

Article

# Dynamic Camera Reconfiguration with Reinforcement Learning and Stochastic Methods for Crowd Surveillance <sup>†</sup>

Niccolò Bisagno <sup>1,\*</sup> , 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, 38121 Trento, Italy; alberto.xamin@studenti.unitn.it (A.X.); francesco.denatale@unitn.it (F.D.N.); nicola.conci@unitn.it (N.C.)

<sup>2</sup> Institute of Networked and Embedded Systems (NES), University of Klagenfurt, 9020 Klagenfurt, Austria; bernhard.rinner@aau.at

\* Correspondence: niccolo.bisagno@unitn.it

<sup>†</sup> This paper is an extended version of our paper published in Bisagno, N.; Conci, N.; Rinner, B. Dynamic Camera Network Reconfiguration for Crowd Surveillance. In Proceedings of the 12th International Conference on Distributed Smart Cameras, Eindhoven, The Netherlands, 3–4 September 2018.

Received: 30 June 2020; Accepted: 8 August 2020; Published: 20 August 2020

**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, as 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]. An 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 devices in the network while optimising coverage and target resolution given the current state of the crowded scene. The goal is to ensure a good resolution for common tasks such as face recognition in critical areas, while providing a sufficient video quality in others. UAVs have been particularly studied as a flexible and effective system for crowd gatherings surveillance in recent years [10,11]. Reinforcement learning approaches have shown great potential for distributed camera networks optimisation [2,12–15]. However, they have not been applied to the dynamic coverage of crowded scenes.

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

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

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

The remainder of this paper is organized as follows: Section 2 discusses related-work papers. Section 3 describes our greedy approach for camera reconfiguration. Section 3.7 introduces the evaluation metrics and Section 3.8 discusses our RL-based approach. Section 4 presents the results of our simulation study, and Section 5 provides some concluding remarks together with a discussion about potential future work.

## 2. Related Work

Cooperative video surveillance research has been developed to drastically reduce human supervision [17–19]. This is implemented by allowing cooperative cameras to share real-time information among them in order to capture events and to guarantee global coverage of the area of interest [1–3]. When observing a crowded scenario, the state of the scene evolves dynamically and the camera network should be able to reconfigure and cover events as they happen. Due to their nature, events generated by moving pedestrians are unique and often can not be reproduced, thus making it difficult to test and evaluate different camera network configurations and policies.

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

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

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

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

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

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

### 3. Method

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

#### 3.1. Observation Model

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

- an observation vector  $O_{i,j}$ , which represents the number of pedestrians detected for each cell  $c_{i,j} \in C$ ;
- a spatial confidence vector  $S_{i,j}$ , which describes the confidence of the measures for each cell  $c_{i,j} \in C$ . Our spatial confidence depends only on the relative geometric position of the observing camera and the observed cell;
- a temporal confidence vector  $L_{i,j}^t$ , which depends on the time passed since the cell has last been observed; and
- 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

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

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

Occlusion of targets is one of the main challenges in crowded scenarios. We assume that our camera network is able to robustly detect a pedestrian when its head is captured with a resolution of at 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

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

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

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

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 instant, after which the confidence drops to zero. The value  $l_{i,j}^t$  decays over time if no new observation  $o_{i,j}$  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

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

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

### 3.2. Camera Models

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

### 3.2.1. Fixed Cameras

Fixed cameras (see Figure 1a) 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

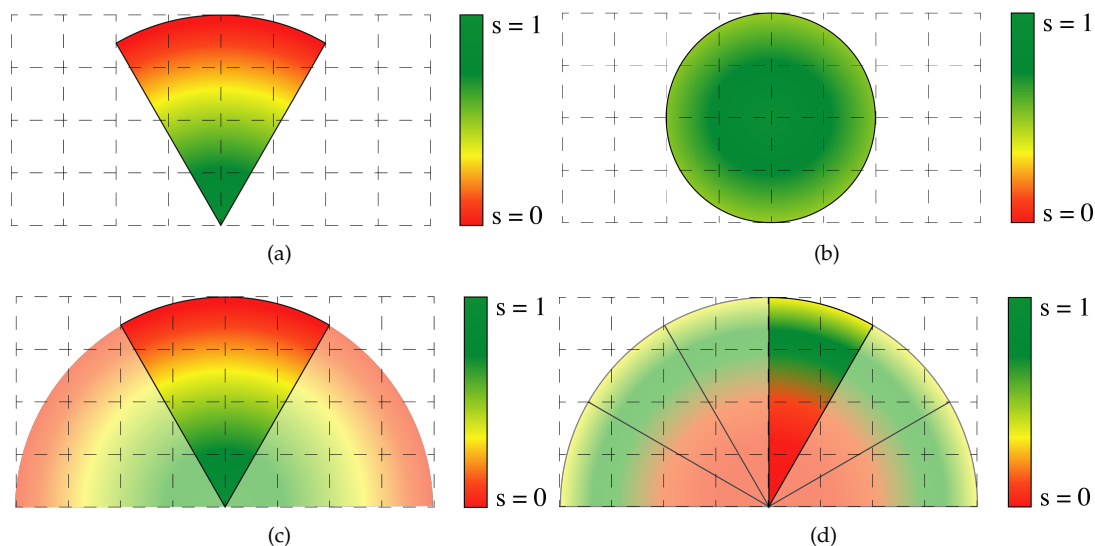
$$s(x, y) = \begin{cases} -\frac{1}{d_{max}} * d + 1 & \text{if } d < d_{max} \\ 0 & \text{if } d \geq d_{max} \end{cases} \quad (8)$$

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

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

### 3.2.2. PTZ Cameras

PTZ cameras are modeled similarly to fixed cameras, with the additional capability to dynamically change the FoV (see Figure 1c).



**Figure 1.** (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 the area surveyed by an unmanned aerial vehicle (UAV). (c) At each time step, pan-tilt-zoom (PTZ) camera can pan between different positions. (d) PTZ cameras can also zoom in to an area, which causes their FoV to shrink, but improves the spatial confidence in areas further away from the camera.

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

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

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

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

### 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 Figure 1b. The spatial confidence of point  $(x, y)$  at a distance  $d$  from the UAV is computed as

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

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

### 3.3. Reconfiguration Objective

The objective of the heterogeneous camera network is to guarantee the coverage of the scene focusing on densely populated areas. The priority metric defines the importance of each cell to be observed. A high value indicates that the cell is crowded or that we have a low confidence on 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

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

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

The objective  $G$  of each camera is to minimise the distance between the confidence vector and the priority vector

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

where  $\alpha$  can vary between 0 and 1

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

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

### 3.4. Reconfiguration Objectives: Custom Policies

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

- The reconfiguration objectives are the same for the different camera types, namely UAVs and PTZs. In the real world, UAVs have a higher cost of deployment and movement with respect to PTZs, while they provide more degrees of freedom for their reconfigurability.

- The priority maps do not share information about camera type and position between different cameras. Especially in the case of UAVs, this can lead to a superposition of different cameras, which decrease the network performances.

We propose two approaches to tackle these limitations.

The first approach, called *split priority*, is to use different priority vectors for different types 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 vectors  $P_{PTZ}^t$  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:

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

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:

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

The last operation is that every UAV normalises its priority in the range  $[0; 1]$  from the range  $[-1; 1]$  so that it is compatible with the cost function in Equation (14) to be minimised.

### 3.5. Update Function

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

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

We estimate the temporal confidence vector as follows:  $L_{time}^{t+1}$  is computed by applying Equation (5) to each element of  $L^t$ . Another temporary temporal confidence vector  $L_{new}^{t+1}$  is computed by setting to 1 all cells currently observed, and setting to 0 all the others. With the estimated temporal and spatial 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:

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

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 observed  $t^0(i, j) = t + 1$  and the corresponding observation vector  $o^t(i, j)$ .

### 3.6. Local Camera Decision: Greedy Approach

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

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

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

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

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

### 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:

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

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

### 3.8. Reinforcement Learning

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



Our novel reinforcement learning approach is based on a set of UAV-based cameras. We focus on UAV-based cameras, being a very challenging scenario, since a high number of degrees of freedom are involved. Using our predefined observation and priority models (Sections 3.1 and 3.4), we control each UAV-based camera using an RL agent, which replaces the greedy approach for local camera decision from Section 3.6 in their local decision-making process.

We rely on the vanilla ML Agents reinforcement learning network provided by [33] for our deployment with UAVs. We use Soft Actor Critic (SAC) [34] as the backbone of our RL method. We define (see Figure 2):

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

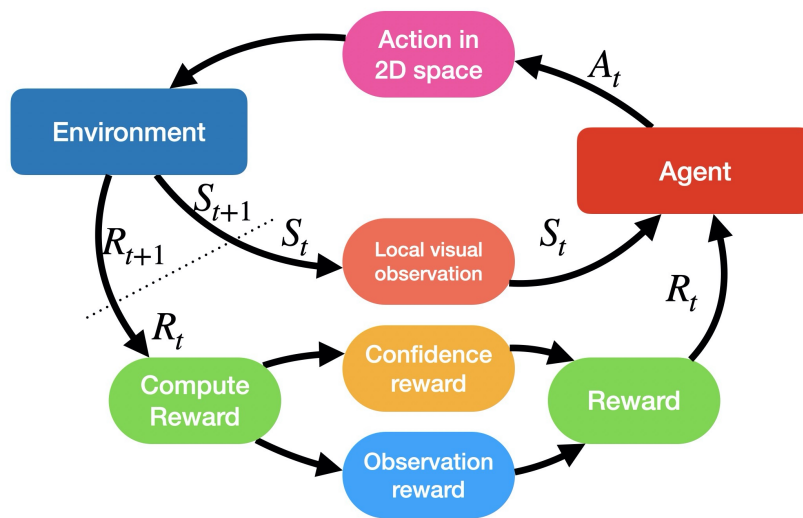


Figure 2. The workflow of our reinforcement learning (RL) approach.

At each timestep  $t$ , the agent is provided with a visual observation embedded in  $S_t \in \mathbf{S}$ , as shown in Figure 3. The visual observation consists of a texture containing 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 colour channel of the visual information is normalised to the range  $[0 - 0.1]$ . Based on 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

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

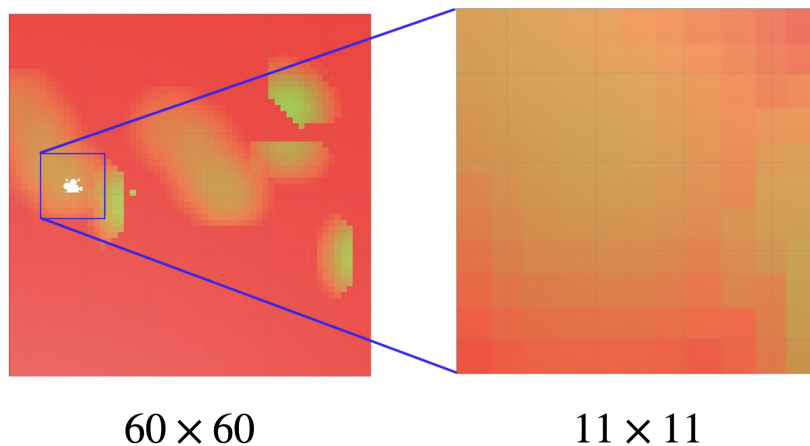
where  $\alpha$  can be set at training time to obtain the same effect described in Section 3.3. The two metrics are defined as

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

and

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

which is the instantaneous people coverage metric.



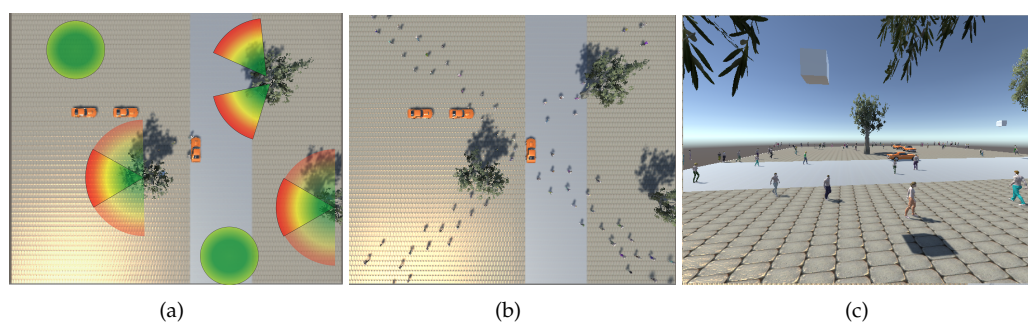
**Figure 3.** Visual observation of a drone is a  $11 \times 11$  portion of the plotted priority vector  $P^t$  in its neighbourhood.

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

#### 4. Experimental Results

For the experiments, we define an environment of size  $60 \times 60$  m<sup>2</sup>. The scene is square-shaped, exhibiting people passing by cars and vegetation. Pedestrians can enter and exit the scene from any point around the square. Each cell  $c_{i,j}$  is a square of  $1 \times 1$  m<sup>2</sup>. In this environment, 2 fixed cameras, 2 UAVs and 2 PTZs are positioned as shown in Figure 4a. Sample images of the environment from a PTZ and a UAV-based camera are shown in Figure 4b,c, respectively. For our experiments, we simulate the movement of 400 pedestrians crossing the scene with the following parameters:

- $T_{max} = 3$  s
- $PED_{max} = 2$
- $d_{max} = 10$  m
- fixed and PTZ cameras height = 5 m
- UAV-based cameras height = 7 m



**Figure 4.** (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.

##### 4.1. Quantitative Results

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

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

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

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

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

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

Experiments (7–9) are conducted using a directional crowd (Figure 4b). When the network focuses only on observation in (9), it obtains the best results in terms of PCM and the worst one in terms of global coverage GCM. As expected, we obtain the best results in terms of coverage of the environment (GCM) in experiments (3) and (6). Since the crowd is uniformly distributed in the space, we also obtain the best results in terms of PCM. In experiments (2) and (5), the network combines global coverage and crowd monitoring, the system under performs compared with the scenes where  $\alpha = 0$  and  $\alpha = 1$ .

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

| 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 2 summarises the results obtained using our *split priority* approach. Splitting the priority for different types of cameras shows how UAVs have a key role when they are allowed to focus on the global observation of the scene (experiments (10–12)). Otherwise, the performances of the whole network decreases (experiments (13–18)).

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

| ID | $g$ and $p$ | $ff_{PTZ}$ | $ff_{UAV}$ | <i>Split Priority</i> |       | <i>Position Aware</i> |       |
|----|-------------|------------|------------|-----------------------|-------|-----------------------|-------|
|    |             |            |            | 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% |

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

To fix this issue, we developed the *position-aware* method, which results are reported in Table 2. With this methodology, which includes the knowledge of the UAVs position, the performance improves. The influence on the GCM with  $\alpha_{UAV} = 0$  is almost negligible, while for greater values the improvement is clearly visible in both metrics (experiments (10–18)).

With this methodology the problem of overlapping UAVs is solved and this leads to performance improvements, as UAVs collect information in different regions.

In Table 3, we report the results obtained using our RL-based approach. Our approach (experiments (19–21)) is able to outperform the greedy approach (experiments (1–3)) when parameters  $g$  and  $p$  are set to 0.2. This method is thus more effective in long-term scenarios, when the temporal decay of the observations is slower and allows for longer-term planning. On the other hand, when  $g$  and  $p$  are set to 0.01, the greedy approach (experiments (4–6)) is more effective when cameras reconfigure rapidly and with a lower confidence threshold with respect to the the RL-based approach (experiments (22–24)).

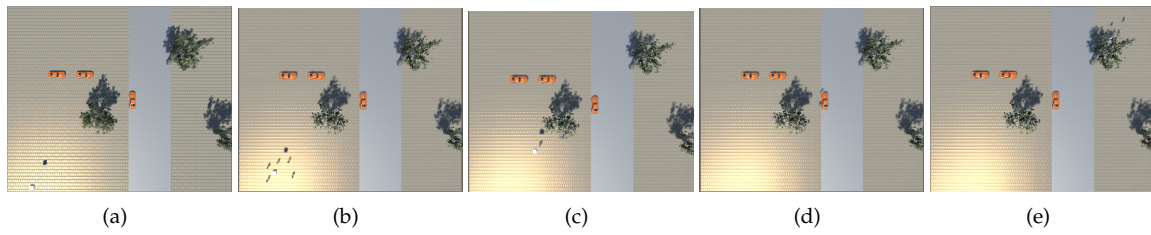
**Table 3.** Simulation experiments with RL UAV control. Mean and standard deviation are computed from the results of 3 runs of each simulation. Soft Actor Critic (SAC) is an algorithm that produces a stochastic policy, a single run would not be enough to evaluate the policy.

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

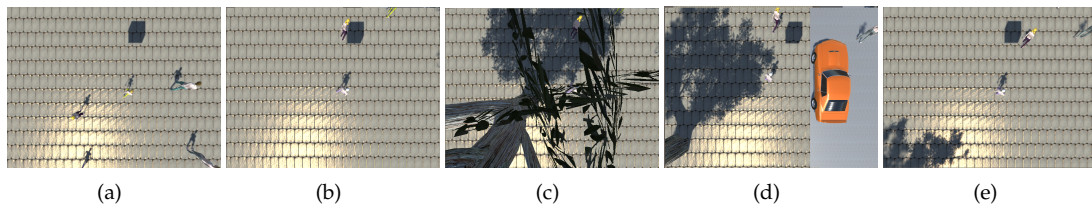
#### 4.2. Qualitative Results

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

For this purposes, we simulate a single group of five pedestrians crossing the scene from bottom left to top right, as shown in the sequence depicted in Figure 5. The UAV is able to closely follow the pedestrians in the environment, scoring a  $PCM = 70.4\%$  and  $GCM = 3.2\%$ , as shown in Figure 6. Figure 7 shows how observation priority and confidences maps are updated over time in order to guide the UAV in the tracking scenario.



**Figure 5.** Image sequence of a group of pedestrians moving from the bottom left of the environment (a) to the top right (e). The image is captured by a top view camera during the simulation to demonstrate the tracking behaviour of our network.



**Figure 6.** Image sequence of a group of pedestrians 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 $F^t$ |
|----------|----------------|-------------------|---------------------------|--------------------------|--------------------------|
| (a)      |                |                   |                           |                          |                          |
| (b)      |                |                   |                           |                          |                          |
| ore g(c) |                |                   |                           |                          |                          |

**Figure 7.** 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: (a) Camera network sample, (b) tracking sample at time  $t = 0$ , (c) tracking sample at time  $t = 10$ . In (b,c) the UAV focuses on the observation matrix, such that the next priority map depends only on previous observations. Red represents the value 0, and green represents the value 1.

### 5. Conclusions

In this paper, we have presented two camera reconfiguration approaches for crowd monitoring, a greedy camera approach and a RL-based one for UAV-mounted cameras. Our methods allow heterogeneous camera networks to focus on high target resolution or wide coverage. Although based on simplified assumptions for camera modelling and control, our approach is able to trade off coverage and resolution of the network in a resource-efficient way. We have demonstrated how different cameras can be used in different manners to optimise the effectiveness of our method. In future work, we aim

at testing our approach in the real world to show its potential development. Moreover, more camera features will be modelled in our framework, such as UAVs' limited time of flight.

**Author Contributions:** Conceptualization, N.B. and B.R.; methodology, N.B.; software, A.X. and N.B.; validation, A.X. and N.B.; formal analysis, N.C.; investigation, N.C.; resources, N.C. and B.R.; writing—original draft preparation, N.B. and A.X.; writing—review and editing, N.C. and B.R.; visualization, N.B.; supervision, F.D.N. and B.R.; project administration, F.D.N.; funding acquisition, F.D.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Konda, K.R.; Conci, N. Optimal configuration of PTZ camera networks based on visual quality assessment and coverage maximization. In Proceedings of the 2013 Seventh International Conference on Distributed Smart Cameras (ICDSC), Palm Springs, CA, USA, 29 October–1 November 2013; pp. 1–8.
2. Lewis, P.; Esterle, L.; Chandra, A.; Rinner, B.; Yao, X. Learning to be different: Heterogeneity and efficiency in distributed smart camera networks. In Proceedings of the 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems, Philadelphia, PA, USA, 9–13 September 2013; pp. 209–218.
3. Reisslein, M.; Rinner, B.; Roy-Chowdhury, A. Smart Camera Networks. *IEEE Comput.* **2014**, *47*, 23–25.
4. Yao, Z.; Zhang, G.; Lu, D.; Liu, H. Data-driven crowd evacuation: A reinforcement learning method. *Neurocomputing* **2019**, *366*, 314–327. [[CrossRef](#)]
5. Qureshi, F.Z.; Terzopoulos, D. Surveillance in virtual reality: System design and multi-camera control. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
6. Taylor, G.R.; Chosak, A.J.; Brewer, P.C. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
7. Altahir, A.A.; Asirvadam, V.S.; Hamid, N.H.B.; Sebastian, P.; Hassan, M.A.; Saad, N.B.; Ibrahim, R.; Dass, S.C. Visual Sensor Placement Based on Risk Maps. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 3109–3117. [[CrossRef](#)]
8. Bour, P.; Cribelier, E.; Argyriou, V. Crowd behavior analysis from fixed and moving cameras. In *Multimodal Behavior Analysis in the Wild*; Academic Press: Cambridge, MA, USA, 2019; pp. 289–322.
9. Bisagno, N.; Conci, N.; Rinner, B. Dynamic Camera Network Reconfiguration for Crowd Surveillance. In Proceedings of the 12th International Conference on Distributed Smart Cameras, Eindhoven, The Netherlands, 3–4 September 2018; pp. 1–6.
10. Motlagh, N.H.; Bagaa, M.; Taleb, T. UAV-based IoT platform: A crowd surveillance use case. *IEEE Commun. Mag.* **2017**, *55*, 128–134. [[CrossRef](#)]
11. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. [[CrossRef](#)]
12. Hatanaka, T.; Wasa, Y.; Funada, R.; Charalambides, A.G.; Fujita, M. A payoff-based learning approach to cooperative environmental monitoring for PTZ visual sensor networks. *IEEE Trans. Autom. Control* **2015**, *61*, 709–724. [[CrossRef](#)]
13. Khan, M.I.; Rinner, B. Resource coordination in wireless sensor networks by cooperative reinforcement learning. In Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops, Lugano, Switzerland, 19–23 March 2012; pp. 895–900.
14. Khan, U.A.; Rinner, B. A reinforcement learning framework for dynamic power management of a portable, multi-camera traffic monitoring system. In Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, Besancon, France, 20–23 November 2012; pp. 557–564.
15. Rudolph, S.; Edenhofer, S.; Tomforde, S.; Hähner, J. Reinforcement learning for coverage optimization through PTZ camera alignment in highly dynamic environments. In Proceedings of the International Conference on Distributed Smart Cameras, Venezia Mestre, Italy, 4–7 November 2014; pp. 1–6.
16. Helbing, D.; Molnar, P. Social force model for pedestrian dynamics. *Phys. Rev. E* **1995**, *51*, 4282. [[CrossRef](#)] [[PubMed](#)]

17. Micheloni, C.; Rinner, B.; Foresti, G.L. Video Analysis in PTZ Camera Networks - From master-slave to cooperative smart cameras. *IEEE Signal Process Mag.* **2010**, *27*, 78–90. [[CrossRef](#)]
18. Foresti, G.L.; Mähönen, P.; Regazzoni, C.S. *Multimedia Video-Based Surveillance Systems: Requirements, Issues and Solutions*; Springer Science & Business Media: Philadelphia, NY, USA, 2000; Volume 573.
19. Shah, M.; Javed, O.; Shafique, K. Automated visual surveillance in realistic scenarios. *IEEE MultiMedia* **2007**, *14*, 30–39. [[CrossRef](#)]
20. Junior, J.C.S.J.; Musse, S.R.; Jung, C.R. Crowd analysis using computer vision techniques. *IEEE Signal Process Mag.* **2010**, *27*, 66–77.
21. Azzari, P.; Di Stefano, L.; Bevilacqua, A. An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a PTZ camera. In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, Como, Italy, 15–16 September 2005; pp. 511–516.
22. Kang, S.; Paik, J.K.; Koschan, A.; Abidi, B.R.; Abidi, M.A. Real-time video tracking using PTZ cameras. Sixth International Conference on Quality Control by Artificial Vision. *Proc. SPIE Int. Soc. Opt. Eng.* **2003**, 103–112. [[CrossRef](#)]
23. Bevilacqua, A.; Azzari, P. High-quality real time motion detection using ptz cameras. In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, Sydney, Australia, 22–24 November 2006; pp. 23–23.
24. Rinner, B.; Esterle, L.; Simonjan, J.; Nebel, G.; Pflugfelder, R.; Dominguez, G.F.; Lewis, P.R. Self-aware and self-expressive camera networks. *IEEE Computer* **2014**, *48*, 21–28. [[CrossRef](#)]
25. Ryan, A.; Zennaro, M.; Howell, A.; Sengupta, R.; Hedrick, J.K. An overview of emerging results in cooperative UAV control. In Proceedings of the 43rd IEEE Conference on Decision and Control, Nassau, Bahamas, 14–17 December 2004; pp. 602–607.
26. Yanmaz, E.; Yahyanejad, S.; Rinner, B.; Hellwagner, H.; Bettstetter, C. Drone networks: Communications, coordination, and sensing. *Ad Hoc Networks* **2018**, *68*, 1–15. [[CrossRef](#)]
27. Khan, A.; Rinner, B.; Cavallaro, A. Cooperative Robots to Observe Moving Targets: A Review. *IEEE Trans. Cybern.* **2018**, *48*, 187–198. [[CrossRef](#)] [[PubMed](#)]
28. Yao, H.; Cavallaro, A.; Bouwmans, T.; Zhang, Z. Guest Editorial Introduction to the Special Issue on Group and Crowd Behavior Analysis for Intelligent Multicamera Video Surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 405–408. [[CrossRef](#)]
29. Sutton, R.S.; Barto, A.G. *Reinforcement learning: An introduction*; MIT Press: Cambridge, London, UK, 2018.
30. Esterle, L. Centralised, decentralised, and self-organised coverage maximisation in smart camera networks. In Proceedings of the 2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Tucson, AZ, USA, 18–22 September 2017; pp. 1–10.
31. Vejdparast, A.; Lewis, P.R.; Esterle, L. Online zoom selection approaches for coverage redundancy in visual sensor networks. In Proceedings of the 12th International Conference on Distributed Smart Cameras, Eindhoven, The Netherlands, 3–4 September 2018; pp. 1–6.
32. Jones, M.; Viola, P. *Fast Multi-View Face Detection*; Technical Report TR-20003-96; Mitsubishi Electric Research Lab.: Cambridge, MA, USA, 2003; Volume 3.
33. Juliani, A.; Berges, V.P.; Teng, E.; Cohen, A.; Harper, J.; Elion, C.; Goy, C.; Gao, Y.; Henry, H.; Mattar, M.; et al. Unity: A General Platform for Intelligent Agents. *arXiv* **2018**, arXiv:cs.LG/1809.02627
34. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv* **2018**, arXiv:cs.LG/1801.01290.

