

Learning to be Different: Heterogeneity and Efficiency in Distributed Smart Camera Networks

Peter R. Lewis*, Lukas Esterle[†], Arjun Chandra[‡], Bernhard Rinner[†] and Xin Yao*

*CERCIA, School of Computer Science, University of Birmingham, United Kingdom

[†]Institute of Networked and Embedded Systems, Alpen-Adria Universität Klagenfurt and Lakeside Labs, Austria

[‡]Department of Informatics, University of Oslo, Norway

{p.r.lewis|x.yao}@cs.bham.ac.uk, {lukas.esterle|bernhard.rinner}@aau.at, chandra@ifl.uio.no

Abstract—In this paper we study the self-organising behaviour of smart camera networks which use market-based handover of object tracking responsibilities to achieve an efficient allocation of objects to cameras. Specifically, we compare previously known homogeneous configurations, when all cameras use the same marketing strategy, with heterogeneous configurations, when each camera makes use of its own, possibly different marketing strategy. Our first contribution is to establish that such heterogeneity of marketing strategies can lead to system wide outcomes which are Pareto superior when compared to those possible in homogeneous configurations. However, since the particular configuration required to lead to Pareto efficiency in a given scenario will not be known in advance, our second contribution is to show how online learning of marketing strategies at the individual camera level can lead to high performing heterogeneous configurations from the system point of view, extending the Pareto front when compared to the homogeneous case. Our third contribution is to show that in many cases, the dynamic behaviour resulting from online learning leads to global outcomes which extend the Pareto front even when compared to static heterogeneous configurations. Our evaluation considers results obtained from an open source simulation package as well as data from a network of real cameras.

Keywords-distributed smart cameras; heterogeneity; variation; learning; self-organisation;

I. INTRODUCTION

In many cases, collective systems are designed in such a way that the component nodes are given a local objective to pursue, and the system wide behaviour is a product of the actions and interactions of the component nodes. In pursuing local objectives, nodes are typically endowed with a common algorithm or behavioural strategy. However, nodes are often located in different areas, having different views of the world, and are subject to different experiences. In these cases, nodes may be better off adopting different algorithms from each other, in order to better achieve their own local objectives. It has also been shown that such heterogeneity among nodes can lead to better achievement of system wide objectives [1], [2], especially when nodes can adapt independently in response to uncertainty and changes in the environment during the system’s lifetime [3].

In this paper we study the effect of heterogeneity among

nodes in a distributed smart camera network. Smart cameras are fully computationally capable devices endowed with a visual sensor, and typically run computer vision algorithms to analyse captured images. Where standard cameras can only provide plain images and videos, smart cameras can pre-process these videos and provide users with aggregated data and logical information. Since smart cameras are designed to have a low energy footprint, their processing capabilities are also low. Therefore, typically each object of interest is tracked by only one camera at a time. Communication between cameras allows the network as a whole to track objects in a distributed fashion, handing over object tracking responsibilities from camera to camera as objects move through the environment. In previous work [4] we showed that by endowing cameras with self-interested agents, which traded responsibilities for tracking objects of interest in a market, the network as a whole could achieve an efficient allocation of objects to cameras, without any central coordination or *a priori* knowledge about the network topology. To sell objects, cameras host Vickrey auctions, sending auction invitations to other cameras, who may then bid to buy objects. The cameras use pheromone-based online learning to determine which other cameras they trade with most often. This neighbourhood relationship graph (the *vision graph*), enables them to selectively target their auction invitations and achieve higher levels of efficiency.

Six different behavioural strategies were used by camera nodes, which determined the level of marketing activity they undertook, given the learnt vision graph. Some strategies incurred higher levels of communication overhead but typically obtained higher levels of tracking confidence; other strategies obtained the opposite results. However, the trade-off realised by each strategy was found to be highly scenario dependent; as camera positions varied and object movements differed, the relative benefits of the strategies was greatly influenced. Additionally, cameras often operated inefficiently since the homogeneous deployment of strategies forces a *one size fits all* approach, despite local differences in the vicinities of the cameras. By permitting heterogeneity between cameras in terms of their strategies, this paper shows that more Pareto efficient global outcomes can be obtained.

Although heterogeneity can improve global efficiency, given the virtually limitless possibilities for camera network deployments and accompanying environmental dynamics, identifying by hand the most appropriate behaviour for each node in a given scenario and at a particular point in time is not feasible. To overcome this problem we propose using online learning algorithms, specifically multi-armed bandit problem solvers (e.g. [5]) within each camera to learn the appropriate strategy for each node during runtime. These so-called bandit solvers balance exploitation behaviour, where a camera achieves high performance by using its currently known best strategy, with exploration, where the camera explores the effect of using other strategies to build up its knowledge. By employing bandit solvers in each camera, we are able to obtain near Pareto efficient global outcomes in many cases. In other cases, the dynamic nature of the online learning algorithms actually extends the Pareto efficient frontier, improving upon the best static configurations. While in principle there are many possible marketing strategies which could be conceived of and used by a bandit solver, for comparison purposes, in this paper we focus on the six strategies in [4]. These are recapitulated in section III. To evaluate the effects first of allowing heterogeneity and secondly introducing online learning, we make use of several qualitatively different deployments of smart camera networks, both in simulation and using real video data.

The rest of the paper is structured as follows. In section II we summarise recent work investigating heterogeneity and inter-agent variation in self-organising systems. In section III we describe the problem studied in more detail as well as the different scenarios being used for evaluation. In section IV we show how heterogeneity can improve the global system performance in our problem. In section V the online learning approach is introduced and we present results from its use in our simulated scenarios as well as a real camera network. We conclude the paper and discuss future work in section VI.

II. RELATED WORK

Nature provides numerous examples of heterogeneity (or variation or diversity) enabling populations to successfully self-organise to achieve their objectives [1]. When using self-organisation to engineer decentralised collective systems, differences between system components can also be an important factor in enabling the collective to obtain high performance [1], [6]. Heterogeneity in sensor networks may take on various forms. Some of those which may be imagined include variation of hardware between nodes, differences in behaviour, and diverse parameters or objectives. In engineering such systems, the challenge is to find self-organisation algorithms which give rise to optimal forms of such heterogeneity, which in turn lead to high performance at the global level. Prasath et al. [6] highlight two key issues:

- 1) Whether heterogeneity allows optimisation beyond that possible in the homogeneous case, and

- 2) What algorithms to use to achieve near-optimal heterogeneous networks.

Campbell et al. [1] investigated the effect of inter-agent variation on a multi-agent task allocation problem, showing that such variation creates more possible *organisations* (configurations) of the system. This larger configuration space provides more possibilities, some of which may enable a collective system to better achieve its goal.

The heterogeneity considered by Prasath et al. [6] is in the out-degree and wireless communication radius of nodes. They permit only two possibilities for each node's configuration, and compare the effect of using three different cooperative algorithms for determining node types, comparing the outcomes with ideal best possible outcomes. Rojković et al. [7] present a technique for assigning roles to different nodes in a sensor network, which is compared with the near-optimal solution found by a genetic algorithm with global knowledge. Nakamura et al. [8] reactively assign roles for data routing to different sensor nodes based on events to save energy during idle periods. Römer et al. [9] propose the adaptation of nodes' roles based on their location and purpose. This adaptation is done using a predefined set of rules which are the same for all nodes in the network. In smart camera networks, Dieber et al. [10] adapt the number of cameras in the network, changing their settings and the tasks being assigned to the cameras. They use a combination of an expectation-maximisation algorithm and evolutionary algorithm to satisfy predefined constraints.

Salazar et al. [3] highlight the importance of dynamic heterogeneous configurations for sensor networks deployed in uncharted environments, i.e. in scenarios about which there is a lack of *a priori* information. They argue that, in response to environmental changes over time, nodes should be able to reconfigure themselves according to local events, possibly in different ways from each other. Anders et al. [2] also study the effect of inter-agent variation on the performance of a self-organising system in an uncertain environment. They found that in two algorithms, one based on schooling fish and the other on honey bees, the performance of the algorithms obtained a higher performance with heterogeneity. Their results suggest the presence of a critical threshold, a particular amount of variation required to ensure near-optimal solutions. They also found that in some cases too much variety could lead to negative effects such as oscillatory behaviour or slower arrival at the solution.

III. PROBLEM STATEMENT

Although cameras make decisions based on local information, we are primarily interested in performance at the global level. This consists of two network-level measurements:

- 1) *Tracking confidence*, the achieved tracking confidence during a small time window for each object by the camera tracking that object, summed over all objects.

- 2) *Number of auction invitations*, the number of invitations sent by all cameras as a result of auction initiations, during a small time window, a proxy for communication and processing overhead.

The tracking confidence for an object is the posterior probability obtained from the tracking algorithm (a classifier). In our simulation, if the object is in the FOV of the camera, this is simply the inverse Euclidean distance of the object from the camera, normalised to $[0, 1]$. For the real camera results, confidence was obtained from the real tracker, which uses SIFT [11].

While these measurements report instantaneous performance, we are interested in the online performance of the network over time. Each metric is therefore the summation of these measurements over the lifetime of the deployment. We therefore have two conflicting objectives: to maximise the tracking confidence metric while minimising the number of auction invitations metric. By considering these objectives separately, we are able to obtain results in a two-dimensional objective space, which represent different points on the trade off between the two. An operator may then choose between different configurations leading to Pareto efficient outcomes, based on their relative preference between the objectives.

In [4], based on the trading behaviour of the network, the cameras learn a vision graph describing the spatial relationships between their fields of view (FOVs). This learning used a technique inspired by the pheromone altering mechanisms governing foraging behaviours in ant colonies. A trade between cameras strengthens the link between them from their individual perspectives, whilst cameras which no longer trade weaken their links over time, eventually forgetting them. Building on such a vision graph, cameras are able to reduce their communication without significantly sacrificing tracking performance. Specifically, the cameras could employ one of six possible strategies for marketing objects to other cameras. Two auction initiation schedules, combined with three communication policies, give us six marketing strategies. The auction initiation schedules are:

- 1) *ACTIVE*, in which a camera advertises each object for handover every time it calculates the tracking confidence associated with the object.
- 2) *PASSIVE*, in which a camera advertises each object for handover when that object is about to leave its FOV.

A camera combines one of the above auction initiation schedules with one of the following communication policies:

- 1) *BROADCAST*, which communicates with all available cameras in the network. This approach does not miss any camera but also generates a high overhead since it includes cameras which are not likely to respond.
- 2) *SMOOTH*, in which the probability of communicating with another camera is based on the ratio between its link strength and that of the strongest link in its graph.
- 3) *STEP*, in which an advertisement is sent to a camera

if the strength of the link to that camera is above a certain threshold, otherwise communicating with that camera with a very low probability.

However, in [4] the same marketing strategy (e.g. *ACTIVE SMOOTH*) was employed by all the cameras in the network for the lifetime of a deployment. In this paper, we consider the cases where (i) all cameras need not employ the same strategy at all times, and (ii) a camera may learn which strategy to use during runtime. We therefore refer to the strategies employed by the cameras across the network as the configuration of the network. Based on the variation in the employed strategies across the network, we may describe two types of configurations:

- 1) *Homogeneous*: A network configuration where all cameras use the same marketing strategy.
- 2) *Heterogeneous*: A network configuration where at least two cameras use different marketing strategies.

We have refined Prasath et al's [6] key issues for engineering heterogeneity in self-organising systems, to fit the context of market-based distributed smart camera networks. Our research questions are therefore as follows:

- 1) Do heterogeneous configurations enable outcomes which are more Pareto efficient than those possible in the homogeneous case?
- 2) How can a decentralised network of self-interested smart cameras self-organise to a Pareto efficient configuration, given a particular scenario?

For the purposes of our evaluation, a scenario comprises a set of cameras with associated positions and orientations, along with a set of objects and their movement paths through the environment. In this paper, we simulate and evaluate configurations within eleven qualitatively different scenarios using our open source CamSim¹ software. We also acquired video feed data from a real smart camera network, which gives us a twelfth scenario. All simulated scenarios are depicted in figure 1, while the snapshots from the video-based scenario are shown in figure 11. A summary of all scenarios is given in table I. Unless otherwise stated, in all experiments reported in this paper, each scenario was run for 1000 discrete time steps (each corresponding to one measurement time window). Due to stochasticity, 30 independent runs were conducted for each evaluation.

IV. PARETO EFFICIENCY OF HETEROGENEOUS NETWORKS

Despite the six available marketing strategies presented in [4], all cameras in each network used the same strategy, i.e. the networks were homogeneous. In this section we relax this unnecessary restriction, considering the case when individual nodes (cameras) in a network can use different strategies to govern how they advertise their auctions.

¹CamSim is available at <https://github.com/EPiCS/CamSim>. All scenarios are available within the CamSim repository.

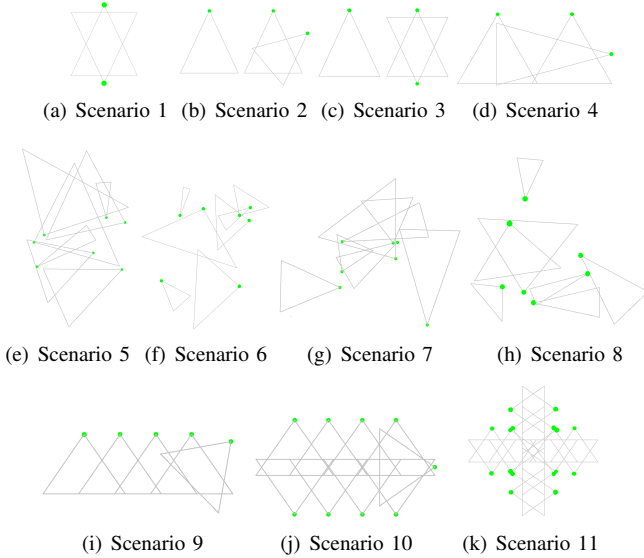


Figure 1. The scenarios tested with our simulation tool *CamSim*. A dot represents a camera, the associated triangle represents its FOV.

| ID | No. of Cameras | No. of Objects | Object Movement Path | Timesteps | No. of Possible Configurations |
|----|----------------|----------------|----------------------|-----------|--------------------------------|
| 1 | 2 | 4 | Random | 1000 | 36 |
| 2 | 3 | 11 | Random | 1000 | 216 |
| 3 | 3 | 4 | Random | 1000 | 216 |
| 4 | 3 | 4 | Random | 1000 | 216 |
| 5 | 7 | 9 | Random | 1000 | $\sim 2.7 \times 10^5$ |
| 6 | 7 | 9 | Random | 1000 | $\sim 2.7 \times 10^5$ |
| 7 | 7 | 9 | Random | 1000 | $\sim 2.7 \times 10^5$ |
| 8 | 7 | 9 | Random | 1000 | $\sim 2.7 \times 10^5$ |
| 9 | 5 | 3 | Predefined | 1000 | 7,776 |
| 10 | 9 | 1 | Predefined | 1000 | $\sim 1.0 \times 10^7$ |
| 11 | 16 | 5 | Predefined | 1000 | $\sim 2.8 \times 10^{12}$ |
| 12 | 5 | 1 | Predefined | 7120 | 7,776 |

Table 1

SUMMARY OF SCENARIOS USED IN OUR STUDY. A RANDOM OBJECT MOVEMENT PATH MEANS THAT EACH OBJECT MOVES IN A STRAIGHT LINE UNTIL IT REACHES THE BORDER OF THE SIMULATION AREA AND BOUNCES BACK WITH A RANDOMLY CHOSEN VECTOR. A PREDEFINED OBJECT MOVEMENT PATH MEANS THAT EACH OBJECT FOLLOWS A PREDETERMINED PATH THROUGH THE SIMULATION AREA.

Permitting this heterogeneity in the network enables nodes to specialise to their local situation and has the effect of permitting a wider range of global outcomes than was possible in the homogeneous case. As will be shown in this section, this can lead to the global performance of the network being strictly better in terms of both the considered objectives, extending the Pareto efficient frontier.

However, heterogeneity itself does not necessarily lead to better outcomes. It is also possible that nodes specialise wrongly, leading to a strictly worse global outcome than was possible in any homogeneous case. Indeed, when considering all possible heterogeneous configurations for a given network, the number of configuration points increases greatly compared to the homogeneous-only case.

A. A Baseline Scenario

We first consider scenario 1, a baseline scenario with two cameras and four objects. Figure 2 shows the mean global performance, calculated over 30 independent runs. Each point represents the global outcome from one configuration over 1000 time steps, in terms of both metrics: its total tracking confidence and the number of auction invitations within the entire network. As in [4], values are normalised by those obtained by ACTIVE BROADCAST, since this obtains both the maximum possible confidence as well as sending the maximum possible number of auction invitations.

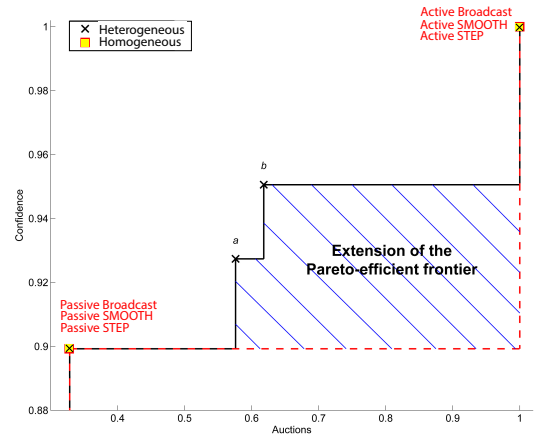


Figure 2. Results for a baseline scenario (scenario 1) with two overlapping cameras. The original Pareto frontier when homogeneity is enforced is depicted by the dashed line. The solid line indicates the newly extended Pareto frontier when heterogeneous configurations are permitted.

By enforcing homogeneity, as done previously in [4], we have six possible deployment options. The outcomes from these homogeneous configurations are depicted as squares. In this scenario, despite the six possible homogeneous configurations, there are only two extreme observed outcomes in the objective space, one favouring each objective. This is since, in some very simple scenarios, some strategies give rise to the same communication behaviour as each other; homogeneity does not permit any more balanced outcomes in this case. However, allowing the cameras to adopt different strategies from each other introduces new possibilities. When heterogeneous configurations are included, there are 36 possible deployment options. The heterogeneous configuration outcomes are depicted as crosses.

Outcomes *a* and *b* in figure 2 extend the Pareto efficient frontier, indicating new efficient configurations for tracking objects within the network. Additionally, both of these points lie on the newly extended Pareto frontier, since for each, no other outcome is better on both objectives. It is therefore clear from this example that heterogeneous configurations can lead to additional efficient outcomes.

B. More Complex Scenarios

In this section we consider more complex scenarios. We evaluated all six homogeneous configurations in all scenarios, and all possible heterogeneous configurations in scenarios 1 – 9. Due to the large number of cameras in scenarios 10 and 11, an exhaustive evaluation of all heterogeneous configurations was computationally infeasible.

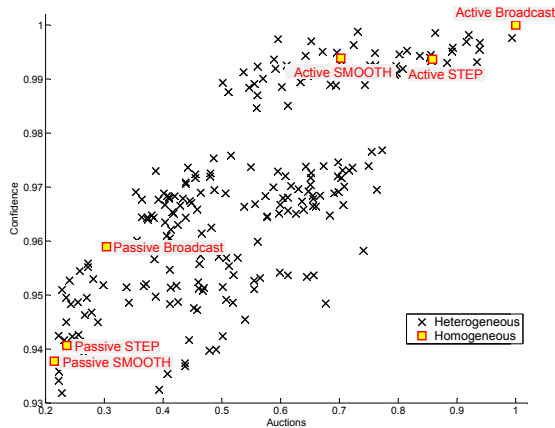


Figure 3. Performance for scenario 4 showing homogeneous and heterogeneous assignment of strategies. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each.

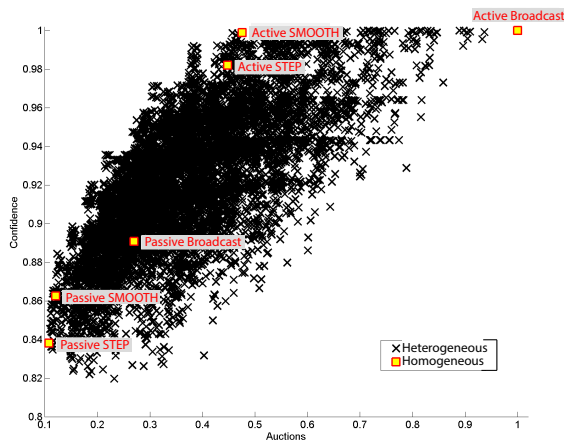


Figure 4. Performance for scenario 9 showing homogeneous and heterogeneous assignment of strategies. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each.

Figures 3 and 4 compare outcomes from heterogeneous and homogeneous configurations in scenarios 4 and 9 respectively. In these more complex scenarios, heterogeneous configurations led to many more outcomes in the objective space. In each case, the extension of the Pareto efficient frontier brought about by heterogeneity is also apparent. However, it is also clear that the outcomes of many heterogeneous configurations are dominated, and many are strictly worse than the original outcomes from the homogeneous

cases. Indeed, in all evaluated scenarios, when heterogeneous configurations of cameras are allowed, we observed system wide outcomes which both dominate and are dominated by outcomes from homogeneous configurations. In all cases, heterogeneity extended the Pareto efficient frontier.

V. DECENTRALISED ONLINE LEARNING OF PARETO EFFICIENT CONFIGURATIONS

In section IV, we showed that by permitting heterogeneous configurations of nodes, global outcomes may be obtained which are more Pareto efficient than in the homogeneous case. However, it is not clear which particular heterogeneous configuration should be chosen, in order to achieve an efficient global outcome, in a particular scenario. We are faced with the problem of choosing from numerous combinations, out of which only a few are Pareto efficient, which configuration the network should adopt.

This could be tackled as a classical offline search problem, as part of multi-camera calibration. However, doing so would assume that we know the characteristics of the scenario in advance, including camera placement and orientation, expected object movement patterns and runtime failures or additions (e.g. as studied in [12]). Indeed, this lack of *a priori* scenario knowledge is a key problem characteristic motivating this approach. Therefore, we extend the idea followed in our previous work, where individual cameras learn behaviours online during run time.

In this manner, a node’s strategy selection is made autonomously using a learning approach, which provides adaptation at runtime based on feedback from locally observed metrics: in this case the number of auction invitations sent by the node and its tracking confidence (as opposed to the equivalent metrics for the network as a whole).

A. Learning Efficient Configurations using Bandit Solvers

From the perspective of an individual camera, its task is to select a marketing strategy from those available, which maximises its expected tracking confidence while minimising its auction overhead, over time. We therefore consider that a camera is faced with an online algorithm selection problem [13]. Our approach is to consider this as an variant of the multi-armed bandit problem [5]. This problem is analogous to being faced with n slot machine arms, where each pull of an arm returns a random reward drawn from an unknown distribution associated with that arm. Given m total arm pulls, the task is to select which arms to pull such that the total reward obtained is maximised. If the player were to know the distributions behind each arm, then he can simply select the best arm for every pull. However, since the distributions are unknown, he must sample from each arm in order to gain some knowledge of each arm’s reward distribution. The multi-armed bandit problem therefore encapsulates the classic *exploration vs exploitation* dilemma. However, some of the assumptions present in the classic multi-armed bandit

problem formulation may not be appropriate in this setting. Firstly, the reward distributions are usually assumed to be static over time, and secondly it is assumed that the bounds on the obtainable rewards are also known. It does not appear that either assumption can be made in our problem.

Nevertheless, the bandit framework is useful and each marketing strategy can be considered an arm of a bandit. Each camera node can choose to use one strategy (i.e. pull an arm) at each time step, and can receive a resulting reward, derived from its local metrics. In this way, a camera learns which strategy performs well in its current situation within the scenario, and exploit that knowledge to maximise its performance. There are a number of so called bandit solving algorithms to be found in the literature. In this paper we consider three well known bandit solvers: the simple EPSILON-GREEDY [14], UCB1, which is known to perform well in static problems [5], and SOFTMAX [15]. Of these, EPSILON-GREEDY requires an ϵ value to determine the amount of exploration, UCB1 requires no parameters, while SOFTMAX uses a temperature value to govern how an arm’s expected reward influences its probability of selection.

In applying bandit solvers to algorithm selection at the local level in a self-organising system, we must define local reward functions, such that the global system’s objectives are achieved. In this case, this is further complicated by the presence of multiple objectives at the global level, and corresponding multiple metrics locally at each node. In this paper, we use a linear combination of the local metrics:

$$\text{reward} = \alpha \times \text{utility} - (1 - \alpha) \times \text{auction invitations} \quad (1)$$

where *utility* is the utility function given in [4], which sums obtained tracking confidence over all objects tracked by this camera, plus its balance of payments from all trading activity during this time step. The number of auction invitations sent by this camera at this time step is denoted by *auction invitations*. α allows us to change the node’s preference in favour of either maximising tracking confidence or minimising the number of auction invitations. Therefore, α may be used to direct local learning such that outcomes at the global level favour appropriate regions of the Pareto efficient frontier.

Unfortunately, due to complex interactions between trading, vision graph learning and bandit solvers, the local-global mapping assumed in this local reward function is likely to be over-simplistic. Though the results presented in this section are encouraging, more work is required in this area. Fundamentally, how to define multi-attribute bandit reward functions at the local level, such as to achieve Pareto efficient outcomes at the global level, remains an open question.

Figure 5 shows the outcomes in scenario 1, when configurations learnt using bandit solvers are compared with static homogeneous and heterogeneous configurations. For EPSILON-GREEDY, ϵ values of 0.1, 0.01 and 0.001 were tried. In all scenarios, with 1000 time steps, $\epsilon = 0.1$ obtained the most Pareto efficient outcomes and is therefore used in

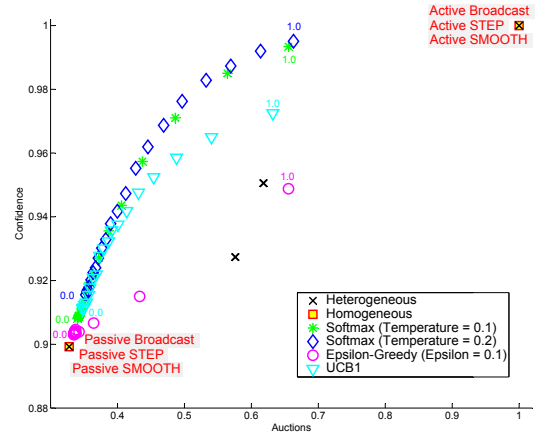


Figure 5. Performance for scenario 1 showing homogeneous and heterogeneous assignment of strategies as well as assignments done by bandit solvers. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each.

all results in this paper. Outcomes are shown for EPSILON-GREEDY, UCB1, and SOFTMAX, the latter with temperature values 0.1 and 0.2. For each bandit solver, results are shown when α is varied between 0 and 1 in intervals of size 0.05.

The results in figure 5 clearly show that configurations found with bandit solvers provide many more outcome points in the objective space than were possible in the static homogeneous and heterogeneous cases, and that many of these outcomes are highly Pareto efficient. Even though we presented the static heterogeneous configuration outcomes exhaustively, bandit solvers were able to obtain system wide outcomes which extend the Pareto efficient frontier obtained in the static heterogeneous case. This unexpected result suggests that dynamic configurations (i.e. those which change over time, in this case through online learning) can outperform even the best static heterogeneous configurations. This opens up an exciting prospect for future research.

B. Camera Level Normalisation by Distribution

In section V-A, we showed how bandit solvers can be used within cameras in order to select marketing strategies during runtime, appropriate to the scenario and the node’s preferences between objectives (in this case its α value). We also showed that by varying the α value, global outcomes ranged over the Pareto efficient frontier. However, it is also clear from figure 5 that the results from the bandit solvers cluster towards the lower left of the Pareto front, while outcomes in the upper right are more thinly spread.

This bias in outcomes is a result of the nature of the observed metrics at the camera level, and their combination in the local reward function. Ideally, α would be used to weigh the two objectives evenly, such that the outcome position on the Pareto frontier can be determined directly by setting α . E.g. an α value of 0.25 would lead to an outcome value 25% of the way along the length of the achieved front.

In order to achieve this, we would need to normalise the two components of the reward function. However, although a camera knows its own tracking confidence associated with an object, it cannot know what payment it might have received, had it advertised the object to a camera which it did not. The upper bound on the camera’s utility is therefore not known, and will vary significantly with every time step. Nevertheless, we are still able to mitigate the bias effect somewhat by attending to the second component of the reward function, the number of auction invitations issued by the camera. The upper bound on this value will also vary, but in this case only with the number of objects and other cameras currently known to the camera.

We are therefore able to perform some estimated normalisation of the number of auction invitations at the local level. Figure 7 shows the frequency distribution with which a camera sent auction invitations to other cameras over time, in a typical run of scenario 1. Clearly, cameras are less communicative more often than they are more communicative. As it turns out, this skew in the distribution appears to have a large effect on the bias observed in the outcome Pareto front. We are able to account for this skew effect by introducing a normalisation by distribution process into the auction invitation component of the local reward.

More specifically, each camera records the value of *auction invitations* for each time step throughout its lifetime. When a new value is observed, its rank within the historical values is calculated, and then scaled to be between 0 and 1. The original value is then added to the historical record. For example, if the new value is greater than the largest observed value so far, its normalised value is 1. Similarly, if a new value falls half way along the list of historically observed values, its normalised value is 0.5.

By normalising in this manner, we obtain a more even spread of outcomes along the achieved frontier. Figure 6 shows this for scenario 1, and can be compared with figure 5. A bias, though less pronounced, is still present with EPSILON-GREEDY and UCB1. This skewed distribution pervades all scenarios we evaluated, therefore we adopted this normalisation method in all subsequent experiments.

C. Learning in More Complex Scenarios

Figures 8, 9 and 10 show results for scenarios 3, 5 and 11 respectively. In all cases, bandit solvers used the normalisation by distribution method described in section V-B.

In each of these more complex scenarios, bandit solvers were able to obtain outcomes which extended the Pareto efficient frontier of the evaluated static configurations. This is particularly true of SOFTMAX (with both temperature values) and UCB1, all of which obtained a range of highly Pareto efficient outcomes. The spread of outcomes can be observed to vary depending on the particular scenario and the choice of bandit solver employed. The bias associated with EPSILON-GREEDY, and to a lesser extent UCB1 re-

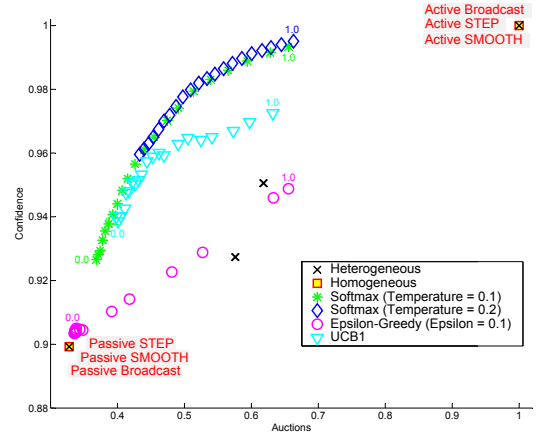


Figure 6. Performance for scenario 1 showing static homogeneous and heterogeneous assignment of strategies as well as assignments done by bandit solvers. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each. The bandit solvers’ reward functions normalised the number of auction invitations by distribution at runtime.

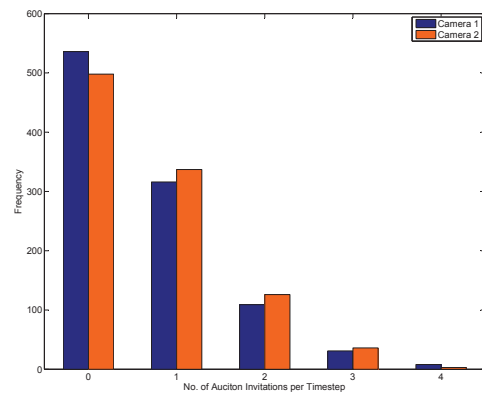


Figure 7. Histogram showing the frequency distribution of auction invitations sent per time step in scenario 1 when cameras use SOFTMAX-0.1. Each camera is represented by one bar per bin. $\alpha = 0.5$.

mains; outcomes from the other bandit solvers are evenly spread as α varies. Of all the bandit solvers, SOFTMAX typically obtains the best spread across the frontier.

D. Real Camera Network Results

We continued our evaluation using video feed data from a real smart camera network; this is referred to as scenario 12 in table I. A SURF-based tracking approach [11] was used to detect and track a person within the network of cameras. The SURF-based approach initially extracts SURF features from a model image and tries to re-identify these model features in the consecutive frames. The percentage of re-identified features represents the confidence of our SURF-based tracker. Figure 11 shows snapshots from each camera at five different points in time. Each camera captured 1780 frames, looped four times to create a total of 7120 frames, each with a resolution of 640×480 . When PASSIVE

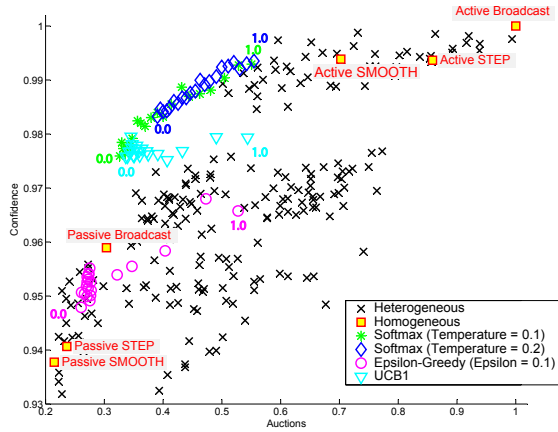


Figure 8. Performance for scenario 3 showing homogeneous and heterogeneous assignment of strategies as well as assignments done by bandit solvers. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each. The bandit solvers' reward functions normalised the number of auction invitations by distribution at runtime.

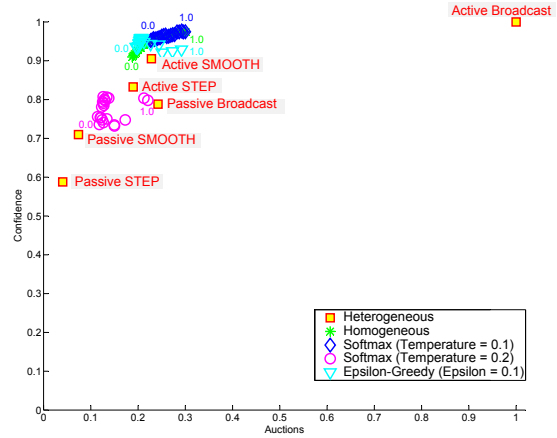


Figure 10. Performance for scenario 11 showing homogeneous of strategies as well as assignments done by bandit solvers. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each. The bandit solvers' reward functions normalised the number of auction invitations by distribution at runtime.

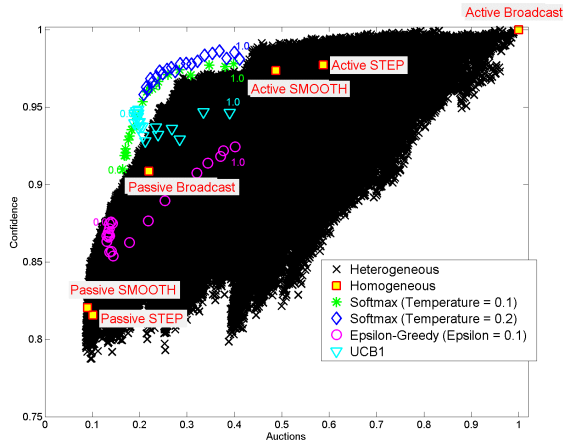


Figure 9. Performance for scenario 5 showing homogeneous and heterogeneous assignment of strategies as well as assignments done by bandit solvers. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each. The bandit solvers' reward functions normalised the number of auction invitations by distribution at runtime.

strategies were employed, auctions were initiated when the tracked object was within 20 pixels of the border of the FOV.

Figure 12 shows the results obtained from all homogeneous and heterogeneous as well as those obtained by decentralised online learning. As with the results in section IV, heterogeneous configurations lead to system wide outcomes which are more Pareto efficient than those possible in the homogeneous case. Furthermore, as with the results in section V, the use of decentralised online learning of marketing strategies also extended the Pareto efficient frontier when compared to the homogeneous case. However, in this instance learning was not able to generate outcomes dominating the most Pareto efficient heterogeneous cases. We speculate that this is due to the presence of already

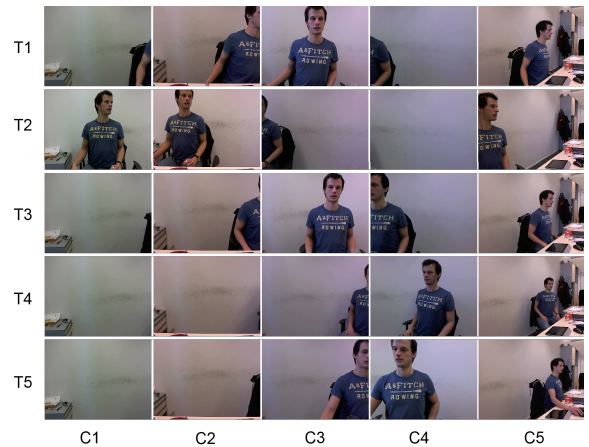


Figure 11. Shots from five participating cameras tracking a single person.

highly Pareto efficient outcomes from static heterogeneous configurations, which would be difficult to find while still exploring the space sufficiently. It is likely that dynamic configurations in more complex real world scenarios will yield similar interesting results to those described in section V-A, though this remains an area for future research.

E. Quantifying the Efficiency Increase

One way of comparing Pareto efficient frontiers is to compute the hypervolume [16] under each frontier, given a reference point. With two metrics, this amounts to computing the area under the frontier. The reference point can be specified as the vector of worst case values. Thus, a tracking confidence value of 0.0, and a number of auction invitations value of 1.0 specifies our reference point. The greater the hypervolume of a Pareto frontier, the more efficient it is.

If an outcome a is dominated by none of the outcomes

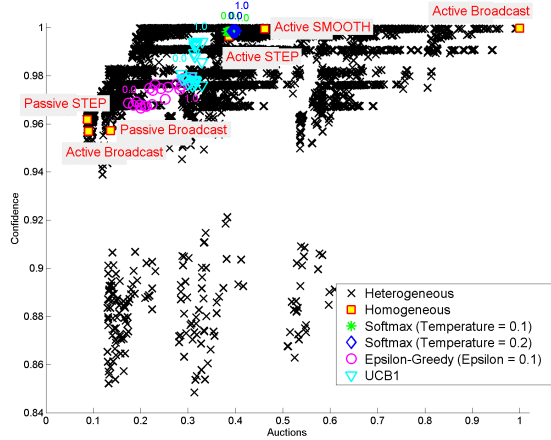


Figure 12. Performance of all configurations on scenario 12.

defining a frontier, it extends this frontier, resulting in a new frontier containing a , and having a larger hypervolume. We exemplified such an extension of the frontier in figure 2. The Pareto efficient frontier considering the outcomes from static homogeneous configurations gives us a frontier which we call h . The Pareto efficient frontier considering all the outcomes from both static homogeneous and static heterogeneous configurations gives us a frontier that we call h - he . Outcomes due to a bandit solver, together with static homogeneous configuration outcomes, give us frontiers h - eg , h - sm , or h - ucb , depending on the considered bandit solver being EPSILON-GREEDY, SOFT-MAX, or UCB1 respectively. Similarly, outcomes due to a bandit solver, together with static homogeneous configuration outcomes and static heterogeneous configuration outcomes, give us frontiers h - he - eg , h - he - sm , or h - he - ucb . Table II shows the medians (across 30 independent runs) of the hypervolumes of the aforementioned frontiers, indicating statistical significance in the extensions with respect to h and h - he , across the scenarios considered in this paper.

It is evident that heterogeneity of marketing strategies, which results in outcomes contained in the frontiers h - he , extend the frontiers h , regardless of the scenario considered. The frontiers h - eg , h - sm , and h - ucb , which contain outcomes from bandit solvers, often extend the frontier arising from static homogeneous configuration outcomes h . Moreover, the frontiers h - he - eg , h - he - sm , and h - he - ucb , apart from extending the frontier h , sometimes further extend the frontier that includes both static homogeneous and static heterogeneous configuration outcomes h - he . Thus, decentralised online learning, based on bandit solvers, can lead to the network self-organising towards global outcomes that are more Pareto efficient than those from static configurations. Extensions of the Pareto efficient frontier tell us that dynamic configurations induced by online learning allow the network to reach favourable parts of the objective space, which are inaccessible in the static case.

VI. CONCLUSIONS AND FUTURE WORK

We have studied the self-organising behaviour of smart camera networks which use auctions to exchange object tracking responsibilities during runtime. We showed that heterogeneous configurations of marketing strategies in the network can lead to increased network-level tracking confidence while simultaneously decreasing the number of auction invitations, a proxy for communication and processing overhead. I.e. heterogeneity led to more Pareto efficient outcomes than those possible in homogeneous configurations. We demonstrated this on a range of scenarios, both using an open source simulation package and real video feeds.

Since scenarios are not known in advance, and the performance of configurations varies significantly between scenarios, configurations which lead to Pareto efficiency in a given deployment will also not be known in advance. Our second contribution was to show that multi-armed bandit solving algorithms can be used to achieve online learning of marketing strategies at the individual camera level, to find high performing heterogeneous configurations. These outcomes extended the Pareto efficient frontier when compared to the homogeneous case, from the system point of view. Furthermore, in many cases, the dynamic behaviour resulting from online learning led to outcomes which extended the Pareto frontier even when compared to the best possible outcomes from static heterogeneous configurations.

Although this paper is based on camera networks, the principles behind both heterogeneous configuration and decentralised online learning are not limited to camera networks. Indeed, the benefits observed due to increased configuration possibilities in the heterogeneous case should be applicable to other networked systems. Similarly, the technique used for decentralised online learning of configurations should also be more widely applicable. Evaluation of these principles in other systems is an area for future work.

Indeed, there are many avenues for extending this line of research. Firstly, further evaluation of the approaches introduced here using more complex deployments may provide insight into when different bandit solvers outperform each other, and when they outperform static configurations. Secondly, in this paper we have focussed our study on classic, well known bandit solvers. There are many other bandit solver algorithms to be found in the literature, some of which are designed explicitly with unbounded rewards and dynamic reward distributions in mind. These, along with other online learning techniques, may provide further improvements. Finally, how to derive multi-attribute reward functions, for use at the local level, such that they give rise to a good spread of Pareto efficient outcomes in the global objective space, remains an open question.

ACKNOWLEDGMENT

The research leading to these results was conducted in the EPiCS project and received funding from the European

| Pareto front Scenario ID | Static fronts | | Fronts with learnt outcomes | | | | | |
|-----------------------------|---------------|----------|-----------------------------|-----------|-----------|-----------|-----------|------------|
| | h | $h-he$ | $h-eg$ | $h-sm$ | $h-ucb$ | $h-he-eg$ | $h-he-sm$ | $h-he-ucb$ |
| 1 | 0.6059 | 0.6271 * | 0.6337 * | 0.6612 *† | 0.6516 *† | 0.6443 *† | 0.6612 *† | 0.6532 *† |
| 2 | 0.7926 | 0.8255 * | 0.8107 * | 0.8080 * | 0.8061 * | 0.8320 *† | 0.8338 *† | 0.8313 *† |
| 3 | 0.7129 | 0.7608 * | 0.7457 * | 0.7607 * | 0.7558 * | 0.7808 *† | 0.7841 *† | 0.7830 *† |
| 4 | 0.7658 | 0.7824 * | 0.7834 * | 0.7864 * | 0.7839 * | 0.8012 *† | 0.7993 *† | 0.7993 *† |
| 5 | 0.8552 | — | 0.8702 * | 0.8895 * | 0.8778 * | — | — | — |
| 6 | 0.8957 | — | 0.9095 * | 0.9097 * | 0.9094 * | — | — | — |
| 7 | 0.8191 | — | 0.8420 * | 0.8450 * | 0.8446 * | — | — | — |
| 8 | 0.8258 | — | 0.8455 * | 0.8608 * | 0.8503 * | — | — | — |
| 9 | 0.8487 | 0.8800 * | 0.8581 * | 0.8762 * | 0.8706 * | 0.8810 * | 0.8840 *† | 0.8823 * |
| 10 | 0.9219 | — | 0.9229 | 0.9243 | 0.9244 | — | — | — |
| 11 | 0.8384 | — | 0.8490 | 0.8954 * | 0.8832 * | — | — | — |
| 12 | 0.8998 | 0.9153 * | 0.9053 * | 0.9002 | 0.9039 * | 0.9153 * | 0.9153 * | 0.9153 * |

Table II

MEDIANS (OVER 30 INDEPENDENT RUNS) OF THE HYPERVOLUME OF THE PARETO FRONT RESULTING FROM VARIOUS CONFIGURATIONS: STATIC HOMOGENEOUS (h), BOTH STATIC HOMOGENEOUS AND STATIC HETEROGENEOUS ($h-he$), BOTH STATIC HOMOGENEOUS AND LEARNT ($h-eg$, $h-sm$, $h-ucb$), AND STATIC HOMOGENEOUS, STATIC HETEROGENEOUS AND LEARNT ($h-he-eg$, $h-he-sm$, $h-he-ucb$). THE WILCOXON RANK SUM TEST WAS USED WITH A 95% CONFIDENCE LEVEL TO ASSESS STATISTICAL SIGNIFICANCE. “*” DENOTES A SIGNIFICANT INCREASE W.R.T. STATIC FRONT h . “†” DENOTES A SIGNIFICANT INCREASE W.R.T. STATIC FRONT $h-he$. “—” DENOTES TESTS THAT WERE NOT PERFORMED DUE TO COMPUTATIONAL INFEASIBILITY OF EVALUATING ALL HETEROGENEOUS CONFIGURATIONS.

Union Seventh Framework Programme under grant agreement n° 257906. <http://www.epics-project.eu/>

REFERENCES

- [1] A. Campbell, C. Riggs, and A. Wu, “On the impact of variation on self-organizing systems,” in *Proceedings of the Fifth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2011)*. Washington, DC, USA: IEEE Computer Society Press, 2011, pp. 119–128.
- [2] G. Anders, C. Hinrichs, F. Siefert, P. Behrmann, W. Reif, and M. Sonnenschein, “On the influence of inter-agent variation on multi-agent algorithms solving a dynamic task allocation problem under uncertainty,” in *Proceedings of the Sixth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2012)*. Washington, DC, USA: IEEE Computer Society Press, 2012, pp. 29–38.
- [3] N. Salazar, J. A. Rodriguez-Aguilar, and J. L. Arcos, “Self-configuring sensors for uncharted environments,” in *Proceedings of the Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010)*. Washington, DC, USA: IEEE Computer Society Press, 2010, pp. 134–143.
- [4] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, “Socio-economic vision graph generation and handover in distributed smart camera networks,” *ACM Transactions on Sensor Networks*, vol. 10, no. 2, 2014, in press.
- [5] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, vol. 47, pp. 235–256, 2002.
- [6] A. Prasath, A. Venuturumilli, A. Ranganathan, and A. A. Minai, “Self-organization of sensor networks with heterogeneous connectivity,” in *Sensor Networks: Where Theory Meets Practice*, G. Ferrari, Ed. Springer, 2009, pp. 39–59.
- [7] D. Rojković, T. Crnic, and I. Cavrak, “Agent-based topology control for wireless sensor network applications,” in *MIPRO, 2012 Proceedings of the 35th International Convention*. Rijeka, Croatia: MIPRO, 2012, pp. 277–282.
- [8] E. F. Nakamura, H. S. Ramos, L. A. Villas, H. A. de Oliveira, A. L. de Aquino, and A. A. Loureiro, “A reactive role assignment for data routing in event-based wireless sensor networks,” *Computer Networks*, vol. 53, no. 12, pp. 1980–1996, 2009.
- [9] K. Römer, C. Frank, P. J. Marrón, and C. Becker, “Generic role assignment for wireless sensor networks,” in *Proceedings of the 11th ACM SIGOPS European Workshop*. New York, NY, USA: ACM Press, 2004.
- [10] B. Dieber, C. Micheloni, and B. Rinner, “Resource-aware coverage and task assignment in visual sensor networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1424–1437, 2011.
- [11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [12] L. Esterle, P. R. Lewis, B. Rinner, and X. Yao, “Improved adaptivity and robustness in decentralised multi-camera networks,” in *Proceedings of the Sixth ACM/IEEE International Conference on Distributed Smart Cameras*. IEEE Press, 2012, pp. 1–6.
- [13] M. Gagliolo and J. Schmidhuber, “Algorithm portfolio selection as a bandit problem with unbounded losses,” *Annals of Mathematics and Artificial Intelligence*, vol. 61, no. 2, pp. 49–86, 2011.
- [14] J. Vermorel and M. Mohri, “Multi-Armed Bandit Algorithms and Empirical Evaluation,” in *Machine Learning: ECML 2005*, J. Gama, R. Camacho, P. Brazdil, A. Jorge, and L. Torgo, Eds. Springer, 2005, pp. 437–448.
- [15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [16] L. While, P. Hingston, L. Barone, and S. Huband, “A faster algorithm for calculating hypervolume,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.