


Article

# Dynamic camera reconfiguration with reinforcement learning and stochastic methods for crowd surveillance

Niccoló Bisagno <sup>1,\*</sup>  0000-0001-5704-5785, Alberto Xamin <sup>1</sup>, Francesco De Natale <sup>1</sup>, Nicola Conci <sup>1</sup> and Bernhard Rinner <sup>2</sup>

<sup>1</sup> Department of Information Engineering and Computer Science (DISI) - University of Trento; name.surname@unitn.it

<sup>2</sup> Institute of Networked and Embedded Systems (NES) - University of Klagenfurt; bernhard.rinner@aau.at

\* Correspondence: niccolo.bisagno@unitn.it

† This paper is an extended version of our paper published in ICDSC 2018.

Version October 22, 2020 submitted to Sensors

**Abstract:** Crowd surveillance plays a key role to ensure safety and security in public areas. Surveillance systems traditionally rely on fixed camera networks, which suffer from limitations in coverage of the monitored area, video resolution and analytic performance. On the other hand, a smart camera network provides the ability to reconfigure the sensing infrastructure by incorporating active devices such as pan-tilt-zoom (PTZ) cameras and UAV-based cameras, thus enabling the network to adapt over time to changes in the scene. We propose a new decentralised approach for network reconfiguration, where each camera dynamically adapts its parameters and position to optimise scene coverage. Two policies for decentralised camera reconfiguration are presented: a greedy approach and a reinforcement learning approach. In both cases, cameras are able to locally control the state of their neighbourhood and dynamically adjust their position and PTZ parameters. When crowds are present, the network balances between global coverage of the entire scene and high resolution for the crowded areas. We evaluate our approach in a simulated environment monitored with fixed, PTZ and UAV-based cameras.

**Keywords:** Distributed Camera Network, Reinforcement Learning, Crowd Surveillance, UAV, PTZ, Simulation

## 1. Introduction

Camera networks for surveillance applications play a key role to ensure safety of public gatherings [1–4]. Security applications in crowded scenarios have to deal with a variety of factors which can lead to critical situations [5–7]. In such scenarios, a camera network must be able to record local events as well as to ensure a global coverage of the area of interest [8].

Ensuring both coverage of the whole monitoring area and a good video quality of moving individuals is challenging using non-reconfigurable (*fixed*) cameras [5,9]. A high number of fixed cameras would provide the required coverage of the scene, but at a high cost. Moreover, fixed cameras, especially the ones with a large field of view (FoV) or a fisheye lens, would also capture areas of the scene where pedestrians are not present, thus creating an excessive amount of irrelevant data.

Reconfigurable cameras can dynamically adapt their parameters, such as FoV, resolution and position. For example, pan-tilt-zoom (PTZ) cameras and cameras mounted on unmanned aerial vehicles (UAVs) can dynamically adapt their position and FOV. Such cameras allow to greatly reduce the number of cameras in the network while optimising coverage and target resolution given the current state of the crowded scene. The goal of such system is to ensure a good resolution for common

31 tasks such as face recognition in critical areas, while providing a sufficient video quality in other  
32 parts. UAVs have been particularly studied as a flexible and effective system for crowd gatherings  
33 surveillance in recent years [10,11].

34 Reinforcement learning approaches have great potential for distributed camera networks  
35 optimization [2,12–15]. However, they have not been applied to the dynamic coverage of crowded  
36 scenes.

37 In [9], we proposed a greedy approach to control the trade-off between covering the widest  
38 possible area of the scenario of interest (*global coverage*) and focusing on the most crowded parts of the  
39 scene (*people coverage*). In our previous work, we proposed a decentralised greedy empirical approach,  
40 where each camera aims at optimising the coverage performance in its local neighbourhood. In this  
41 paper, we introduce a novel decentralised approach based on Reinforcement Learning (RL) which  
42 allows every camera to *learn* how to optimise the coverage performances. Both approaches rely on the  
43 estimation of the state of the crowd by merging the observations from individual cameras at a global  
44 level while each camera locally decides on its next state. Both RL and greedy approaches allow the  
45 cooperative use of fixed, PTZ and UAV-mounted cameras which can track and survey a crowd relying  
46 only on cooperation and map sharing, without using classical tracking by detection algorithms.

47 Our approach aims at guaranteeing the best possible coverage of the scene, exploiting the trade-off  
48 between global coverage and people coverage. For this goal, we employ different cameras, namely,  
49 fixed cameras, PTZ, and UAV-based cameras, which have different features and capabilities. Using  
50 multiple heterogeneous cameras enriches our coverage of an area of interest by providing different  
51 point of views and possible camera configurations, thus increasing the reliability of the collected data.  
52 Being able to reconfigure camera parameters, such as position and field of view, allows our network  
53 to seamlessly work in both static and dynamic scenarios in which people move continuously in the  
54 environment.

55 Our contribution can be summarised as (1) a policy to trade-off between global coverage and  
56 people coverage, which can be fine tuned for different cameras types, (2) a new metric to evaluate  
57 the performances of the surveillance task, (3) a greedy framework to track the crowd flow based  
58 on cooperative approach, (4) a distributed machine learning framework based on reinforcement  
59 learning (RL) for covering crowded areas, and (5) a 3D simulator of crowd behaviors based on [16]  
60 and heterogeneous camera networks.<sup>1</sup>

61 The remainder of this paper is organized as follows: Section 2 discusses related papers. Section 2  
62 describes our greedy approach for camera reconfiguration. Section 3.7 introduces the evaluation metrics  
63 and Section 3.8 discusses our RL-based approach. Section 4 presents the results of our simulation  
64 study, and Section 5 provides some concluding remarks together with a discussion about potential  
65 future work.

## 66 2. Related Work

67 Cooperative video surveillance system research has been developed to drastically reduce human  
68 supervision [17–19]. This framework usually allows cooperative cameras to share real-time information  
69 between them in order to capture events and to guarantee global coverage of the area of interest [1–3].

70 When observing a crowded scenario, the state of the scene evolves dynamically and the camera  
71 network should be able to reconfigure and cover events as they happen. Due to their nature, events  
72 generated by moving pedestrians are unique and often can not be reproduced, thus making it difficult  
73 to test and evaluate different camera networks configuration and policies.

74 Leveraging on simulators and virtual environments can be an effective tool to deal with these  
75 limitations. Virtualisation paradigms have been exploited both in camera surveillance [5,6] and crowd  
76 analysis [9,20].

---

<sup>1</sup> Simulator available at <https://github.com/nick1392/HeterogenousCameraNetwork>

77 In camera surveillance, fixed cameras can be used together with reconfigurable cameras such  
 78 as UAV-based and PTZ cameras [5,6,21]. PTZs can dynamically set their parameter to optimise the  
 79 coverage of areas of interest, progressively scanning a wide area or zooming in on events of interest.  
 80 These cameras have been particularly employed to cooperatively track pedestrians, for example  
 81 [21–24].

82 UAVs have been employed for civil and military tasks, such as environmental pollution  
 83 monitoring, agriculture monitoring, and management of natural disaster rescue operations [25–27].  
 84 Military applications also involve surveillance, but their use in common crowd surveillance scenarios  
 85 is limited because of regulations.

86 In [28], the key features of a distributed network for crowd surveillance are (1) locating and  
 87 re-identifying a pedestrian across multiple cameras, (2) tracking people, (3) recognising and detecting  
 88 local and global crowd behavior, (4) clustering and recognising actions, and (5) detecting abnormal  
 89 behaviors. To achieve these features, the following issues need to be tackled: how to fuse information  
 90 coming from multiple cameras, performing crowd behavior analysis tasks, how to learn crowd behavior  
 91 patterns, and how to cover an area with particular focus on key events.

92 Reinforcement learning approaches [29] have been applied to distributed systems in the context  
 93 of surveillance for different purposes. Hatanaka et al. [12] investigate the optimal theoretical coverage  
 94 that a network of PTZ cameras can achieve in an unknown environment. In [2,13], online tracking  
 95 applications using reinforcement learning are shown to outperform static heterogeneous cameras  
 96 configuration. Khan et al. [14] employ reinforcement learning for resource management and power  
 97 consumption optimisation in distributed cameras system. In [15] dynamic alignment of PTZ cameras  
 98 is exploited to learn coverage optimisation. Although RL has demonstrated its effectiveness in camera  
 99 networks, dynamic coverage of crowded scenes using UAVs has not been tackled yet.

100 Recently, Altahir et al. [7] solve the camera placement problem with predefined risk maps which  
 101 have an higher priority to be covered. In [30], a distributed PSO (Particle Swarm Optimisation) is  
 102 employed to maximise the geometric coverage of the scene. Vejdandarast et al. [31] focus on the best  
 103 zoom level selection for redundant coverage of risky areas using a distributed camera network.

### 104 3. Method

105 In this section, we introduce key part of our method. First, the observation model for the  
 106 environment establishes a relation between the observation and its confidence. Next, camera types  
 107 and features are described in detail. Finally we describe how the greedy reconfiguration policy and the  
 108 RL-based approach exploit the network-wide trade-off between global coverage and crowd resolution.

#### 109 3.1. Observation Model

110 The region of interest  $C$ , which has to be surveyed is divided into a uniform grid of  $I \times J$  square  
 111 cells, where the indexes  $i \in \{1, 2, \dots, I - 1\}$  and  $j \in \{1, 2, \dots, J - 1\}$  of each cell  $c_{i,j} \in C$  represent the  
 112 position of the cell in the grid. We assume a scenario evolving at discrete time steps  $t = 0, 1, 2, \dots, t_{end}$ .  
 113 At each time step, the network is able to gather the observation over the scene to be monitored, process  
 114 it, and share it with the other camera nodes. Given the observation, each camera is able to compute its  
 115 next position. For this purpose, we define

- 116 • an observations vector  $O_{i,j}$ , which represents the number of pedestrians detected for each cell  
 117  $c_{i,j} \in C$ ;
- 118 • a spatial confidence vector  $S_{i,j}$ , which describes the confidence of the measures for each cell  
 119  $c_{i,j} \in C$ . Our spatial confidence depends only on the relative geometric position of the observing  
 120 camera and the observed cell;
- 121 • a temporal confidence vector  $L_{i,j}^t$ , which depends on the time passed since the cell has last been  
 122 observed; and
- 123 • an overall confidence vector  $F_{i,j}^t$ , which depends on the temporal and spatial confidences.

The observations vector is defined as

$$O_{i,j} = \{o_{1,1}, o_{1,2}, \dots, o_{i,j}, \dots, o_{I,J}\}. \quad (1)$$

The value  $o_{i,j}$  for each cell  $c_{i,j}$  is given as

$$o_{i,j} = \begin{cases} \frac{ped}{PED_{max}} & \text{if } ped \leq PED_{max} \\ 1 & \text{if } ped > PED_{max} \end{cases} \quad (2)$$

124 where  $ped$  is the current number of pedestrians in a cell and  $PED_{max}$  is the threshold for the number  
125 of pedestrians a cell is considered as crowded.  $PED_{max}$  can be manually tuned depending on the  
126 application. Crowded cells should be monitored with a higher resolution.

127 Occlusion of targets is one of the main challenges in crowded scenarios. We assume that our  
128 camera network is able to robustly detect a pedestrian when its head is captured with a resolution of at  
129 least  $24 \times 24$  pixels, which is in line with the smaller bound for common face detection algorithms [32].

For each cell, a spatial confidence vector is defined as

$$S_{i,j} = \{s_{1,1}, s_{1,2}, \dots, s_{i,j}, \dots, s_{I,J}\} \quad (3)$$

130 where the value  $0 < s_{i,j} \leq 1$  is bounded and decreases as the distance between the observing camera  
131 and the cell of interest  $c_{i,j}$  increases. The actual value of the spatial confidence  $s_{i,j}$  in a given cell  
132 depends on the type of observing camera and is described in Section 3.2.

Similarly, a temporal confidence vector is defined as

$$L_{i,j} = \{l_{1,1}^t, l_{1,2}^t, \dots, l_{i,j}^t, \dots, l_{I,J}^t\}. \quad (4)$$

Each value  $l_{i,j}^t$  is defined as

$$l_{i,j}^t = \begin{cases} 1 - \frac{t - t_{i,j}^0}{T_{MAX}} & \text{if } t - t_{i,j}^0 \leq T_{MAX} \\ 0 & \text{if } t - t_{i,j}^0 > T_{MAX} \end{cases} \quad (5)$$

133 where  $t_{i,j}^0$  is the most recent time instant, in which cell  $c_{i,j}$  was observed, and  $T_{MAX}$  represents the time  
134 instant, after which the confidence drops to zero. The value  $l_{i,j}^t$  decays over time if no new observation  
135  $o_{i,j}$  on cell  $c_{i,j}$  becomes available.

Given the spatial and temporal confidence metrics, the overall confidence vector is defined as

$$F^t = \{f_{1,1}^t, f_{1,2}^t, \dots, f_{i,j}^t, \dots, f_{I,J}^t\} \quad (6)$$

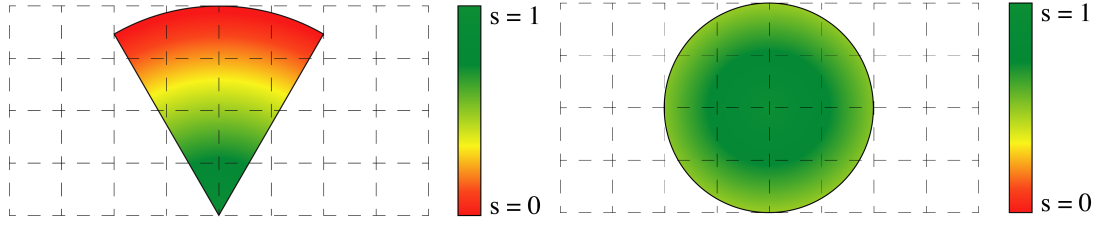
with

$$f_{i,j}^t = s_{i,j} * l_{i,j}^t. \quad (7)$$

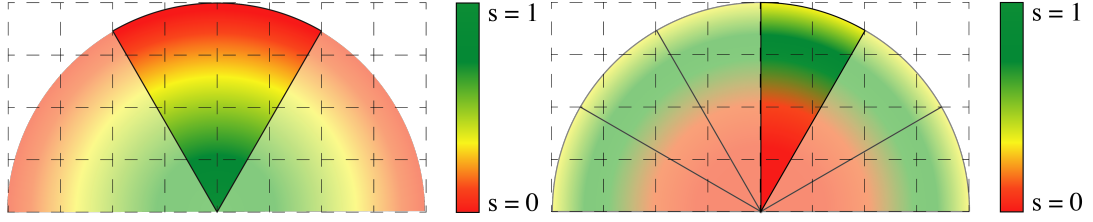
136 Thus, for each cell  $c_{i,j}$  we have an observation  $o_{i,j}$  with an overall confidence  $f_{i,j}^t$ . The confidence  
137 value varies between 0 and 1, where 1 represents the highest possible confidence. If a sufficient number  
138 of cameras is available for covering all cells concurrently, the overall confidence vector is given as  
139  $F^I = \{1, \dots, 1\}$ .

### 140 3.2. Camera Models

141 We briefly describe the models adopted for the three different camera types: fixed cameras, PTZ  
142 cameras, and UAV-based cameras. We assume that all fixed and PTZ cameras are mounted at a fixed  
143 height. For the same reason, UAV-based cameras fly at a fixed altitude, which also helps in reducing  
144 the computational complexity of the problem.



(a) A fixed camera observes the environment without varying the spatial confidence for each cell at each time step. (b) Example of the distribution of the spatial confidence in varying the spatial confidence for each cell at each time step the area surveyed by an UAV.



(c) At each time step, PTZ camera can pan between different positions. (d) PTZ cameras can also zoom in to an area, which causes their field of view to shrink, but improves the spatial confidence in areas further away from the camera.

### 145 3.2.1. Fixed Cameras

Fixed cameras (see Fig. 1(a)) provide a confidence matrix, which gradually decreases as the distance from the camera increases. Being  $(x, y)$  a point in the space at a distance  $d$  from a fixed camera, the value of the spatial confidence  $s(x, y)$  is defined as

$$s(x, y) = \begin{cases} -\frac{1}{d_{max}} * d + 1 & \text{if } d < d_{max} \\ 0 & \text{if } d \geq d_{max} \end{cases} \quad (8)$$

where  $d_{max}$  is the distance from the camera, over which the spatial confidence is zero. Thus, the confidence value  $s_{i,j}$  of cell  $c_{i,j}$  is defined as

$$s_{i,j} = \max\{s(x, y)\}_{\forall (x,y) \in c_{i,j}}. \quad (9)$$

### 146 3.2.2. PTZ Cameras

147 PTZ cameras are modeled similarly to fixed cameras, with the additional capability to dynamically  
148 change the field of view (see Fig. 1(c)).

149 PTZ cameras are able to pan-tilt and zoom between 9 different configurations and cover an area  
150 of  $180^\circ$  as shown in Figs. 1(c) and 1(d).

151 Fig. 1(c) shows how a PTZ camera can achieve different configurations using only the pan  
152 movement along the horizontal axis. Each confidence map is defined as the one of a fixed camera. In  
153 Fig. 1(d) the camera is able to zoom on an area further away from the camera, which causes 3 effects:  
154 the FOV decreases, the confidence in the zoomed area increases, and the confidence in other areas  
155 decreases. Let  $(x, y)$  represent a point in the scene at distance  $d$  from a fixed camera, then the value of  
156 the spatial confidence for a PTZ camera while zooming  $s(x, y)$  is defined as

$$s(x, y) = \begin{cases} 0 & \text{if } d < d_0 \\ -\frac{1}{d_{max}-d_0} * d + \frac{d_{max}}{d_{max}-d_0} & \text{if } d_0 \leq d < d_{max} \\ 0 & \text{if } d \geq d_{max} \end{cases} \quad (10)$$

157 where  $d_{max}$  is the distance from the camera over which we have 0 spatial confidence and  $d_0$  the closest  
158 distance captured in the FOV.

### 159 3.2.3. UAV-based Cameras

For UAV-based cameras, the FOV projection on the ground plane is different with respect to the previous models, as shown in Fig. 1(b). The spatial confidence of point  $(x, y)$  at a distance  $d$  from the UAV is computed as

$$s(x, y) = \begin{cases} -\frac{1}{d_{uav}} * d + 1 & \text{if } d < d_{uav} \\ 0 & \text{if } d \geq d_{uav} \end{cases} \quad (11)$$

160 where  $d_{uav}$  is the distance after which the confidence on the observation drops below a threshold  $g$   
161 over which we consider the observation reliable.

### 162 3.3. Reconfiguration Objective

163 The objective of the heterogeneous camera network is to guarantee the coverage of the scene  
164 while focusing on more densely populated areas. The priority metric defines the importance of each  
165 cell to be observed. A high value indicates that the cell is crowded or that we have a low confidence on  
166 its current state, thus requiring an action.

In order to formalise the reconfiguration objective, a priority vector  $P$  is defined as

$$P^t = \{p_{1,1}^t, p_{1,2}^t, \dots, p_{i,j}^t, \dots, p_{l,j}^t\}. \quad (12)$$

The priority for each cell is defined as

$$p_{i,j}^t = \alpha * o_{i,j}^t + (1 - \alpha) f_{i,j}^l \quad (13)$$

167 where  $0 \leq \alpha \leq 1$  represents a weighting factor to tune the configuration and  $f_{i,j}^l$  represents the  
168 predefined ideal confidence for the cell.

The objective  $G$  of each camera, given its possible set of actions, is to minimise the distance between the confidence vector and the priority vector

$$G = \min\{\|F^{t+1} - P^t\|\} \quad (14)$$

169 where  $\alpha$  can vary between 0 and 1

$$\begin{cases} \min\{F^{t+1} - F^l\} & \text{if } \alpha = 0 \\ \min\{F^{t+1} - O^t\} & \text{if } \alpha = 1. \end{cases} \quad (15)$$

170 Setting  $\alpha = 1$  causes the network to focus on observing more densely populated areas with no  
171 incentive to explore unknown cells. In contrast,  $\alpha = 0$  causes the network to focus on global coverage  
172 only without distinguishing on the crowd density of the cells.

### 173 3.4. Reconfiguration objectives: custom policies

174 The policy presented in [9] and reported in Sec. 3.3 suffers from two main limitations:

- 175 • The reconfiguration objectives are the same for the different camera types, namely UAVs and  
176 PTZs. In the real world, UAVs have a higher cost of deployment and movement with respect to  
177 PTZs, while they provide more degrees of freedom for their reconfigurability.
- 178 • The priority maps do not share information about cameras' type and position between different  
179 cameras. Especially in the case of UAVs, this can lead to a superposition of different cameras  
180 which decrease the network performances.

181 We propose two approaches to tackle these limitations.

The first approach, called *split priority*, is to use different priority vectors for different type of cameras, namely UAVs and PTZs. This allows to use different values of  $\alpha$  for UAVs and PTZs, thus allowing for different functionalities, such as ensuring a better coverage with UAVs while the PTZs can focus on target areas, or vice versa. The two priority vector  $P_{PTZ}^t$  and  $P_{UAV}^t$  are defined as:

$$P_{PTZ}^t = \alpha_{PTZ} \cdot O^t + (1 - \alpha_{PTZ})(1 - F^t)$$

and

$$P_{UAV}^t = \alpha_{UAV} \cdot O^t + (1 - \alpha_{UAV})(1 - F^t).$$

This second approach, called *position-aware UAVs*, aims at solving the superposition issue which comes from the different UAVs not being aware of each other's position. The vector  $P_{UAV}^t$  is modified as follows

$$P_{UAV}^t = \alpha_{UAV} \cdot O^t + (1 - \alpha_{UAV})(1 - F^t) + U^t$$

where  $U^t$  is a position vector containing a value  $u_{ij}$  for each cell, such that  $u_{ij}$  can take on two values:

$$u_{i,j} = \begin{cases} 0 & \text{if } \nexists \text{ UAV in } (i,j) \\ -1 & \text{if } \exists \text{ UAV in } (i,j). \end{cases}$$

By doing so, the cell priority is kept low whenever there is a UAV, thus penalizing the locations where other UAVs are present. In order not to penalize its current position, each UAV  $UAV_k$  updates its priority vector  $p_{UAV_k-i,j}^t$  by recovering its contribution to  $U^t$  by adding 1 to its current position:

$$p_{UAV_k-i,j}^t = \begin{cases} p_{UAV-i,j}^t + 1 & \text{if } \exists \text{ UAV}_k \text{ in } (i,j) \\ p_{UAV-i,j}^t & \text{otherwise.} \end{cases}$$

182 The last operation is that every UAV normalises its priority in the range [0 1] from the range [-1 1] so  
183 that it is compatible with the cost function in Eq. 14 to be minimised.

### 184 3.5. Update Function

185 At each time step  $t$ , the network has knowledge about the current observation vector  $O^t$ , the  
186 spatial confidence vector  $S^t$ , the temporal confidence vector  $L^t$ , and the overall confidence vector  $F^t$ .  
187 In order to progress to the next time step  $t + 1$ , an update function for these vectors is required.

188 The temporary spatial confidence vector  $S_{temp}^{t+1}$  is determined by the geometry of cameras at time  
189  $t + 1$ . For each cell, the value  $s_{temp,i,j}^{t+1}$  is the maximum spatial confidence value of all cameras observing  
190 the cell  $(i, j)$ . Cells that are not covered by any camera have a spatial confidence value of 0.

We estimate the temporal confidence vector as follows:  $L_{time}^{t+1}$  is computed by applying Eq. 5 to each element of  $L^t$ . Another temporary temporal confidence vector  $L_{new}^{t+1}$  is computed by setting to 1 to all cells currently observed, and setting to 0 all other cells. With the estimated vectors, we compute two estimations of the overall confidence vector:

$$F_{time}^{t+1} = S^t * L_{time}^{t+1} \quad (16)$$

and

$$F_{new}^{t+1} = S_{temp}^{t+1} * L_{new}^{t+1}. \quad (17)$$

The new overall confidence vector is then computed as

$$F^{t+1} = \max\{F_{new}^{t+1}, F_{time}^{t+1}\}_{\forall(i,j)}. \quad (18)$$

191 For each cell  $(i, j)$  in which  $f_{new}^{t+1} > f_{time}^{t+1}$ , we also need to update the last time the cell has been  
 192 observed  $t^0(i, j) = t + 1$  and the observation vector  $o^t(i, j)$ .

### 193 3.6. Local Camera Decision: Greedy Approach

194 In our approach, all the information vectors described in Section 3.1 are shared and known to all  
 195 cameras. Each camera locally decides its next position using a greedy approach to minimise the cost  
 196 defined in Eq. 14 in its neighbourhood.

197 At each time step, each PTZ and UAV-based camera select a neighbourhood that can be explored.  
 198 The UAV's neighbourhood is defined as a square centered at the cell where the drone is currently  
 199 placed (see Fig. 1(b)). The PTZ neighbourhood is a rectangle which covers the space in front of the  
 200 camera as shown in Fig. 1(c).

201 For each cell in the neighbourhood, we center a window  $W$  of size  $N_w \times N_w$  on each cell  $c_W \in W$   
 202 and we store in the cell the value

$$c_W = \sum \|f_{i,j}^{t+1} - p_{i,j}^t\|. \quad (19)$$

203 The UAV will then move toward the cell in its neighbourhood with the largest  $c_W$ , and the PTZ  
 204 steers its FOV to be centered on that cell. If two or more cells have the same value of  $c_W$ , the camera  
 205 selects one of them randomly.

### 206 3.7. Evaluation Metrics

We define the Global Coverage Metric (GCM) for evaluating the network coverage capability as

$$GCM(t) = \frac{\sum_{\forall c_{i,j} | f_{i,j}^t > g} 1}{I * J} \quad (20)$$

with  $g$  being the threshold over which the cell is considered to be covered. We then average the results  
 for the whole duration of the observation as

$$GCM_{avg} = \frac{\sum_{t=0, \dots, t_{end}} GCM(t)}{t_{end} + 1}. \quad (21)$$

We define the People Coverage Metric (PCM) for evaluating the network capability to cover  
 pedestrian in the scene as

$$PCM_{tot} = \frac{\sum_{\forall person \in c_{i,j} | f_{i,j}^t > p} 1}{totalPeople} \quad (22)$$

207 with  $p$  being the threshold over which the person is considered to be covered.

### 208 3.8. Reinforcement learning

209 On the one hand, an approach based on reinforcement learning presents a few advantages with  
 210 respect to a greedy approach, such as better performance and the ability to have longer-term planning  
 211 since the decision of each agent does not depend only on the last observation but also from past  
 212 observations. On the other hand, reinforcement learning requires a training phase which is not needed  
 213 in case of an empirical greedy approach.

214 Our novel reinforcement learning approach is based on a set of UAV-based cameras. We focus  
 215 on UAV-based cameras because it is the most challenging camera type with the most degrees of  
 216 freedom. Using our predefined observation and priority models (Sec. 3.1 and Sec. 3.4), we control each  
 217 UAV-based camera using an RL agent, which replaces the greedy approach for local camera decision  
 218 from Sec. 3.6 in their local decision-making process.



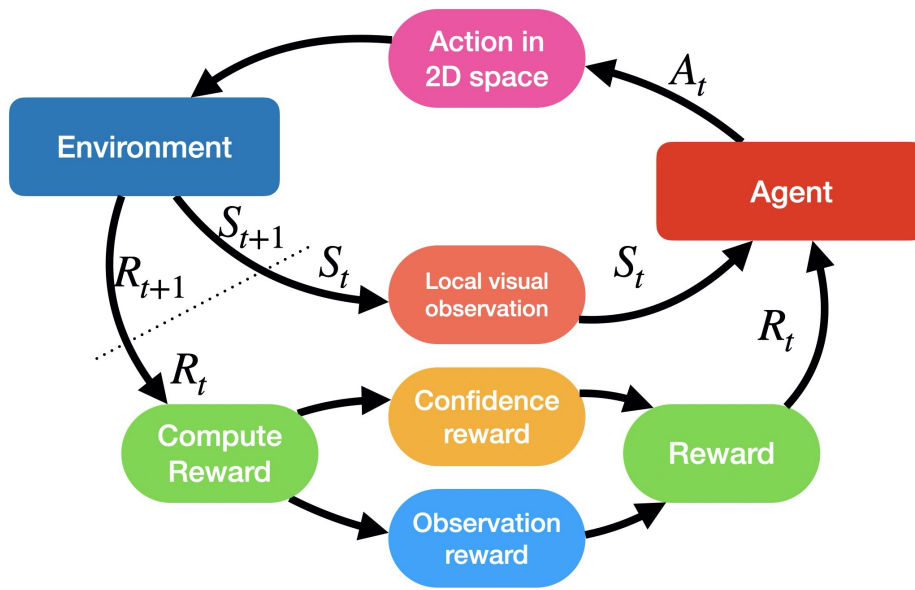


Figure 1. The workflow of our RL approach.

219 We rely on the vanilla ML Agents reinforcement learning network provided by [33] for our  
 220 deployment with UAVs. We use Soft Actor Critic (SAC) [34] as the backbone of our RL method. For  
 221 our approach (shown in Fig. 1), we define:

- 222 • a set of states  $\mathbf{S}$ , which encodes the local visual observation of each UAV,  
 223 • a set of possible actions  $\mathbf{A}$ , which each UAV can choose to perform at the next time step, and  
 224 • a set of rewards  $\mathbf{R}$ , which depends on the observation vector  $O^t$  and its related confidence  $F^t$ .

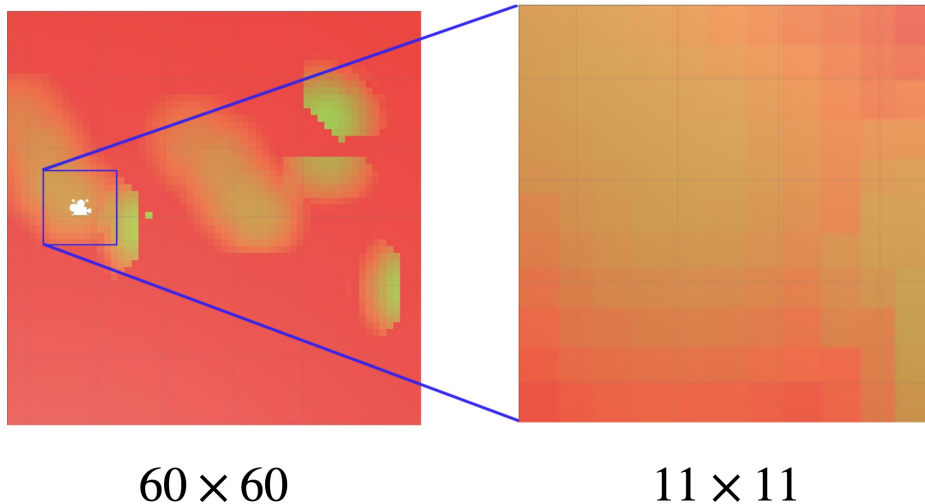


Figure 2. Visual observation of a drone is a  $11 \times 11$  portion of the plotted priority vector  $P^t$  in its neighbourhood.

At each timestep  $t$ , the agent is provided with a visual observation embedded in  $S_t \in \mathbf{S}$ , as shown in Fig. 2. The visual observation consists of a texture which contains a visualisation of the priority vector  $P^t$ , centered on the drone position with size  $11 \times 11$  cells. The visual observation is embedded in the state vector  $S_t$  of each agent's internal neural network. Each pixel and color channel of the

visual information is normalised to the range  $[0 - 0.1]$ . Based of the state  $S_t$ , the agent selects an action  $A_t \in \mathbf{A}$ .  $A_t$  is composed of all possible positions the drone can travel to in the observed window  $S_t$ . With the state-action pair  $(S_t, A_t)$  the time  $t$  is incremented to  $t + 1$ , the environment is transitioned to a new state  $S_{t+1} \in \mathbf{S}$  and a reward  $R_{t+1} \in \mathbf{R}$  is provided to the agent. Our reward is computed as

$$Reward = (\alpha - 1) \cdot \Delta GCM_t + \alpha PCM_t \quad (23)$$

where  $\alpha$  can be set at training time to obtain the same effect described in Sec. 3.3. The two metrics are defined as

$$\Delta GCM_t = GCM(t) - GCM(t - 1)$$

and

$$PCM_t = \frac{\sum_{\forall person \in c_{i,j} | f_{i,j}^t > p} 1}{totalCurrentPeople}$$

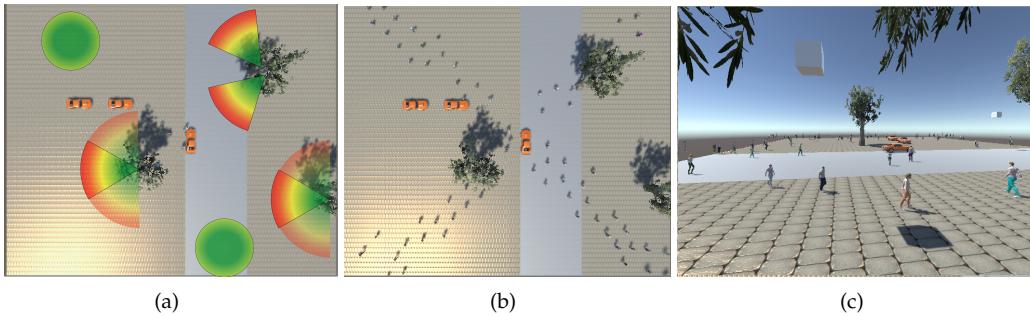
225 which is the instantaneous People Coverage metric.

226 For training, we set  $T_{max} = 1$  s and execute each episode for 50 timesteps such that the drone can  
 227 experience loss of coverage early and improve on it. An episode is completed if the whole map is  
 228 covered or if the timestep limit has been reached.

## 229 4. Experimental Results

230 For the experiments, we define an environment of size  $60 \times 60m^2$ . The scene is square-shaped,  
 231 exhibiting people passing by cars and vegetation. Pedestrians can enter and exit the scene from any  
 232 point around the square. Each cell  $c_{i,j}$  is a square of  $1 \times 1m^2$ . In this environment, 2 fixed cameras, 2  
 233 UAVs, and 2 PTZs are positioned as shown in Fig. 3(a). Sample images of the environment from a PTZ  
 234 and a UAV-based camera are shown in Figures 3(b) and 3(c), respectively. For our experiments, we  
 235 simulate the movement of 400 pedestrians crossing the scene with the following parameters :

- 236 •  $T_{max} = 3$  s
- 237 •  $PED_{max} = 2$
- 238 •  $d_{max} = 10$  m
- 239 • fixed and PTZ cameras height = 5 m
- 240 • UAV-based cameras height = 7 m



**Figure 3.** (a) Top view of the simulation environment including the camera positions. (b) Top view of the simulation environment including people. (c) Sample image from a PTZ camera.

### 241 4.1. Quantitative Results

242 In this section, we present the quantitative results obtained with our 4 different approaches (*greedy*,  
 243 *split priority*, *position aware* and *RL-based*) in the simulated environment. The goal is to evaluate the

ID	$g$ and $p$	$\alpha$	GCM	PCM
1	0.2	0	12.4 %	17.4 %
2	0.2	0.5	14.3 %	20.5 %
3	0.2	1	10.4 %	13.5 %
4	0.01	0	42.9 %	47.6 %
5	0.01	0.5	30.3 %	33.1 %
6	0.01	1	22.9 %	28.2 %
7	0.01	0	43.1 %	45.6 %
8	0.01	0.5	28.7 %	54.4 %
9	0.01	1	26.1 %	61.2 %

**Table 1.** Simulation experiments. Legend: ID–experiment;  $g, p$ –cell coverage thresholds; GCM–global coverage metric; PCM–people coverage metric. Experiments (1-6) refer to a uniformly distributed crowd, experiments (7-9) refer to a crowd with directional motion properties.

244 capabilities of the system to survey a crowded scene using the metrics defined in Sec. 3.7. We run 33  
245 different simulation experiments with varying values of  $g$ ,  $p$ , and  $\alpha$ .

246 The same simulation setup (starting cameras' positions and number of pedestrian in the scene) is  
247 used to evaluate the 4 different approaches: *greedy approach* (experiments(1-6), Tab. 1), *split priority*  
248 and *position aware* approaches (experiments(10-18), Tab. 2) and *reinforcement learning based* approach  
249 (experiments(19-24), Tab. 3). Experiments (7-9) display a single group of 10 pedestrians moving across  
250 the map and it is used to show the ability of our approach to track people in the scene.

251 The values  $g$  and  $p$  indicate the threshold above which we consider an observation reliable in  
252 time and space, respectively. A threshold of 0.2 indicates that our observation is at most 2.4 seconds  
253 old, when taken with a spatial confidence equal to 1. A threshold of 0.01 represents the cells and  
254 pedestrians about which we have a minimum level of information.

255 As baseline approach we assume that all 6 cameras are not able to change their configurations.  
256 Doing so, they are able to cover 6 % of the entire area with  $g = 0.2$  and 12 % with  $g = 0.01$ .

257 Table 1 summarises the results obtained using our *greedy* approach [9]. In experiments (3) and (6),  
258  $\alpha$  is set to 1, causing our camera network to focus only on observing pedestrians with no incentive to  
259 explore new areas in the environment. In experiments (1) and (4),  $\alpha$  is set to 0 resulting in maximizing  
260 the coverage regardless of the position of pedestrians. In experiments (2) and (5),  $\alpha$  is set to 0.5 aiming  
261 for balancing coverage and pedestrian tracking in crowded areas. We can observe that in experiments  
262 (1) and (4) we obtain the lowest values of GCM, which is expected since we are focusing on pedestrians.  
263 We also achieve the lowest scores in terms of PCM because cameras have no incentive in exploring  
264 new areas.

265 Experiments (7-9) are conducted using a directional crowd (Fig. 3(b)). When the network focuses  
266 only on observation in (9), it obtains the best results in terms of PCM and the worst one in terms of  
267 global coverage GCM. As expected, we obtain the best results in terms of coverage of the environment  
268 (GCM) in experiments (3) and (6). Since the crowd is uniformly distributed in the space, we also obtain  
269 the best results in terms of PCM. In experiments (2) and (5), the network combines global coverage  
270 and crowd monitoring, the system under performs compared with the scenes where  $\alpha = 0$  and  $\alpha = 1$ .

271 Table 2 summarises the results obtained using our *split priority* approach. Splitting the priority  
272 for different type of camera shows how UAVs have a key role when they are allowed to focus on  
273 the global observation of the scene (experiments (10-12)). Otherwise the performances of the whole  
274 network decreases (experiments (13-18)).

275 Both the *greedy* and *split priority* methods experience a decrease in performances when they have  
276 to focus on observing the more densely populated areas. When  $\alpha_{UAV} = 1$ , the UAVs tend to overlap  
277 and cover the same zone with a loss in the overall performance as shown in experiments (13-18).

278 To fix this issue, we developed the *position-aware* method, which results are reported in Table 2.  
279 With this methodology, which includes the knowledge of the UAVs position, the performance

ID	$g$ and $p$	$ff_{PTZ}$	$ff_{UAV}$	Split priority		Position aware	
				GCM	PCM	GCM	PCM
10	0.2	0	0	15.6%	18.8%	15.5%	20.3%
11	0.2	0.5	0	16.7%	18.8%	16.7%	19.1%
12	0.2	1	0	16.8%	18.5%	16.6%	20.6%
13	0.2	0	0.5	11.3%	14.4%	15.5%	20.7%
14	0.2	0.5	0.5	11.5%	14.3%	16.7%	21.8%
15	0.2	1	0.5	11.5%	12.0%	16.5%	21.2%
16	0.2	0	1	11.3%	11.6%	15.5%	20.4%
17	0.2	0.5	1	11.5%	14.0%	16.3%	19.1%
18	0.2	1	1	11.5%	11.2%	16.1%	20.4%

**Table 2.** Results of the simulations with method *split priority* and *position aware*

ID	$g$ and $p$	$\alpha$	GCM	PCM
19	0.2	0	$14.2 \pm 0.1\%$	$12.2 \pm 0.2\%$
20	0.2	0.5	$14.7 \pm 0.3\%$	$13.6 \pm 0.5\%$
21	0.2	1	$11.7 \pm 0.5\%$	$13.0 \pm 0.9\%$
22	0.01	0	$26.1 \pm 2.4\%$	$25.23 \pm 4.2\%$
23	0.01	0.5	$26.5 \pm 1.1\%$	$24.0 \pm 2.0\%$
24	0.01	1	$24.4 \pm 0.9\%$	$20.8 \pm 1.1\%$

**Table 3.** Simulation experiments with RL UAV control. Mean and standard deviation are computed from the results of 3 runs of each simulation. SAC is an algorithm that produces a stochastic policy, a single run would not be enough to evaluate the policy.

280 improves. The influence on the GCM with  $\alpha_{UAV} = 0$  is almost negligible, while for greater values the  
 281 improvement is clearly visible in both metrics (experiments (10-18)).

282 With this methodology the problem of overlapping UAVs is solved and this lead to performance  
 283 improvements since the UAVs collect information in different regions.

284 In Table 3, we report the results obtained using our *RL-based* approach. Our approach (experiments  
 285 (19-21)) is able to outperform the greedy approach (experiments (1-3)) when parameters  $g$  and  $p$   
 286 are set to 0.2. This method is thus more effective in long-term scenarios, when the temporal decay of the  
 287 observations is slower and allows for longer-term planning. On the other hand, when  $g$  and  $p$   
 288 are set to 0.01, the greedy approach (experiments (4-6)) is more effective in aggressively moving camera  
 289 configuration to the best place in a short amount of time and with a lower confidence threshold with  
 290 respect to the the RL-based approach (experiments (22-24)).

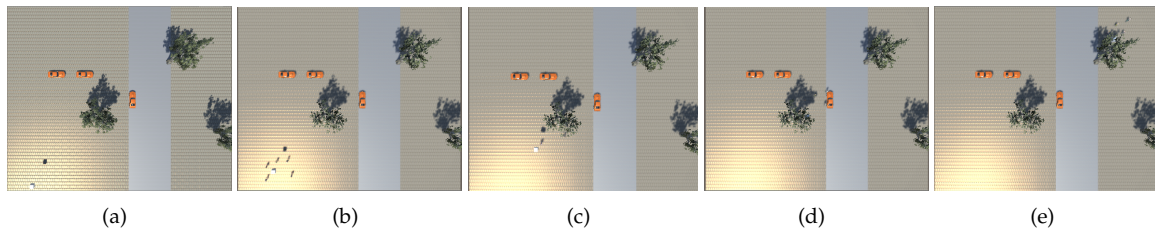
#### 291 4.2. Qualitative Results

292 In this section, we present the qualitative results obtained with our model in the simulated  
 293 environment. The goal is to demonstrate how our system is able to follow the crowd relying only on  
 294 detection of pedestrians in still frames rather than on classical tracking algorithms.

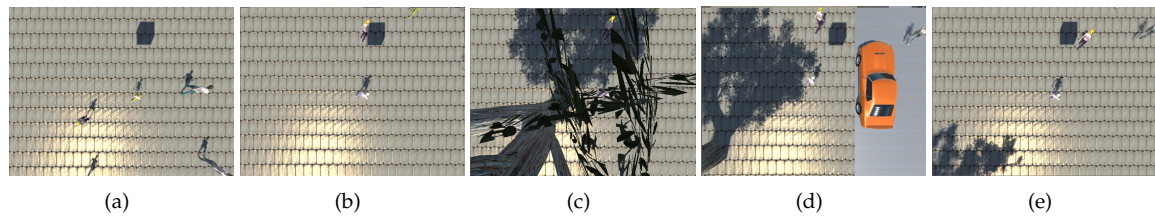
295 For this purposes, we simulate a single group of five pedestrians crossing the scene from the  
 296 bottom left to the top right as shown in the sequence depicted in Fig. 4. The UAV is able to closely  
 297 follow the pedestrians in the environment, scoring a  $PCM = 70.4\%$  and  $GCM = 3.2\%$ , as shown in  
 298 Fig. 5. Fig. 6 shows how observation priority and confidences maps are updated over time in order to  
 299 guide the UAV in the tracking scenario.

### 300 5. Conclusion

301 In this paper, we have presented two camera reconfiguration approaches for crowd monitoring,  
 302 a greedy camera approach and a RL-based one for UAV-mounted cameras. Our methods allow  
 303 heterogeneous camera networks to focus on high target resolution or wide coverage. Although based  
 304 on simplified assumptions for camera modeling and control, our approach is able to trade off coverage



**Figure 4.** Image sequence of a group of pedestrian moving from the bottom left of the environment (a) to the top right (c). The image is captured by a top view camera during the simulation to demonstrate the tracking behavior of our network.



**Figure 5.** Image sequence of a group of pedestrian moving from the bottom left of the environment (a) to the top right (e) captured by a UAV surveying the scene.

Scenario	Priority $P^t$	Observation $O^t$	temporal confidence $L^t$	Spatial confidence $S^t$	Overall confidence $F^t$
(1)					
(2)					
(3)					

**Figure 6.** Graphical representation of priority  $P^t$ , observation  $O^t$ , temporal confidence  $L^t$ , spatial confidence  $S^t$  and overall confidence  $F^t$  for 3 different scenarios: (1) Camera Network Sample, (2) Tracking sample at time  $t = 0$ , (3) Tracking sample at time  $t = 10$ . In (2) and (3) the UAV focuses on the observation matrix, such that the next priority map depends only on previous observations. Red represent the value 0, and green represents value 1.

305 and resolution of the network in a resource-efficient way. We have demonstrated how different cameras  
306 can be used in different manners to optimise the effectiveness of our method. In future work, we aim  
307 at testing our approach in the real world to show its potential development. Moreover, more camera  
308 features will be modeled in our framework, such as UAVs limited time of flight.

## 309 References

- 310 1. Konda, K.R.; Conci, N. Optimal configuration of PTZ camera networks based on visual quality assessment  
311 and coverage maximization. Proc. International Conference on Distributed Smart Cameras. IEEE, 2013,  
312 pp. 1–8.
- 313 2. Lewis, P.; Esterle, L.; Chandra, A.; Rinner, B.; Yao, X. Learning to be different: Heterogeneity and efficiency  
314 in distributed smart camera networks. Proc. IEEE 7th International Conference on Self-Adaptive and  
315 Self-Organizing Systems, 2013, pp. 209–218.
- 316 3. Reisslein, M.; Rinner, B.; Roy-Chowdhury, A. Smart Camera Networks. *IEEE Computer* **2014**, *47*, 23–25.
- 317 4. Yao, Z.; Zhang, G.; Lu, D.; Liu, H. Data-driven crowd evacuation: A reinforcement learning method.  
318 *Neurocomputing* **2019**, *366*, 314–327.
- 319 5. Qureshi, F.Z.; Terzopoulos, D. Surveillance in virtual reality: System design and multi-camera control.  
320 Proc. Conference on Computer Vision and Pattern Recognition. IEEE, 2007, pp. 1–8.
- 321 6. Taylor, G.R.; Chosak, A.J.; Brewer, P.C. Ovvv: Using virtual worlds to design and evaluate surveillance  
322 systems. Proc. Conference on Computer Vision and Pattern Recognition. IEEE, 2007, pp. 1–8.
- 323 7. Altahir, A.A.; Asirvadam, V.S.; Hamid, N.H.B.; Sebastian, P.; Hassan, M.A.; Saad, N.B.; Ibrahim, R.; Dass,  
324 S.C. Visual Sensor Placement Based on Risk Maps. *IEEE Transactions on Instrumentation and Measurement*  
325 **2019**, *69*, 3109–3117.
- 326 8. Bour, P.; Cribelier, E.; Argyriou, V. Crowd behavior analysis from fixed and moving cameras. In *Multimodal*  
327 *Behavior Analysis in the Wild*; Elsevier, 2019; pp. 289–322.
- 328 9. Bisagno, N.; Conci, N.; Rinner, B. Dynamic Camera Network Reconfiguration for Crowd Surveillance.  
329 Proceedings of the 12th International Conference on Distributed Smart Cameras, 2018, pp. 1–6.
- 330 10. Motlagh, N.H.; Bagaa, M.; Taleb, T. UAV-based IoT platform: A crowd surveillance use case. *IEEE*  
331 *Communications Magazine* **2017**, *55*, 128–134.
- 332 11. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah,  
333 A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research  
334 challenges. *Ieee Access* **2019**, *7*, 48572–48634.
- 335 12. Hatanaka, T.; Wasa, Y.; Funada, R.; Charalambides, A.G.; Fujita, M. A payoff-based learning approach to  
336 cooperative environmental monitoring for PTZ visual sensor networks. *IEEE Transactions on Automatic*  
337 *Control* **2015**, *61*, 709–724.
- 338 13. Khan, M.I.; Rinner, B. Resource coordination in wireless sensor networks by cooperative reinforcement  
339 learning. 2012 IEEE International Conference on Pervasive Computing and Communications Workshops.  
340 IEEE, 2012, pp. 895–900.
- 341 14. Khan, U.A.; Rinner, B. A reinforcement learning framework for dynamic power management of a portable,  
342 multi-camera traffic monitoring system. 2012 IEEE International Conference on Green Computing and  
343 Communications. IEEE, 2012, pp. 557–564.
- 344 15. Rudolph, S.; Edenhofer, S.; Tomforde, S.; Hähner, J. Reinforcement learning for coverage optimization  
345 through PTZ camera alignment in highly dynamic environments. Proceedings of the International  
346 Conference on Distributed Smart Cameras, 2014, pp. 1–6.
- 347 16. Helbing, D.; Molnar, P. Social force model for pedestrian dynamics. *Physical review E* **1995**, *51*, 4282.
- 348 17. Micheloni, C.; Rinner, B.; Foresti, G.L. Video Analysis in PTZ Camera Networks - From master-slave to  
349 cooperative smart cameras. *IEEE Signal Processing Magazine* **2010**, *27*, 78–90.
- 350 18. Foresti, G.L.; Mähönen, P.; Regazzoni, C.S. *Multimedia video-based surveillance systems: Requirements, Issues*  
351 *and Solutions*; Vol. 573, Springer Science & Business Media, 2012.
- 352 19. Shah, M.; Javed, O.; Shafique, K. Automated visual surveillance in realistic scenarios. *IEEE MultiMedia*  
353 **2007**, *14*.
- 354 20. Junior, J.C.S.J.; Musse, S.R.; Jung, C.R. Crowd analysis using computer vision techniques. *Signal Processing*  
355 *Magazine* **2010**, *27*, 66–77.

- 356 21. Azzari, P.; Di Stefano, L.; Bevilacqua, A. An effective real-time mosaicing algorithm apt to detect motion  
357 through background subtraction using a PTZ camera. *Proc. IEEE Conference on Advanced Video and*  
358 *Signal Based Surveillance*, 2005, pp. 511–516.
- 359 22. Kang, S.; Paik, J.K.; Koschan, A.; Abidi, B.R.; Abidi, M.A. Real-time video tracking using PTZ cameras.  
360 *Proc. International Conference on Quality Control by Artificial Vision. International Society for Optics and*  
361 *Photonics*, 2003, Vol. 5132, pp. 103–112.
- 362 23. Bevilacqua, A.; Azzari, P. High-quality real time motion detection using ptz cameras. *Proc. IEEE*  
363 *International Conference on Advanced Video and Signal Based Surveillance*, 2006, pp. 23–23.
- 364 24. Rinner, B.; Esterle, L.; Simonjan, J.; Nebehay, G.; Pflugfelder, R.; Dominguez, G.F.; Lewis, P.R. Self-aware  
365 and self-expressive camera networks. *IEEE Computer* **2014**, *48*, 21–28.
- 366 25. Ryan, A.; Zennaro, M.; Howell, A.; Sengupta, R.; Hedrick, J.K. An overview of emerging results in  
367 cooperative UAV control. *Proc. 43rd IEEE Conference on Decision and Control*, 2004, Vol. 1, pp. 602–607.
- 368 26. Yanmaz, E.; Yahyanejad, S.; Rinner, B.; Hellwagner, H.; Bettstetter, C. Drone networks: Communications,  
369 coordination, and sensing. *Ad Hoc Networks* **2018**, *68*, 1–15.
- 370 27. Khan, A.; Rinner, B.; Cavallaro, A. Cooperative Robots to Observe Moving Targets: A Review. *IEEE*  
371 *Transactions on Cybernetics* **2018**, *48*, 187–198.
- 372 28. Yao, H.; Cavallaro, A.; Bouwmans, T.; Zhang, Z. Guest Editorial Introduction to the Special Issue on Group  
373 and Crowd Behavior Analysis for Intelligent Multicamera Video Surveillance. *IEEE Transactions on Circuits*  
374 *and Systems for Video Technology* **2017**, *27*, 405–408.
- 375 29. Sutton, R.S.; Barto, A.G. *Reinforcement learning: An introduction*; MIT press, 2018.
- 376 30. Esterle, L. Centralised, decentralised, and self-organised coverage maximisation in smart camera networks.  
377 2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO). IEEE,  
378 2017, pp. 1–10.
- 379 31. Vejdandarast, A.; Lewis, P.R.; Esterle, L. Online zoom selection approaches for coverage redundancy in  
380 visual sensor networks. *Proceedings of the 12th International Conference on Distributed Smart Cameras*,  
381 2018, pp. 1–6.
- 382 32. Jones, M.; Viola, P. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96* **2003**, *3*, 2.
- 383 33. Juliani, A.; Berges, V.P.; Teng, E.; Cohen, A.; Harper, J.; Elion, C.; Goy, C.; Gao, Y.; Henry, H.; Mattar, M.;  
384 Lange, D. Unity: A General Platform for Intelligent Agents, 2018, [[arXiv:cs.LG/1809.02627](https://arxiv.org/abs/cs.LG/1809.02627)].
- 385 34. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep  
386 Reinforcement Learning with a Stochastic Actor, 2018, [[arXiv:cs.LG/1801.01290](https://arxiv.org/abs/cs.LG/1801.01290)].