also rely on the global knowledge of available objects within the network. Here the sum of tracked objects by the individual cameras is divided by the total number of objects in the scenario.

$$proportion_G = \sum_{i=1}^{|C|} \frac{o(c_i)}{O} \quad (2)$$

where  $o(c_i)$  is the number of tracked objects by camera  $c_i$ .  $O$  is the total number of objects within the current scenario, including tracked as well as untracked objects. The global overlap,  $overlap_G$ , is calculated from the true overlap resulting from the sum of overlapping circular segments of different FOVs. The overlap between two cameras is only added once to the accumulated overlap, i.e. not once *per camera*.

$$overlap_G = \left( \frac{r(c_1)^2}{2} * (a(c_1) - \sin(a(c_1))) \right) + \left( \frac{r(c_2)^2}{2} * (a(c_2) - \sin(a(c_2))) \right) \quad (3)$$

where  $r(c_i)$  refer to the radii of the two cameras' FOVs and  $a(c_i)$  to the angle between the intersecting points of camera  $i$ , respectively.

In our application scenarios we make three important assumptions: (i) the cameras have no knowledge about their own position and do not know which nodes are in their vicinity, (ii) cameras are able to communicate with each other via message passing, and (iii) cameras are able to (re-)identify objects without errors.

#### IV. THE IMPACT OF ZOOM LEVEL ON TRACKING

In visual tracking, a model of an object to be tracked is used to search the current FOV. This is a widely researched area and many approaches have been proposed in the literature. In this paper we are agnostic with respect to the particular tracking algorithm to be used. While we use tracking algorithms to identify objects within the FOVs of our cameras, we do not investigate the different approaches here. Nevertheless, the performance of a tracking algorithm is typically highly influenced by the level of detail captured. This can be expressed as *pixel count* or *pixel density*, where a pixel defines the size of the smallest, clearly observable object with distinct boundaries. Reducing the FOV of a camera, and hence increasing the pixels on target, allows us to achieve a better performance in the observation process in terms of confidence [8].

For a camera with a given pixel density, the employed level of optical zoom determines the number of pixels used to

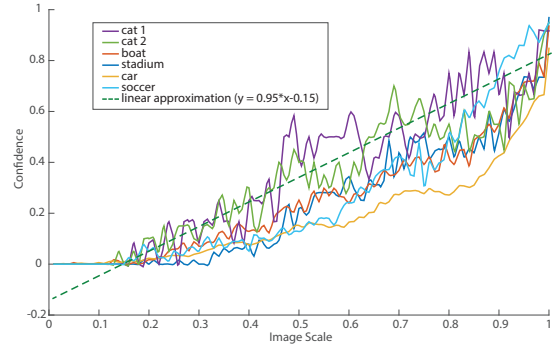


Fig. 2. Detection confidence of the SURF-detection algorithm at different scales of the image. The dashed green line indicates the approximation of the mean detection confidence of the SURF-detection algorithm

represent the region of interest. This is illustrated in Figure 1 for an example image. While the exact impact of pixel density on object tracking will vary depending on camera type and environmental factors such as lighting, a set of profiling experiments which factor these issues out identified a polynomial relationship between pixel density and classification success rate. In the profiling experiments, a small region was extracted from each image as template for a detection algorithm (illustrated as red-black-red square in Figure 1). We used simple SURF keypoints to detect features in the image. The confidence of the detection algorithm is based on equation 4.

$$\wp(j) = \frac{feat_{det}(j)}{feat_{total}(j)}. \quad (4)$$

Here  $feat_{det}(j)$  refers to the features from the internal representation of object  $j$  positively matched with the detected features in the current image, while  $feat_{total}(j)$  refers to the total number of features extracted from the original model representing object  $j$ .

To profile the deviation in the tracking performance based on the number of pixels, we used six different example images and scaled them in 1-percent steps. For each scaled image we performed a detection using the template from the full scale image. The deviation in tracking confidence based on equation 4 is shown for all six samples in Figure 2. 'Cat 2' is shown as an example in Figure 1. Based on the mean value, an approximation of the tracking confidence is shown as a dashed green line. While 360-degree cameras introduce visual distortions affecting the detection confidence, for the purpose

of the simulation study in this paper, we assumed a simplified relationship informed by these profiling experiments, as described in equation 5.

$$\kappa(c_i) = 0.95 * \left(1 - \frac{r(c_i)}{\argmax(r(c_i))}\right) - 0.15 \quad (5)$$

where  $r(c_i)$  is the radius of camera  $c_i$  based on its zoom level and  $\argmax(r(c_i))$  defines the maximum range based on the maximum zoom possible for the same camera. In order to ensure non-negative confidence values on the camera level due to a negative  $\kappa(c_i)$ , we only allow zoom levels on each camera resulting in a  $\kappa(c_i)$ -factor of at least 0.

## V. OFFLINE ANALYSIS OF TRADE-OFFS

For our evaluation, we performed various experiments in using our simulation tool CamSim [5]<sup>1</sup>. We designed a total of 12 different scenarios in order to evaluate the impact between achieved confidence of the individual cameras with varying proportion of tracked objects at different zoom levels. Due to stochasticity, all our experiments have been repeated 30 times and the mean result is shown. We measured the global metrics as discussed in Section III to establish a profile of the performance against each of the corresponding objectives. An overview of the different scenarios of the visual sensor networks used for our evaluation is given in Figure 3. We assume a varying number of omnidirectional cameras within each scenario. The location of the camera is static and is illustrated as a green dot. The corresponding FOVs are indicated as black and yellow circles. In scenarios 1-8, objects each followed random vectors, until they hit the boundary of the scenario, after which they bounced back again in a random direction. For scenarios 9-12 we defined the movement of the objects *a priori*, i.e. each object followed a predefined path through the environment. For all scenarios, we kept the number of objects to be tracked constant throughout the entire simulation run.

In an initial exploration of the solution space, we discretised the range of possible zoom lengths into six, and statically assigned each camera with one of the six different levels of zoom, comparing the possible configuration combinations exhaustively. The discrete zoom levels are equally distributed between 0 and the maximum zoom level of the given camera. In our simulations we assume the same cameras to be used for the entire network resulting in the same possible zoom levels. The mean results of all three global objectives have been normalized by the maximum mean value of all configurations in this scenario. Furthermore, the result of each scenario (depicted as a single dot in the plot) has been colour-coded in red, green, and blue based on its normalised outcome. The normalised confidence value represents the ratio of red for each result, the ratio of green represents the normalised proportion of tracked objects accumulated over time and all cameras in the network, and the amount of blue represents the overlap again accumulated over time and the cameras of the network normalised by the maximum value. Figure 5 shows examples of the exploration of the solution space. The trade-off between the different possible preferences an operator could choose from becomes apparent. More details are shown in

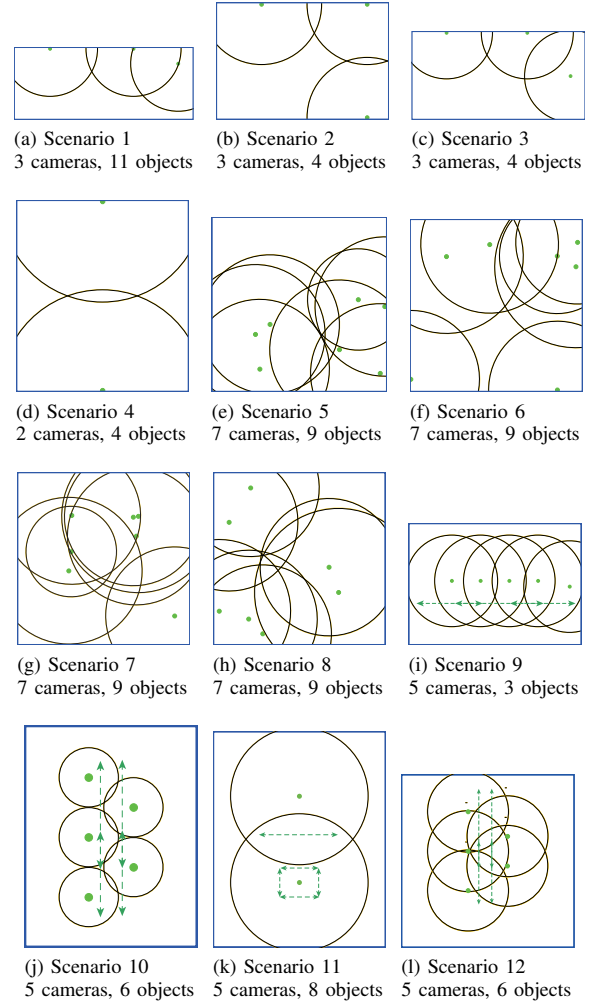


Fig. 3. The different scenarios used for our tests with our simulation tool CamSim. A green dot represents a camera, the associated yellow and black circles represents its FOV. Green arrows indicate the predefined movement paths. Blue boundaries indicate the limits of the simulation area.

Figure 4 which illustrates the results for scenario 3 when performing pairwise comparisons of the three metrics. The negative correlation between overlap and confidence as well as the positive correlation between proportion and overlap becomes apparent. This negative correlation is due to the fact that smaller zoom levels may only exhibit little to no overlap but due to the higher pixel density achieve higher tracking confidence. The positive correlation between overlap and proportion is based on the fact that the probability of tracking all objects is higher if the cameras cover a larger area. Of particular interest is the correlation between proportion and confidence, where the highest confidence is only reached when a certain percentage of achievable proportion is considered. From this point of view, we can infer that it is beneficial to the camera not to try tracking all objects possible and focus its zoom and attention on a certain area.

## VI. ONLINE REDUNDANCY MANAGEMENT

Considering the trade-off between the global metrics exposed in Section V, one could perform a multi-objective

<sup>1</sup>CamSim is open source and can be downloaded from <http://www.epics-project.eu/CamSim/>

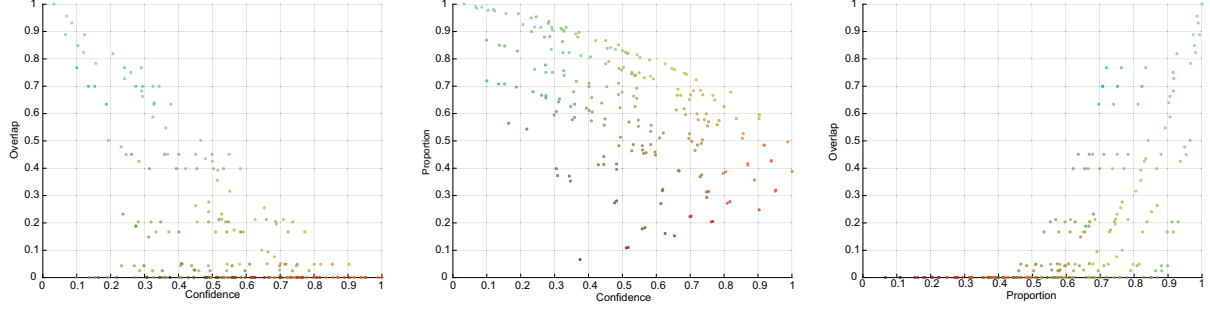


Fig. 4. From left to right the figures show overlap against confidence, proportion against confidence, and proportion against overlap for scenario 3. The results have been obtained in 30 independent runs and the mean values, normalised by the maximum outcomes, are shown. Each result is colour-coded: confidence is represented by ratio of red, proportion is represented by ratio of green, and overlap is represented by ratio of blue.

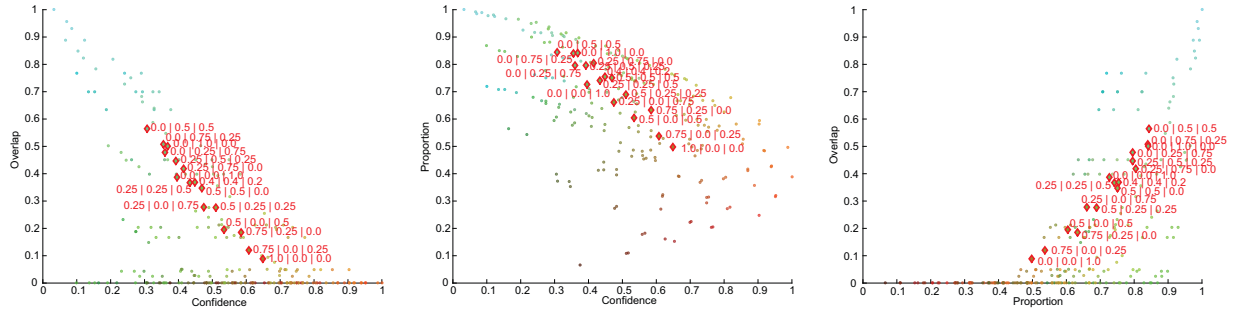


Fig. 6. From left to right the figures show overlap against confidence, proportion against confidence, and proportion against overlap for scenario 3. The results have been obtained in 30 independent runs and the mean values, normalised by the maximum outcomes, are shown. Each result is colour-coded: confidence is represented by ratio of red, proportion is represented by ratio of green, and overlap is represented by ratio of blue. Additionally, the results of the bandit solver have been plotted as red diamonds. The numbers next to the diamond represent the  $\alpha$ ,  $\beta$ , and  $\gamma$  value of the reward function used for the bandit.

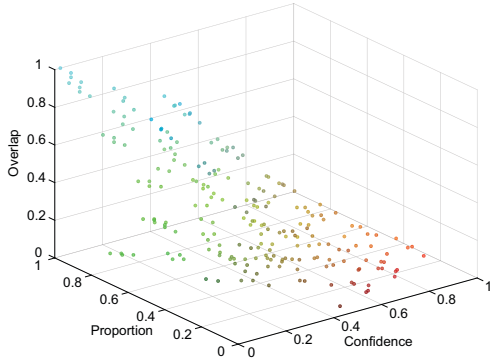


Fig. 5. The performance results of scenario 1 comparing all three metrics against each other for all possible combination of 6 zoom levels.  $3^6$  data points have been obtained over 30 runs and the mean values, normalised by the maximum outcomes, are shown. Each result is colour-coded: confidence is represented by ratio of red, proportion is represented by ratio of green, and overlap is represented by ratio of blue.

optimisation of network-wide zoom factors, to be solved in a centralised fashion. Such an approach could even be tackled online, with optimal zoom factors sent to each camera in a command-and-control manner as the environment changed. However, this would violate the assumption of a lack of central

control, which motivates the underlying camera handover mechanism [7]. The aim of this paper is to augment decentralised handover with self-organising zoom behaviour. For this reason, we transfer the adaptation capabilities to the individual cameras themselves, while still analysing the global behaviour of the entire network as described in Section III. While we use global information for evaluation at the network level, the cameras themselves do not have such global information for the purposes of decision making. Therefore, only local information acquired by a camera is used when deciding how to adapt the zoom level of that camera. Nevertheless, we are still interested in local decisions being made such that they perform well against the three global metrics: high **confidence** of those objects which are tracked, high **proportion** of owned objects tracked, and high **overlap**. It is therefore necessary to translate these global metrics into local camera-level analogues.

The local confidence is the average confidence of the currently owned and visible objects to the respective camera. In our simulation we use the inverse Euclidean distance between the camera and the tracked object, normalised by the maximum distance a camera could zoom out to. We defined our local confidence  $confidence_L(c_i)$  for a camera  $c_i$  in equation 1.

Adapting the FOV of cameras gives rise to a local trade-off between achieved average confidence of all currently tracked objects and the proportion of objects tracked by this camera at the given time. The proportion is defined by the objects owned



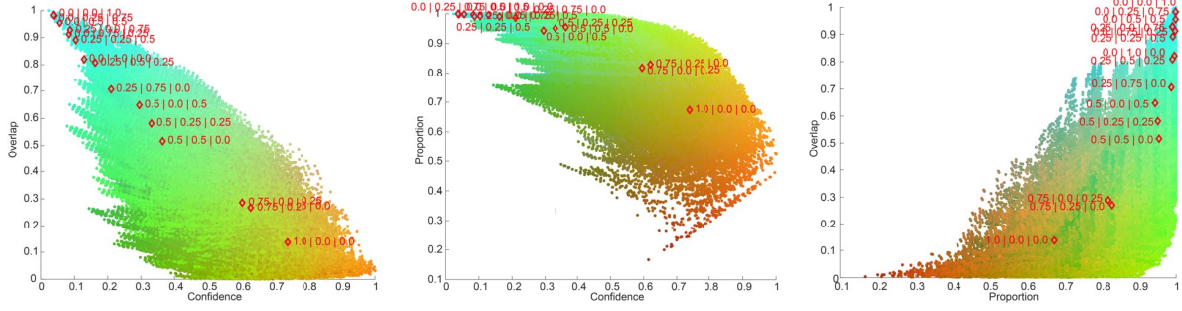


Fig. 7. From left to right the figures show overlap against confidence, proportion against confidence, and proportion against overlap for scenario 8. The results have been obtained in 30 independent runs and the mean values, normalised by the maximum outcomes, are shown. Each result is colour-coded: confidence is represented by ratio of red, proportion is represented by ratio of green, and overlap is represented by ratio of blue. Additionally, the results of the bandit solver have been plotted as red diamonds. The numbers next to the diamond represent the  $\alpha$ ,  $\beta$ , and  $\gamma$  value of the reward function used for the bandit.

and visible to the camera divided by all owned objects.

Without having knowledge of the position and zoom level of the other cameras in the network, calculating the overlap cannot be done by using geometric information. To overcome this problem, we rely on cameras querying their own FOV exploiting trading-information of tracked objects from the initially discussed approach by Esterle et al. [7]. This allows to infer redundancies in observed areas for multiple cameras during runtime based on already available information from the autonomous trading of tracked objects. The FOV of a camera  $i \in C = \{1, \dots, n\}$  can be described as a 2-dimensional space represented by a set of points  $P_i = \{p_1, \dots, p_l\}$ . Each point  $p_k \in P_i$  within this space can be categorised in one of the classes  $\chi$ . Initially, a point  $p_k$  does not belong to any class of  $\chi$ . Actively observing an object enables a camera to interact with another camera on a common basis and therefore to classify the point occupied by the tracked object. By interacting with another camera, the classification function  $Q : P_i \rightarrow \chi$  is able to select an appropriate category for a given point. We define three categories for our classification:

- PRIVATE defines the set of points that are, at the time of the query, only observed by the querying camera.
- SHARED contains points that are, at the time of the query, observed by at least one other camera than the querying one. The querying as well as the queried camera store this information.
- OTHERS defines the set of points that are, at the time of the query, only observed by another camera.

With each query, the camera can potentially receive one out of these three different responses. The frequency of querying objects is based on the processing power of the individual camera as well as the preferences of the operator. Based on heuristics, having cameras perform 1 to 2 queries every second is feasible. We outlined this classification in Algorithm 1.

Knowing the location at which other cameras can see an object better, allows each individual camera to adapt its FOV. When reducing the FOV we receive two benefits: (i) due to the reduced observed space, the observed object is perceived in greater detail as the number of pixels used increases and (ii) the searched space for newly arriving objects or auctions can be reduced with respect to the observed environment. On the other hand, when increasing the FOV the individual camera covers a

---

**Algorithm 1** The FOV classification algorithm

---

- 1) *Object trading of camera  $i$  as described in [7].*
  - 2) For each object  $o$  advertising to other cameras and object  $q$  advertised by other cameras, keep track of the location  $p_o$  and  $p_q$ .
    - a) In the case when no bids are received for an object  $o$ , add  $p_o$  to the set of PRIVATE locations in the FOV of camera  $i$ .
    - b) When receiving bids for object  $o$ , and  $o$  is within the FOV of  $i$  add  $p_o$  to the set of SHARED points of camera  $i$ .
    - c) When receiving bids for object  $o$ , and  $o$  is outside the FOV of  $i$  add  $p_o$  to the OTHERS set of camera  $i$ .
    - d) For those objects  $q$  which have been advertised to camera  $i$  and are visible to  $i$  but are not handed over, add  $p_q$  to the SHARED set of  $i$ .
  - 3) Repeat at regular intervals.
- 

larger space and can therefore track objects for a longer time, as well as possibly being able to cover more objects at the same time. Over time, each individual camera can therefore categorise its own FOV based on the received answers of its queries. We illustrated two of those classes in Figure 8, allowing for an approximation of the overlap using only locally available information. Using this local information allows the camera to infer its local overlap,  $overlap_L$ , with other cameras in the network. The relationship between SHARED and the sum of all identified points gives the camera an approximation of its current overlap with other cameras.

$$\begin{aligned}
 overlap_L &= \frac{|shared|}{|total|} \\
 &= \frac{|shared|}{|shared| + |private| + |others|}
 \end{aligned} \tag{6}$$

While the trade-off between the different objectives was presented in Section V, tasking an operator with choosing a configuration for the zoom levels will most likely not result in an solution which balances the competing objectives efficiently. This issue becomes trickier still when the operator has certain preferences to be fulfilled by the network, and finding a particular region of the resulting Pareto frontier is desired. In order to enable our cameras to achieve an efficient solution at runtime, we employ multi-armed bandit problem solvers

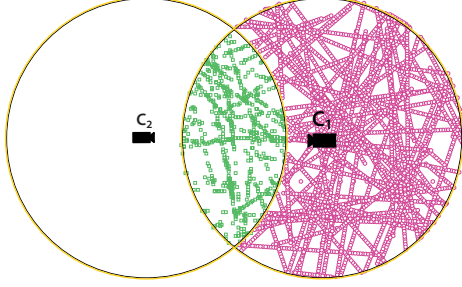


Fig. 8. Illustration of two omnidirectional cameras  $c_1$  and  $c_2$  with overlapping FOVs (in black/yellow) with queried observation points within the FOV of  $c_1$ . Pink circles in the FOV of  $c_1$  refer to queried locations which are private to  $c_1$ , green squares refer to locations within a overlapping FOV.

locally to trade off overlap, proportion of tracked objects, and accumulated confidence of all tracked objects. Based on its previous success in similar decentralised learning tasks [12], we use the well known multi-armed bandit problem solver *SoftMax* [23]. A ‘temperature’ value of 0.1 was used, since this performed well in initial profiling experiments. The bandit solver selects an arm, representing a predefined zoom level, based on previous rewards. The temperature governs how an arm’s expected reward influences its probability of selection. The reward function for a selected zoom level is defined by the local metrics of the individual camera given in equation 7.

$$\text{reward} = \left( \alpha * \frac{\sum_{t=1}^{t_c} \text{confidence}_L(t)}{t_c} + \beta * \frac{\sum_{t=1}^{t_p} \text{proportion}_L(t)}{t_p} + \gamma * \frac{\sum_{t=1}^{t_o} \text{overlap}_L(t)}{t_o} \right) \quad (7)$$

where  $t_c$  the accumulated number of objects the camera has tracked over time,  $t_p$  is the number of  $\text{proportion}_L$  values recorded, and  $t_o$  the number of timesteps where an  $\text{overlap}_L$  has been calculated. This allows the individual to normalise its reward by the actual measurements the cameras has made over time. Using  $\alpha$ ,  $\beta$ , and  $\gamma$ , where  $\alpha + \beta + \gamma = 1$ , we provide an operator with a simple handle to trade-off the three different metrics in the bandit solver. In order to allow a camera to learn about its environment and give a reasonable chance for objects to pass through its FOV, the bandit solver selects a zoom level only every 10 timesteps. In practice, this value would need to be adapted based on the object arrival frequency.

Figure 6 gives an overview of the trade-offs and the performance of the bandit solver used by the individual cameras in scenario 3. It is important to note that the cameras only use local information approximating global properties to drive their adaptation, while global information is used to plot their performance. Nevertheless, we are able to explore the available solution space using our weights ( $\alpha$ ,  $\beta$  and  $\gamma$ ) on total confidence, overlap and proportion of tracked objects respectively. It is important to keep in mind, that these illustrations are 2-dimensional representations of 3-dimensional plots. This means, a result of the bandit solver may not be on the Pareto-frontier of the results of the bandit solver in one

2-dimensional representation but is on the frontier in another solution space (e.g. the point  $\alpha = 0.0$ ,  $\beta = 0.5$ , and  $\gamma = 0.5$  is not on the frontier in the comparison between proportion and confidences (6, left) nor proportion and overlap (6, middle) but in the comparison between overlap and confidence (6, right)).

Figure 7 gives an overview of the trade-off between the metrics in scenario 8. This scenario contains 7 randomly deployed cameras and with 6 different zoom levels for each camera, resulting in 279936 different possible variations in how to set these zooms. Using our multi-armed bandit problem solver approach, the cameras are able to select their own zoom level based on an operator’s preferences. While the bandit solvers are not able to achieve extreme outcomes, they are able to select configurations during runtime reflecting the defined preferences. This is again based on the difficulty of capturing the global goals of overlap/redundancy, proportion of tracked objects, and total confidence by using only local information. This is an instance of the problem of decomposing global goals into local rewards, when assuming decentralisation, as discussed in a related problem by Lewis et al. [11], [12]. It is also important to note that our results are not perfectly even distributed in the solution space. This is due to the problem of selecting weight values for preferences, when global state information is not available to local learners. Van Moffaert et al. [17] discuss some of the issues involved in weight selection, given varying assumptions about the available information.

## VII. RESILIENCE AND SELF-HEALING

In the previous section, we showed how cameras can use multi-armed bandit problem solvers to manage the redundancy of their FOVs during runtime, in order to balance confidence, proportion of tracked objects, and overlap, according to an operator’s preferences. While the trade-off between the proportion of tracked objects and tracking confidence is reflective of the tension between objectives relating to the tracking task, the overlap between FOVs has an important impact on the resilience of the network, in the case of failing cameras.

To illustrate this we introduce the simple scenario 11 with only two cameras. While the upper camera can only observe the upper objects moving around, the lower camera is able to observe all objects in the environment with its largest zoom level. When the upper camera fails after 1000 timesteps, focussing on proportion or confidence does not mitigate the impact on global proportion. In Figure 9 we show not only that focussing on overlap reduces the drop when measuring proportion but also allows the remaining camera to perform better in comparison to when focussing on proportion alone.

In scenario 12 we simulate a corridor with cameras mounted on both sides (see Figure 3 for an illustration). While a single side would suffice to cover the entire corridor, having additional cameras can allow for either increased tracking confidence brought about by narrower zooms, or increased redundancy brought about by overlapping FOVs, if required. We remove 2 cameras (the middle one from the left column and the upper one on the right) after 1000 timesteps. Figure 10 illustrates the zoom level after 2000 timesteps determined by the bandit solvers based on local information available to the visual sensor nodes. In this example, the bandit solvers only focussed on a single objective rather than a weighted sum of



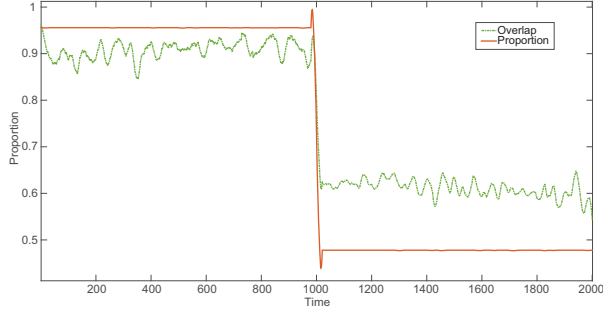


Fig. 9. Measuring proportion over 2000 timesteps, smoothed result using a least-squares smoothing filter. A camera fails after 1000 timesteps. Focussing on **overlap** achieves a better result than focussing on **proportion** after a node failure occurred.

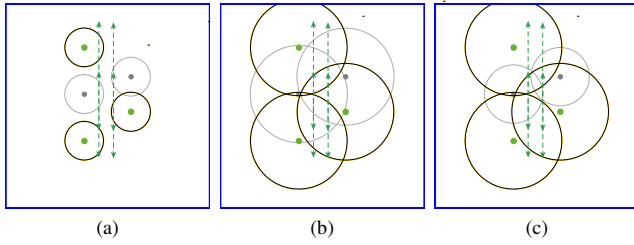


Fig. 10. The different zoom levels of the cameras after 2000 timesteps in a single exemplary test run. From left to right shows a focus on confidence (a), overlap (b) and proportion (c).

the three objectives. Grey dots and corresponding grey circles indicate cameras are not participating in the network anymore. Nevertheless, the size of the corresponding circle shows the last zoom level for the respective camera.

In Figure 10a the bandit solvers focus on confidence alone, resulting in smaller zoom levels and only covering the left line of people moving up and down the corridor. In such a setting (which ignores the proportion of total objects tracked), cameras receive more reward for focussing on objects for a short time, rather than having a wider FOV which would allow them to track objects for a longer period of time. In Figure 10b, the cameras increased their FOV as much as possible in order to maintain a high overlap with other cameras. In Figure 10c the bandit solver focussed on maximising the proportion of tracked objects and therefore widens the zooms of the cameras until all objects are covered by at least one camera. While after 2000 time steps the remaining cameras have the same zoom levels as when focusing on overlap, this was not the case before the failure occurred. This is indicated by the last zoom level of the failed cameras which is much smaller than those when focussing on overlap.

While our proposed approach enables the cameras to adapt to changes during runtime based on local knowledge and an operator’s preferences, we are also interested in the network’s resilience after a change occurred. We therefore defined a global performance measurement which captures the case when an operator’s preferences between the two task-related objectives (maximising confidence and maximising proportion) are equal, and overlap is not considered. This is defined in

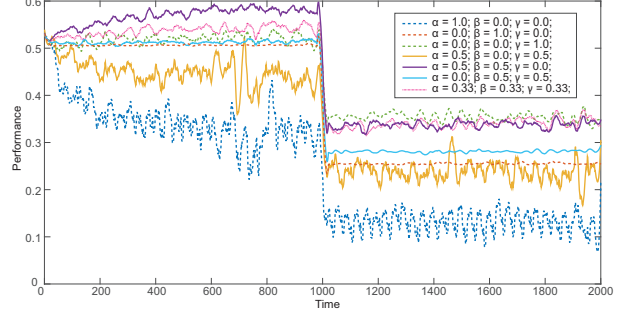


Fig. 11. Task-related performance as defined by equation 8 for three extreme sets of weights in the reward functions, three heterogeneous sets of weights, and evenly balanced weights for scenario 11 with a failing camera. Results have been smoothed using a least-squares smoothing filter.

equation 8.

$$performance = \frac{confidence_G + proportion_G}{2} \quad (8)$$

We measured this performance over the entire duration of 2000 timesteps of scenario 11 where the the upper camera failed after 1000 timesteps. Figure 11 illustrates the highlights of bandit solvers solely focussing on overlap, proportion and confidence respectively as well as combinations of two of these metrics. Finally we also illustrate the performance when evenly distributing the weight among all three metrics. It is not with surprise to note that a focus on a combination between proportion and confidence initially performs best. Nevertheless, focusing on overlap allows the system to perform better than any other approach after the camera fails. This is since the lower camera tries to generate overlap and is able to take over objects immediately after the failure. Only after some time, the strategy focussing on confidence and proportion is able to achieve similar performance as focussing on overlap. A similar result is achieved by the bandit solvers focussing on all three metrics evenly.

## VIII. CONCLUSIONS

In this paper we proposed a mechanism for the self-organisation of zoom factors and the resulting FOVs in a distributed visual sensor network, according to factors associated with attention and redundancy. Continuous adaptation directs attention towards objects of interest, optimising sensor coverage according to the areas which are most important to observe, as the scenario changes. Decentralised learning, where camera sensors learn their own behaviour independently based on local information, is able to achieve efficient outcomes in a multi-objective solution space comprising objectives to (i) maximise the tracking confidence for those objects which are tracked, (ii) maximise the proportion of tracked objects, and (iii) maximise the overlap between FOVs. Though cameras could simply use their maximum zoom level in order to cover as much ground as possible, this would result lower confidence on the tracked objects due to the reduced pixel count. On the other hand, reducing the zoom level and focusing on the local environment, the network as a whole will require more time to recover in the case of camera failures. We explored this trade-off and presented an online learning approach, using

only information available locally, able to balance an operator's preferences between the three stated global objectives.

A resulting and challenging problem is the decomposition of global objectives into local reward functions, in the absence of global information, and such that decentralised learning based on local information drives the global system towards desired outcomes. We identified this challenge and presented an approximation appropriate to this application domain, allowing us to demonstrate the feasibility of the overall approach. We were able to show, even with such an approximation that this learning approach allows the cameras to self-organise in order to fulfil an operator's preferences, in a dynamic environment.

Finally we introduced uncertainties to the network itself, allowing cameras to be removed during operation. We capture the trade-off between achieved performance throughout the run with built in self-healing capabilities through ongoing learning, against anticipation of such failures through increased redundancy. This anticipation capability leads to lower impact when failure occurs, as well as faster recovery. This is achieved in a self-organising and fully decentralised manner, according to tunable preferences, using decentralised online learning.

There are still various open questions to be tackled in our future work. How can a camera make sure objects are not lost by the network when the currently tracking camera explores other zoom levels? This becomes especially important when the camera reduces its FOV which may lead to not having the object within its FOV any more. An interesting question also arises on the interplay between the different zoom levels of cameras and the ability of the network to track as many objects as possible. Another direction of future research would compare the performance of our approach to the performance of coverage optimisation algorithms adapted so as to fit our objectives. Finally, we presented our work based on omnidirectional cameras which one might consider a simplification. Indeed, mapping our work on unidirectional cameras able to adapt the orientation as well as their zoom of the FOV is the next step on our roadmap.

#### ACKNOWLEDGMENT

This research was supported by the research initiative Mobile Vision, with funding from the Austrian Institute of Technology and the Austrian Federal Ministry of Science, Research and Economy HRSM program.

#### REFERENCES

- [1] N. Capodiceci, E. Hart, and G. Cabri, "Artificial Immune System Driven Evolution in Swarm Chemistry," in *Proc. of the Int. Conf. on Self-Adaptive and Self-Organizing Systems*. IEEE Press, 2014, pp. 40–49.
- [2] D. G. Costa and L. A. Guedes, "The Coverage Problem in Video-Based Wireless Sensor Networks: A Survey," *Sensors*, vol. 10, no. 9, pp. 8215–8247, 2010.
- [3] B. Dieber, C. Micheloni, and B. Rinner, "Resource-Aware Coverage and Task Assignment in Visual Sensor Networks," *Trans. on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1424–1437, 2011.
- [4] X. Du, M. Zhang, K. Nygard, M. Guizani, and H.-H. Chen, "Distributed Decision Making Algorithm for Self-Healing Sensor Networks," in *Proc. of the Int. Conf. on Communications*, vol. 8. IEEE Press, 2006, pp. 3402–3407.
- [5] 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 Press, 2013, pp. 19–20.
- [6] L. Esterle, P. R. Lewis, B. Rinner, and X. Yao, "Improved Adaptivity and Robustness in Decentralised Multi-Camera Networks," in *Proc. of the Int. Conf. on Distributed Smart Cameras*. IEEE Press, 2012, pp. 1–6.
- [7] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, "Socio-Economic Vision Graph Generation and Handover in Distributed Smart Camera Networks," *Trans. on Sensor Networks*, vol. 10, no. 2, 2014, 20.
- [8] N. Ferrier, "Active Control of Resolution for Stable Visual Tracking," in *Sensor Based Intelligent Robots*, ser. Lecture Notes in Computer Science. Springer, 1999, vol. 1724, pp. 140–159.
- [9] S. Galzarano, G. Fortino, and A. Liotta, "Embedded Self-healing Layer for Detecting and Recovering Sensor Faults in Body Sensor Networks," in *Proc. of the Int. Conf. on Systems, Man, and Cybernetics*. IEEE Press, 2012, pp. 2377–2382.
- [10] M. Hoffmann, M. Wittke, J. Hähner, and C. Müller-Schloer, "Spatial Partitioning in Self-Organizing Smart Camera Systems," *Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 480–492, 2008.
- [11] P. R. Lewis, L. Esterle, A. Chandra, B. Rinner, and X. Yao, "Learning to be Different: Heterogeneity and Efficiency in Distributed Smart Camera Networks," in *Proc. of the Int. Conf. on Self-Adaptive and Self-Organizing Systems*. IEEE Press, 2013, pp. 209–218.
- [12] —, "Static, dynamic and adaptive heterogeneity in distributed smart camera networks," *Trans. on Autonomous and Adaptive Systems*, vol. 10, no. 2, 2015, 8.
- [13] P. R. Lewis, P. Marrow, and X. Yao, "Resource Allocation in Decentralised Computational Systems: An Evolutionary Market Based Approach," *Autonomous Agents and Multi-Agent Systems*, vol. 21, no. 2, pp. 143–171, 2010.
- [14] L. Massoulie, A.-M. Kermarrec, and A. Ganesh, "Network Awareness and Failure Resilience in Self-organizing Overlay Networks," in *Proc. of the Int. Symp. on Reliable Distributed Systems*, 2003, pp. 47–55.
- [15] A. Mavrinac and X. Chen, "Modeling Coverage in Camera Networks: A Survey," *Int. Journal of Computer Vision*, vol. 101, no. 1, pp. 205–226, 2013.
- [16] R. Misra and C. Mandal, "Self-Healing for Self-Organizing Cluster Sensor Networks," in *Proc. of the Annual IEEE India Conference*. IEEE Press, 2006, pp. 1–6.
- [17] K. V. Moffaert, T. Brys, A. Chandra, L. Esterle, P. R. Lewis, and A. Nowé, "A Novel Adaptive Weight Selection Algorithm for Multi-Objective Multi-Agent Reinforcement Learning," in *Proc. of the Annual Int. Joint Conf. on Neural Networks*. IEEE Press, 2014, pp. 2306–2314.
- [18] Y. Morsly, N. Aouf, M. S. Djouadi, and M. Richardson, "Particle Swarm Optimization Inspired Probability Algorithm for Optimal Camera Network Placement," *Sensors*, vol. 12, no. 5, pp. 1402–1412, 2012.
- [19] C. Picciarelli, L. Esterle, A. Khan, B. Rinner, and G. Foresti, "Dynamic Reconfiguration in Camera Networks: a short survey," *Trans. on Circuits and Systems for Video Technology*, pp. 1–13, 2015, to appear.
- [20] B. Rinner and W. Wolf, "Introduction to Distributed Smart Cameras," *Proc. of the IEEE*, vol. 96, no. 10, pp. 1565–1575, 2008.
- [21] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks," *Proc. of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.
- [22] R. Sterritt, "Towards Autonomic Computing: Effective Event Management," in *Proc. of the Annual Software Engineering Workshop*. IEEE Press, 2002, pp. 40–47.
- [23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [24] N. Vljajic and N. Moniz, "Self-Healing Wireless Sensor Networks: Results That May Surprise," in *Proc. of the Globecom Workshops*. IEEE Press, 2007, pp. 1–6.
- [25] W. Wolf, B. Ozer, and T. Lv, "Smart Cameras as Embedded Systems," *Computer*, vol. 35, no. 9, pp. 48–53, 2002.
- [26] Y. Zhang, J. Zhang, L. Zhang, M. Li, J. Yang, and L. Guo, "A Survivability Information System Based on Service Self-Organization," in *Proc. of the Int. Conf. on Internet Computing for Science and Engineering*, 2010, pp. 158–161.