Optimizing the Detection Performance of Smart Camera Networks Through a Probabilistic Image-Based Model

Christos Kyrkou, *Member, IEEE,* Eftychios Christoforou, *Member, IEEE,* Stelios Timotheou, *Member, IEEE,* Theocharis Theocharides, *Senior Member, IEEE,* Christos Panayiotou, *Senior Member, IEEE,* Marios Polycarpou, *Fellow, IEEE*

***Abstract*—Networks of smart cameras, equipped with on-board processing and communication infrastructure, are increasingly being deployed in a variety of different application fields, such as security and surveillance, traffic monitoring, industrial monitoring, and critical infrastructure protection. The task(s) that a network of smart cameras executes in these applications, e.g., activity monitoring, object identification, can be severely degraded because of errors in the detection module. However, in most cases higher-level tasks and decision making processes in smart camera networks (SCNs) assume ideal detection capabilities for the cameras, which is often not the case due to the probabilistic nature of the detection process, especially for low-cost cameras with limited capabilities. Realizing that it is necessary to introduce robustness in the decision process this paper presents results towards uncertainty-aware SCNs. Specifically, we introduce a flexible uncertainty model that can be used to characterize the detection behaviour in a camera network. We also show how to utilize the model to formulate detection-aware optimization algorithms that can be used to reconfigure the network in order to improve the overall detection efficiency and thus increase the effective number of detected targets. We evaluate our proposed model and algorithms using a network of Raspberry-Pi-based smart cameras that reconfigure in order to improve the detection performance based on the position of targets in the area. Experimental results in the lab as well as in a human monitoring application and extensive simulation results, indicate that the proposed solutions are able to improve the robustness and reliability of SCNs.**

***Index Terms*—Active Vision, Embedded Vision Systems, Smart Camera Networks, Dynamic Reconfiguration, Optimization Methods**

I. Introduction

SMART Camera Networks (SCNs) consist of networked cameras that collaboratively perform various computer vision tasks (activity monitoring, object identification, etc.) while monitoring an area and are thus becoming an integral component towards more intelligent cities. Recently, emerging platforms for SCNs, offer advanced sensing and processing capabilities that facilitate the development of a wide range of applications from security and surveillance, autonomous vehicles, traffic monitoring, personalized healthcare, industrial monitoring, and augmented reality [1], [2]. These advanced features enable a network of active cameras to collaborate and carry out tasks much more efficiently. The performance of applications such as activity monitoring, object identification, etc., relates directly to the detection module capabilities of the cameras in the network, which if not considered can compromise the decision making process. Also, the detection probability directly impacts the image acquisition rate for the targets which is important in applications such as activity recognition that need to capture multiple instances of the target in order to reach an outcome. Hence, it is of key importance to develop models, algorithms, and systems that take into consideration different uncertainties in SCNs (such as the detection performance) and use them to increase the robustness of the application by reconfiguring the camera network.

Realistically, even cameras featuring sophisticated visual sensors and on-board processors, are inherently uncertain due to the probabilistic nature of the machine learning algorithms used for object detection process [3], [4]. However, the majority of existing works in SCNs do not take into consideration this probabilistic aspect of such camera detection modules when developing collaborative coordination and reconfiguration algorithms [5],[6],[7],[8]. This problem intensifies in cases where low-cost camera systems are used [9], which are not equipped with the resources to efficiently run state-of-the-art detection algorithms in real-time, and so either use low resolution (which impacts the representative object resolution) or run less demanding algorithms which compromises the detection performance. Hence, there is the need to address such uncertainty issues in the context of SCNs, related to sensing and low-level processing [10]; an issue that has received little attention in the literature, especially when considering the detection module [11].

Motivated by the importance of dealing with uncertainties in SCN applications this work presents a model to introduce uncertainty awareness to the network and also develop algorithms for reconfiguring the camera network to improve overall detection performance and increase the expected number of detected targets. Our previous work in [12] presented the proposed detection model and a straightforward approach (without any optimization) of configuring the camera network

Manuscript received June 24, 2016, revised September 10, 2016, October 25, 2016, and December 2, 2016, accepted December 27, 2016. This work was supported by the European Research Council Advanced Grant through the Fault-Adaptive Project under Grant 291508.

The authors are with the KIOS Research Center for Intelligent Systems and Networks, Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus, 1678 CY e-mail: {kyrkou.christos, e.christoforou, timotheou.stelios, ttheocharides, christosp, mpolycar}@ucy.ac.cy

in the presence of a single target. In this work we significantly improve our preliminary work in the following ways:

- We formulate the problem of optimizing the overall detection probability in a smart camera network as a Mixed Integer Linear Program (MILP) and develop two optimization algorithms with different objectives for its solution.
- Introduce a systematic procedure for the determination of configurations (i.e., camera pan and tilt angles) of each camera in order to identify all possible combinations of targets that it can be simultaneously monitored by each camera individually.
- Develop algorithms for scenarios with *multiple* targets and varying number of cameras and targets and show experimental as well as simulation results.
- Comparison of the proposed approaches to a baseline case that aims to maximize the number of monitored targets without considering the detection probabilities within its optimization framework.

An experimental SCN platform, targeting low- to mid- range smart cameras, was developed based on low-cost Raspberry-Pi pan-tilt smart camera stations to validate and evaluate the presented model, algorithms, and optimization solutions. Furthermore, the proposed model and algorithms have been evaluated using simulations to test the impact of various parameters. The results show that the uncertainty aware algorithms achieve higher overall detection performance compared to approaches which do not consider the detection capabilities of cameras.

The rest of this paper is organized as follows. Section II outlines some key areas in emerging research concerning SCNs. In Section III we introduce the problem and provide a solution overview. In Section IV we outline the camera and target models and assumptions, and introduce the proposed probabilistic image-based detection model for the cameras that will serve as a basis for further development. Section V outlines an algorithm performed by each camera in the network to determine its possible configurations for target monitoring. Two optimization algorithms that utilize detection information and a baseline algorithm for comparison purposes are presented in Section VI, that determine the best configuration for each camera in the network for different criteria. In Section VII experimental evaluation results are presented, in both the lab and real-world, that demonstrate the validity of the model and the effectiveness of the dynamic reconfiguration scheme when considering an active network scenario where cameras collectively decide on how to adjust their parameters to improve the overall detection performance. Furthermore, in the same section we show simulation results regarding the effect of scaling to more cameras and targets and how it impacts the performance of the proposed algorithms. Finally, Section VIII provides concluding remarks and discusses potential improvements and future work.

II. Related Work

Recent research efforts in multi-camera networks have produced promising results for various computer vision applications (e.g., collaborative tracking [13], collaborative feature extraction [14]) especially for static cameras. Because of the increasing need for enhanced flexibility and adaptability there has been a lot of ongoing research concerning active camera networks with pan-tilt-zoom (PTZ) capabilities [15], as well as dynamic network reconfiguration [11], [16]. Improving the performance, image-based control, and automated tracking aspects of single cameras (static or PTZ) has been the subject of many works in the literature [17], [18], [19]; and such techniques can provide higher efficiency when considering a network of such cameras. However, in a multi-camera setting there are additional challenges to consider related to coordination and control, which need to be addressed [1],[20], [21].

Consequently, there has also been an increasing amount of research effort, towards adapting and improving various aspects of SCNs (utilization of resources, power conservation, area coverage) by utilizing information from multiple cameras and reconfiguring the network (i.e., changing camera orientation, position, and task assignment) [11]. For example, [22] and [23] consider energy aware allocation of vision tasks to cameras in order to increase the lifetime of battery-operated networks, while in [24] the authors propose an algorithm to optimize resource allocation in camera networks. Many works deal with the problem of configuring a camera network in such a way as to maximize the coverage of a monitored area with static cameras such as [25], [26], and usually determine the number, placement position, and orientation of cameras in the area. Our work builds on such works assuming that a deployment has already taken place and the positions of the cameras are known and fixed.

More relevant to our work are methods which steer network reconfiguration at run-time to better monitor specific targets that are present in the area. Esterle et al. [7], present a distributed market-based mechanism that handles target handover between cameras while also updating the network topology to reduce future transactions. The works in [27], [28], deal with the problem of consensus in SCNs where not all cameras observe all targets, but need to maintain a state estimate for each target. They utilize a Kalman filter in order to achieve consensus amongst neighbouring cameras regarding the position and velocity of multiple targets. The work in [29] reconfigures a network of PTZ cameras to maintain the overall coverage of the targets while compensating for viewpoint changes of cameras that opportunistically acquire high-resolution images of targets. A similar framework is used in [20], which deals with the distributed control of a PTZ camera network in order to meet various criteria for target monitoring such as, tracking error, desired resolution, and risk of losing a target. However, the detection module performance is not considered as an objective. Proactive configuration for a hybrid network, comprising of static wide view cameras and PTZ active cameras, has been proposed in [30]. Using the predicted target positions cameras are configured in such a way as to reduce the number of target handoffs and assignments in the network based on predefined goals. Natarajan et al. [31] proposed a decision-theoretic approach for the control and coordination of multiple active cameras in order to maximize the coverage of observed targets at a predefined resolution. Static

cameras are also used in the environment in order to provide location and velocity information for the targets. However, the proposed solution does not scale well with respect to the number of targets. In [6], Piciarelli et al. proposed a technique to determine the PTZ parameters of the active cameras that lead to the optimal coverage of targets, based on activity maps. The authors in [5] develop a distributed optimization mechanism for configuring the PTZ parameters of the cameras in order to permit for high resolution coverage of detected events for a given camera placement. The cameras are also equipped with a laser range node to identify obstacles, while specific tags are used to aid recognition of events. However, the detection performance and identification algorithms are not studied. A reconfiguration approach is proposed in [32] in order to select the most suitable pair of cameras to monitor a target more efficiently. The approach is based on observing human activities over extended time-periods in order to model the frequently occurring events and steer the camera configurations accordingly. However, the detection performance of each camera is not considered in the framework.

Cloud-based platforms [33], [34] have also emerged as another paradigm for on-line video analysis, where the camera views are transmitted over a wireless connection to a cloud server for processing. An example of this is the work in [35] where such a cloud infrastructure is used for on-road pedestrian tracking. In our case we assume that the cameras themselves are capable of performing the visual analysis task without the use of cloud infrastructure for processing.

Although there has been significant research in camera reconfiguration considering various problems and challenges; little attention has been given on modeling the uncertainties regarding the detection module; and formulating appropriate algorithms for selecting a network configuration with the goal of improving the overall detection performance. [5],[7],[8]. Only a few works have considered such issues in the context of SCNs. For example, the work in [36] investigates the impact of errors in the horizontal orientation (i.e., pan) of cameras during target tracking, due to initial calibration inaccuracies (modelled as Gaussian noise) and external effects that cause the camera orientation to take arbitrary values. Another example can be found in [37], which deals with the development of algorithms to compensate for faults and uncertainties related to the localization module of a camera. In this paper, an uncertainty model is presented that can describe the detection behaviour of a camera system and is used for dynamic decision making and reconfiguration to improve the overall monitoring performance of the smart camera network. This model also provides a framework for a systematic reconfiguration procedure complementing existing work in the literature.

III. Problem Description

We consider an active smart camera network consisting of a set of camera nodes $\mathcal{C}$ and a set of targets $\mathcal{T}$. Let also N_C and N_T denote the number of cameras and targets, respectively. A camera can detect a target with a certain probability based on its position and viewpoint. The objective is to configure the pan and tilt angles of the cameras in the network so that the overall detection probability for all observed targets is maximized (or equivalently the miss-detection probability is minimized), thus effectively maximizing the expected number of detected targets. This is important as it can lead to improved performance of computer vision applications such as activity monitoring, object identification, and automated inspection.

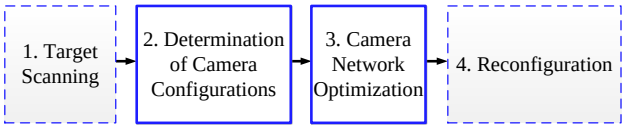


Fig. 1. Overview of the main steps towards reconfiguring the camera network

In the context of this work we define four major steps, outlined in Fig. 1, which need to take place in order to achieve the objective of maximizing the monitoring performance of the camera network. The procedure starts by determining the locations of all the observable targets in the area. This is referred to as *target scanning* and can be performed from the cameras themselves (e.g., [5]) by configuring them to first search the viewing area, or through additional cameras (e.g., [31], [21], [38], or complimentary sensors (e.g., [5]). The output of this step is the location of each target in a global coordinate system which can be used in the *determination of camera configurations* step to identify the possible sets of targets that each camera can monitor. Using this information the *camera network optimization* step finds the best configuration for each camera that maximizes the *overall* detection performance for the targets in the area thus providing improved monitoring. Finally, in the *reconfiguration* step, the new configuration for each camera is applied so that target observations can be captured and proceed with other higher level application requirements. In this work we focus on the second and third steps. Next, we describe a probabilistic detection model that captures the detection behaviour of cameras and forms the basis for the solution of the considered problem. Then we outline an algorithm to identify all possible camera configurations, and subsequent optimization algorithms are formulated to identify the best action for each camera in order to maximize the overall detection probability.

IV. System Model

In this section we describe the main components of the system model that comprises the camera, the targets, as well as the camera detection module and how it extends to multiple cameras.

A. Cameras

Cameras in active networks are endowed with degrees-of-freedom (DoF) that can be utilize in order to change their point of view and adjust to the activity in the monitored area based on collective information. In this work we consider that cameras are located at specific coordinates (x_i^C, y_i^C), $i \in \mathcal{C}$ and have two DoF, which are the pan and tilt angles denoted as θ_i^P and θ_i^T respectively (as shown in Fig. 2). Both angles

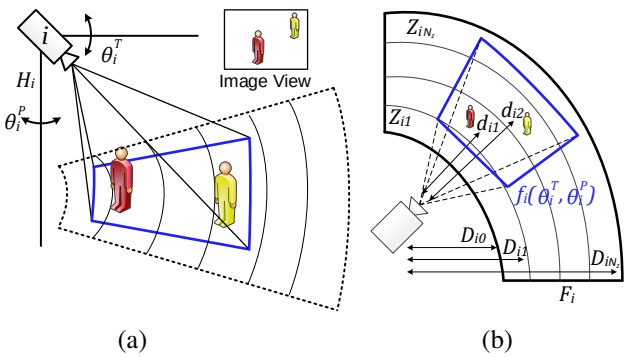


Fig. 2. (a) Camera parameters that affect configuration. (b) Zone model definition

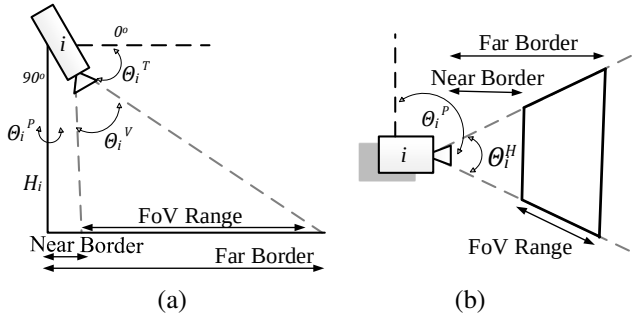


Fig. 3. How the local FoV is shaped by (a) the vertical angle of view θ_i^V. (b) the horizontal angle of view θ_i^H

can be within the full range of motion or be restricted for practical reasons ($0° \leq \theta_i^{P_{min}} \leq \theta_i^P \leq \theta_i^{P_{max}} \leq 360°$, and $0° < \theta_i^{T_{min}} \leq \theta_i^T \leq \theta_i^{T_{max}} < 90°$). All cameras i are located at a height H_i. Given a fixed height camera i can change the area that it monitors by appropriately setting the values of θ_i^P and θ_i^T. Each pair of values creates a new FoV which we denote as $f_i(\theta_i^P, \theta_i^T)$ and refer to it as local FoV. The formation of the local FoV is also determined by the viewing angles θ_i^V (vertical angle-of-view) and θ_i^H (horizontal angle-of-view) as shown in Fig. 3, which we consider to be constant. Hence, a local FoV is a subset of the global FoV F_i which includes all possible areas that camera i can monitor for the different pan and tilt angles, hence $f_i(\theta_i^P, \theta_i^T) \subseteq F_i$. A finite set of configurations $\mathcal{K}_i$ are defined for camera i, where configuration $k \in \mathcal{K}_i$ denotes a specific pan and tilt angle for the camera, following the procedure described in Section V. We define the binary variable b_{ik} which is equal to one if camera $i \in \mathcal{C}$ employs configuration $k \in \mathcal{K}_i$ and zero otherwise. A network can consist of cameras with heterogeneous features meaning different constraints on the pan and tilt angles, sensing range, and viewing angles. Furthermore, it is assumed that every camera in the network operates the same embedded target detection module which allows them to extract target images from the whole viewing frame which can be used to carry out higher level vision tasks.

B. Targets

Various applications of smart camera networks such as activity monitoring, object identification, and automated inspection, first require the detection of a target that may be present in the monitored area. Depending on its position $(x_j^T, y_j^T), j \in \mathcal{T}$ within the monitored area its distance and orientation with respect to each camera can change. As a result this will affect the detection performance of a camera especially as its distance from the camera varies which can either occlude part of it, or decrease its pixel resolution. It is assumed that the location, and distance d_{ij} of each target j relative to camera i can be determined based on the scale size and resolution that it is detected, as well as geometric information. Targets are moving within a 2D planar field. Targets are not considered as point entities but rather as occupying a 2D rectangular area, which is assigned around the centroid (x_j^T, y_j^T) of each identified target (based on preset width and length object dimensions) to account for its physical dimensions. As such, in order for a target to be detectable and able to be captured by the image processing algorithm, its whole area needs to lie within the FoV.

C. Camera Detection Model

The proposed camera detection model is based on the local FoV $f_i(\theta_i^T, \theta_i^P)$ of each camera. Through this model, we attempt to capture how the resolution of the target in the camera image affects the probability of detection, as shown in Fig. 2:

- The global FoV F_i of a camera i is segmented into m detection zones $Z_{im}, m = 1, \cdots, N_z$, where N_z is the last zone that is located further away from the camera origin.
- Zone m is the locus of points within the global FoV of camera i with distance between $D_{i(m-1)}$ and D_{im} from the camera's origin.
- A camera views a subset of the m zones which belong to its local FoV $f_i(\theta_i^T, \theta_i^P)$.
- Within each zone, there is a set of different detection probabilities. However, for simplicity we average the probabilities within the same zone and assume a uniform constant detection probability.
- When a target is in zone Z_{im} for camera i it is assumed that on average is detected with probability P_{im}. Note that the detection probability P_{im} of zone m is not given by a specific equation but is inherent to the probabilistic nature of the machine learning algorithm used for object detection.
- A camera can determine Z_{im} that a target j is detected in through trigonometry using the pan and tilt angles (θ_i^P, θ_i^T) and height H_i.
- Different cameras have distinct detection zones Z_{im} and detection probabilities P_{im}.

Each camera i in the network follows the aforementioned model and independently monitors the targets in its FoV for its current configuration k. The FoV corresponding to configuration k contains a set of targets S_{ik}, where each target j in the set is detected with non–zero detection probability $p_{ijk} = P_{im}$, such that $D_{im-1} \leq d_{ij} \leq D_{im}, m = 1, \ldots, N_z$. It follows that the miss–detection probability of target j from camera i using configuration k can be defined as $q_{ijk} =$

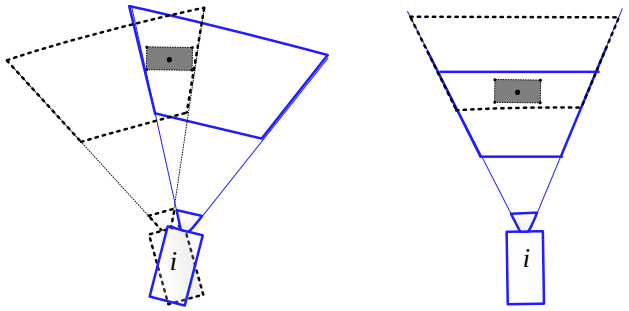


Fig. 4. Illustration of the same target as pan and tilt anchor. (a) Pan anchor at the left side - solid FoV; at the right side - dashed FoV. (b) Tilt anchor at the far border - solid FoV; at the near border - dashed FoV

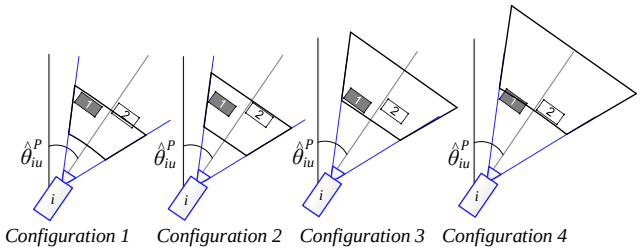


Fig. 5. Illustration of how different configurations are formed for a given anchor-pan-angle $\hat{\theta}^P_{iu}$ (target 1 is tangent to the left side of generated FoVs), for a simple scenario with two targets. Once each configuration is determined the set of targets $\mathcal{S}_{ik}$ is established: $\mathcal{S}_{i1} = \{1\}, \mathcal{S}_{i2} = \{1, 2\}, \mathcal{S}_{i3} = \{1, 2\}, \mathcal{S}_{i4} = \{2\}$.

$1 - p_{ijk}$. When a target j is observed by multiple cameras then the overall detection probability P_j for that target is the combination of the detection probabilities of all cameras. Given that detections from multiple cameras are uncorrelated the combined probabilities can be obtained through the generalized inclusion–exclusion principle. Alternatively, the combined detection probability can be expressed in a simpler form by using the product of the miss–detection probabilities:

$$P_j = 1 - \prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} q_{ijk}^{b_{ik}} \tag{1}$$

V. Determination of Camera Configurations

In order to find the best configuration for each camera that meets the given constraints for detection performance, first all the appropriate configurations that a camera can assume and the corresponding detection probabilities for all targets/objects must be identified. To find this information we devise a systematic procedure that can be executed by each camera where, given that we have extracted the Cartesian positions of the targets in the area, a camera can determine the corresponding detection probabilities by varying its pan and tilt angles, thus forming different FoVs ($f_i(\theta^T_i, \theta^P_i)$) and checking which targets are located inside each one. For each target its coordinates are registered and are available to all cameras using a global coordinate reference system. Then each camera independently runs the procedure to determine all its appropriate configurations and corresponding detection probability for each visible target.

Algorithm 1 Determining Configurations for camera i

```
% Identify targets, extract and register coordinates
for (anchor-pan-angle θ̂^P_iu ∈ A^P_i) do
    for (anchor-tilt-angle θ̂^T_iuv ∈ A^T_iu) do
        register (θ̂^P_iu, θ̂^T_iuv) for configuration k
        determine S_ik from local FoV f_i(θ̂^P_iu, θ̂^T_iuv)
        find p_ijk of targets j ∈ S_ik
        add configuration k to the set K_i if not redundant
    end for
end for
```

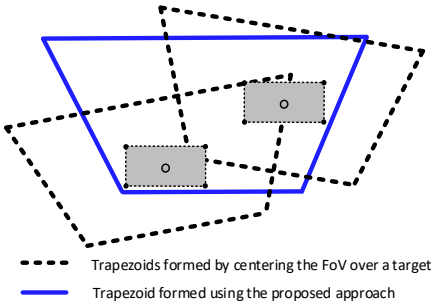


Fig. 6. Illustration of how our approach is able to find a FoV that covers both targets compared to a more intuitive approach.

The position of a camera i in an area is denoted by (x^C_i, y^C_i) which is fixed, whereas its configuration and possible FoVs that it can generate are determined by the tuple (θ^P_i,θ^T_i,H_i,θ^V_i,θ^H_i). Since we assume that H_i,θ^V_i, and θ^H_i are fixed only the former two angles are necessary to determine the local FoV. These angles are adjusted by the corresponding motorized 2 DoF pan-tilt stage on which the camera is mounted. A downwards looking camera's field-of-view is projected on the ground plane over a trapezoid area (Fig. 4). The actual dimensions of the trapezoid for specific pan-tilt angles depend on the height (H_i) of the camera mount and viewing angles of the camera θ^V_i and θ^H_i. Targets located within the FoV will be visible in the captured image, assuming no obstacles.

In principle, there exist an infinite number of possible configurations that a camera can have by changing its pan and tilt angles. To avoid the brute force approach, a systematic procedure for each camera i is performed that involves generating a finite number of configurations based on the location of the existing targets. Rather than centering the FoV at each target, we generate configurations corresponding to FoVs with a target positioned at one of the FoV boundaries, thus maximizing the number of targets within each set S_{ik} (i.e., monitor as many targets as possible with a single configuration), as shown in Fig. 6. Each target is used as reference to find a pair of pan and tilt angles referred to as *anchor angles* that define a FoV which contains a set of targets. Specifically, an *anchor-pan-angle* $\hat{\theta}^P_{iu}$, is a pan angle of camera i where a target is tangent to the left or right side of the resulting local FoV (Fig. 4a). Similarly, an *anchor-tilt-angle* $\hat{\theta}^T_{iuv}$, is a tilt angle of camera i where a target is tangent to the near or far border

of the resulting local FoV (Fig. 4b) for a given anchor-pan-angle $\hat{\theta}^P_{iu}$. These anchor angles are calculated for each target through trigonometry, resulting in the anchor-pan-angle set $\mathcal{A}^P_i$ and the anchor-tilt-angle set $\mathcal{A}^T_{iu}$. These anchor pan and tilt angles determine a unique configuration and a corresponding FoV $f_i(\hat{\theta}^P_{iu}, \hat{\theta}^T_{iuv})$. As such, by only looking at the pan and tilt angles that correspond to anchor targets we only need to examine at most $4N_T^2$ configurations.

Algorithm 1 describes the process followed individually by each camera i to identify the configurations that it can assume. First, in line 1 and 2 a camera identifies the pairs of pan and tilt angles that can form a FoV for the anchor-pan-angle and anchor-tilt-angle sets. The pair of angles define a configuration k which is temporarily marked as a possible configuration, as shown in line 4 of Algorithm 1. Next, in lines 5 and 6 the algorithm determines which targets are viewed by the camera and the corresponding detection probability. Since the distance from the camera is the major factor that affects the detection probability the same target will be viewed with the same detection probability in different configurations $k \in \mathcal{K}_i$. Also, after a certain distance the probability is zero and hence, many targets may be present within the FoV but cannot be detected by the camera. These factors lead to configurations that are the same (cover the same targets with the same probability) or are subsets of other configurations (the same and additional targets are covered with the same probability) that will incur unnecessary computational complexity for the optimization process. Hence, whenever a new configuration k is identified, it is first checked, in line 7 of Algorithm 1, if it is not a subset of any configuration with $\mathcal{K}_i$ in which case it is preserved, otherwise it is discarded as redundant. This procedure is repeated for each target j in the area to generate the configurations for the specific camera. Note that the developed camera detection model can handle occluded targets, i.e., targets which do not have direct line-of-sight with a camera due to known obstacles. This can be achieved by geometrically identifying the occluded target of each camera (no direct line-of-sight), setting the corresponding detection probabilities equal to 0 and excluding them from sets $\mathcal{S}_{ik}$.

VI. Camera Network Optimization

In order to maximize the detection capabilities of a SCN that utilizes the probabilistic camera detection model defined in Section IV, two optimization problems were formulated that maximize different performance metrics. The first proposed algorithm is the MODP (Maximize Overall Detection Probability) algorithm that takes into consideration the detection capabilities of the cameras in order to find the network configuration that maximizes the overall detection probability of all targets in the area (or conversely that minimizes the miss-detection probability). The second algorithm is called MMDP (Maximize Minimum Detection Probability) which aims to maximize the minimum detection probability of all detectable targets. In this way the MMDP ensures that no detected target has a very high miss-detection probability. Hence, both algorithms address different problems and each is suited to a different application scenario. Since, there is no previous work that examines the same problem we have developed a baseline non-probability-aware optimization algorithm for comparison, referred to as MNDT (Maximize Number of Detectable Targets), that does not utilize any detection information and instead focuses on maximizing the number of targets that are covered by the cameras.

A. Maximize Number of Detectable Targets

In order to maximize the number of detectable targets in the network we use the binary variables b_{ik} which are equal to one if camera $i \in \mathcal{C}$ employs configuration $k \in \mathcal{K}_i$ and zero otherwise, as well as u_j which denote whether target $j \in \mathcal{T}$ is detectable ($u_j = 1$) or not ($u_j = 0$). In addition, we define $\mathcal{L}_{ij}$ as the set of all configurations of camera i that can detect target j, i.e., it is not occluded. Based on these definitions, MNDT is formulated as:

MNDT Formulation

$$\max \quad \sum_{j \in \mathcal{T}} u_j \tag{2a}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}_i} b_{ik} = 1, \ i \in \mathcal{C}, \tag{2b}$$

$$\sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{L}_{ij}} b_{ik} \geq u_j, \ j \in \mathcal{T}, \tag{2c}$$

$$u_j \in \{0,1\}, j \in \mathcal{T}, \ b_{ik} \in \{0,1\}, i \in \mathcal{C}, k \in \mathcal{K}_i \tag{2d}$$

In formulation (2), the objective is to assign exactly one configuration to each camera, based on decision variables b_{ik}, in order to maximize the number of detected targets given by (2a). Constraint (2c) identifies which targets are actually detected. Two cases have to be examined to understand why this constraint provides the desired result. In the first case no camera detects target j, i.e., the sum on the left hand side of the inequality is zero, so that $u_j \leq 0$, and variable u_j is forced to be equal to zero. In the second case one or more cameras detect the target so that $u_j \leq 1$; in this case variable u_j can take either value (0 or 1), so the optimizer will choose the value maximizing the objective which is $u_j = 1$. Formulation MNDT belongs to the class of Mixed Integer Linear Programming (MILP) optimization problems which can be solved using standard optimization solvers.

B. Maximize Overall Detection Probability

The problem of maximizing the expected number of detected targets or the overall detection probability is equivalent to the minimization of the overall miss-detection probability. Recall from Section IV that the overall miss-detection probability in case of target j is equal to $\prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} q_{ijk}^{b_{ik}}$, when the detection probabilities of target j from multiple cameras are uncorrelated. Hence, MODP can be formulated as:

$$\min \quad \sum_{j \in \mathcal{T}} \prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} q_{ijk}^{b_{ik}} \tag{3a}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}_i} b_{ik} = 1, \ i \in \mathcal{C}, \tag{3b}$$

$$b_{ik} \in \{0,1\}, i \in \mathcal{C}, k \in \mathcal{K}_i \tag{3c}$$

Notice that the objective of (3) is nonlinear and solution with standard solvers is not possible. To deal with this issue we transform the problem into an equivalent problem based on [39]. Let $2^{-z_j} = \prod_{i\in\mathcal{C}} \prod_{k\in\mathcal{K}_i} q_{ijk}^{b_{ik}}$. Taking the logarithm of both sides gives $z_j = -\sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}_i} b_{ik} \log_2(q_{ijk})$, $z_j \geq 0$, and the formulation becomes

$$\min \quad \sum_{j\in\mathcal{T}} 2^{-z_j} \tag{4a}$$

$$\text{s.t.} \quad \text{Constraints (3b) - (3c)}, \tag{4b}$$

$$z_j = -\sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}_i} b_{ik} \log_2(q_{ijk}), j \in \mathcal{T}, \tag{4c}$$

$$z_j \geq 0, j \in \mathcal{T} \tag{4d}$$

The new formulation (4) is an integer programming problem with the objective function composed of separable monotonically increasing convex terms 2^{-z_j}. Following the analysis from [40], each of these terms can be tightly approximated from the convex envelop $\phi(z_j)$ of a number of piecewise linear functions. Towards this direction, let us assume that each term 2^{-z_j} is approximated by L_j linear segments with slopes $\alpha_{1,j}$,...,$\alpha_{L_j,j}$ and start-points $\beta_{1,j}$,...,$\beta_{L_j,j}$. Let us also assume that $\beta_{L_j+1,j} = z_j^{max}$. Because 2^{-z_j} is convex and monotonically increasing, the envelop approximation $\phi(z_j)$ will also be convex and the slopes will have monotone increasing values: $\alpha_{1,j} < \alpha_{2,j} < ... < \alpha_{L_j,j}$. Let $\xi_{lj}, l = 1, ..., L_j$ be the value of z_j corresponding to the lth linear segment so that $0 \leq \xi_{lj} \leq \beta_{l+1,j} - \beta_{l,j},\ l = 1, ..., L_j$. Under the assumption that $\xi_{ij} = \beta_{i+1,j} - \beta_{i,j}, i = 1, ..., l-1$ when $\xi_{lj} > 0$, it is true that $z_j = \sum_{l=1}^{L_j} \xi_{lj}$ and also that $\phi(z_j) = \sum_{l=1}^{L_j} \alpha_{l,j}\xi_{lj}$. In other words, z_j can be replaced by the sum of variables $\xi_{lj},\ l = 1, ..., L_j$ if we can ensure that the solution of the optimization problem will always be such that each ξ_{lj} is nonzero only when the variables $\xi_{lj},\ i = 1, \ldots, l-1$ have obtained their maximum value. As mentioned earlier, $\alpha_{1,j}$ has the smallest slope value and hence ξ_{1j} will be the first variable associated with z_j to be assigned a nonzero value. Only when ξ_{1j} has been assigned its maximum value variable ξ_{2j} will be assigned a nonzero value and this procedure will continue until z_j becomes equal to the sum of the nonzero variables. Thus, the assumption stated above is satisfied and formulation (4) becomes:

MODP Formulation

$$\min \quad \sum_{j\in\mathcal{T}} \sum_{l=1}^{L_j} \alpha_{l,j}\xi_{lj} \tag{5a}$$

$$\text{s.t.} \quad \text{Constraints (3b) - (3c)}, \tag{5b}$$

$$\sum_{l=1}^{L_j} \xi_{lj} = -\sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}_i} b_{ik} \log_2(q_{ijk}), j \in \mathcal{T}, \tag{5c}$$

$$0 \leq \xi_{lj} \leq \beta_{l+1,j} - \beta_{l,j},\ l = 1, ..., L_j,\ j \in \mathcal{T} \tag{5d}$$

Formulation (5) is a MILP optimization problem that can be solved with standard solvers. Increasing the number of linear segments L_j improves the approximation of 2^{-z_j} by $\phi(z_j)$ but increases the computational complexity. Hence, to compute the slopes and start-points of 2^{-z_j} we employ a piecewise linear approximation scheme that minimizes the number of linear segments limiting the maximum approximation error to a desired value proposed in [41].

C. Maximize the Minimum Detection Probability

MMDP aims to maximize the minimum detection probability of all detectable targets, which means that the developed optimization formulation has to be able to identify the maximum number of detectable targets and then maximize the minimum detection probability or equivalently minimize the maximum miss-detection probability of those targets. To achieve this we combine formulation (2) with the minimization of the maximum miss-detection probability given by function $\max_{j\in\mathcal{T}_d}(\prod_{i\in\mathcal{C}} \prod_{k\in\mathcal{K}_i} q_{ijk}^{b_{ik}})$, where $\mathcal{T}_d = \{j : u_j = 1,$ in formulation (2)$\}$ is the set of detectable targets. This objective is equivalent to the mathematical program $\{\min 2^{-\delta},$ s.t. $\prod_{i\in\mathcal{C}} \prod_{k\in\mathcal{K}_i} q_{ijk}^{b_{ik}} \leq 2^{-\delta}, j \in \mathcal{T}_d\}$, where δ is an auxiliary continuous variable which ensures that term $2^{-\delta}$ minimizes the maximum miss-detection probability of all detectable targets. Taking the logarithm of both sides of the constraints and simplifying the objective function yields

$$\max \ \delta \tag{6a}$$

$$\text{s.t.} \ \sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}_i} b_{ik} \log_2(q_{ijk}) + \delta \leq 0, j \in \mathcal{T}_d. \tag{6b}$$

Combining formulations (2) and (6) yields

MMDP Formulation

$$\max \ \delta + M \sum_{j\in\mathcal{T}} u_j \tag{7a}$$

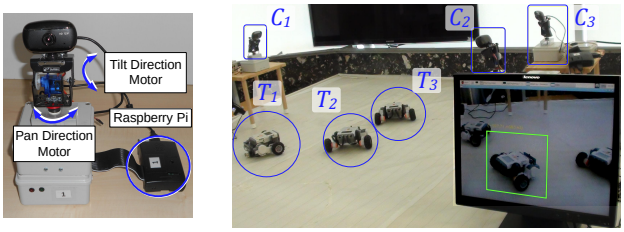
$$\text{s.t. Constraints (2b) - (2d)}, \tag{7b}$$

$$\sum_{i\in\mathcal{C}} \sum_{k\in\mathcal{K}_i} b_{ik} \log_2(q_{ijk}) + \delta + Mu_j \leq M, j \in \mathcal{T} \tag{7c}$$

Observe that we have employed the big-M approach; the presence of M in (7a) and (7c) ensures that the detectability of a new target is more important than an improvement in the value of δ and that (6b) is satisfied only for $j \in \mathcal{T}_d$, respectively. Regarding the practical value of M, $M = 20$ is high enough since an increment of δ to be more beneficial that a unit increase of the detectable targets implies a minimum miss-detection probability smaller than $2^{-20} < 10^{-6}$. Formulation MMDP is accurate and can be solved with standard MILP solvers.

VII. Experimental Setup and Evaluation Results

To evaluate the proposed model and decision-making process we have developed camera stations based on the Raspberry Pi single-board computer [42]. Each Raspberry Pi is connected with a webcam that is mounted on a motorized two DoF pan-tilt stage, as shown in Fig. 7(a). The two angular positions are controlled independently using a corresponding servo which is equipped with a potentiometer-type position sensor. The sensory feedback information allows accurate angular positioning of the pan-tilt system. The servo motors are controlled by the Raspberry Pi and the associated



(a) (b)

Fig. 7. Experimental Setup: (a) The developed Raspberry-Pi pan-tilt smart camera station. (b) The network of cameras collaboratively monitoring the moving robots. View and detection results from camera 2 are also shown.

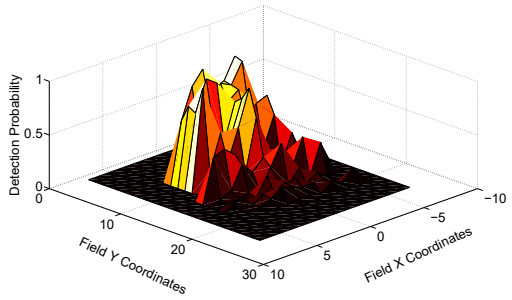


Fig. 8. Detection accuracy degradation for downwards looking cameras. Brighter color corresponds to higher detection probability.

control electronics using a pulse width modulation (PWM) approach. The object detection module of each smart camera is based on the cascade object detection algorithm with Local Binary Pattern (LBP) and Histograms of oriented Gradients (HoG) [43] features based on the seminal work by Viola and Jones [44] which is available in the OpenCV computer vision library [45]. The computer at every camera station does not only execute the computer vision algorithm but also runs a control program necessary to reconfigure according to the optimization algorithm result. Communication between the camera stations is realized via a dedicated local Wi-Fi network. Information regarding the detected targets can either be exchanged between cameras or be sent to a central station using Wi-Fi network. Either way the potential configurations for each camera are calculated and the respective detection probabilities per target are computed. Then each camera or a central server can run the optimization algorithms to find the required configuration for each camera. Finally, for the modelling and solution of the optimization problems defined in Section VI, we used the Gurobi optimization solver [46].

Our first experiments examined the validity of the proposed camera detection model described in Section IV and helped identify its parameters. Then, using a static layout, we experimentally implemented the configurations generated by the optimization algorithms in order to demonstrate the effectiveness of the optimization procedure and also verify that the expected values (calculated using the detection model and Eq. (1)) correspond to the actual camera measurements. Experiments were also performed in a dynamic scenario where the targets move within the field and the cameras dynamically reconfigure to adjust the overall detection performance. Two

TABLE I
DETECTION PROBABILITIES FOR INDIVIDUAL CAMERAS FOR THE CONFIGURATION PRODUCED BY MODP ALGORITHM

Camera ID	Outcome	Target ID					
		1	2	3	4	5	6
Camera 1	Expected	0.9	0	0	0	0.5	0
	Experimental	0.85	0	0	0	0.68	0
Camera 2	Expected	0	0	0	0.5	0	0.9
	Experimental	0	0	0	0.64	0	0.88
Camera 3	Expected	0	0.2	0	0.9	0	0
	Experimental	0	0.16	0	0.88	0	0

TABLE II
COMBINED DETECTION PROBABILITIES OBTAINED USING THE CONFIGURATIONS PRODUCED FROM THE OPTIMIZATION ALGORITHMS

Algorithm	Outcome	Target ID					
		1	2	3	4	5	6
MNDT	Expected	0.2	0.2	0.5	0.95	0.5	0.2
	Experimental	0.24	0.21	0.47	0.99	0.45	0.15
MODP	Expected	0.9	0.2	0	0.95	0.5	0.9
	Experimental	0.85	0.16	0	0.97	0.68	0.88
MMDP	Expected	0.2	0.2	0.5	0.95	0.5	0.2
	Experimental	0.18	0.23	0.48	0.91	0.46	0.23

experimental studies are presented, one using programmable robot as targets in a lab environment with applications in indoor environments [47], and a real case where people are monitored in a building lobby. To test the scalability and further investigate the properties of the proposed optimization algorithms we performed simulations for a varying number of cameras and targets based on the model and experimental setup. All experiments were conducted in non-controlled environments with ambient light and varying factors such as the position, size, and color of the targets.

A. Obtaining the Camera Detection Probabilities

The first step in the evaluation process was to verify the validity of the model described in Section IV using one of the developed Raspberry-Pi smart camera stations. The station was configured with different orientations ranging from ($\theta_i^{T_{min}} \leq \theta_i^T \leq \theta_i^{T_{max}}$) and for each one the detection performance of the target object were measured and averaged for multiple runs over 100 consecutive frames. The target object was positioned within different locations covering the whole field. Thus we effectively created a map of the detection probabilities, as depicted in Fig. 8, which illustrates the detection probabilities within the effective FoV of a camera. Notice that for this specific application as the distance of the target increases the detection probability deteriorates. This is because as the object resolution decreases, and is represented with less pixels, slight variations in a few pixels can cause the detection module to produce the wrong outcome. Of course, the dimensions and specific detection performance depend on the camera resolution as well as the detection algorithm itself, as it will become clearer in the following sections. We have used the state-of-the-art Cascade object detection algorithm available in the OpenCV computer vision library [45] which is a typical and widely used detection algorithm. Hence, the general trend is expected to remain the same and this type of modeling to be also applicable to different camera configurations. The

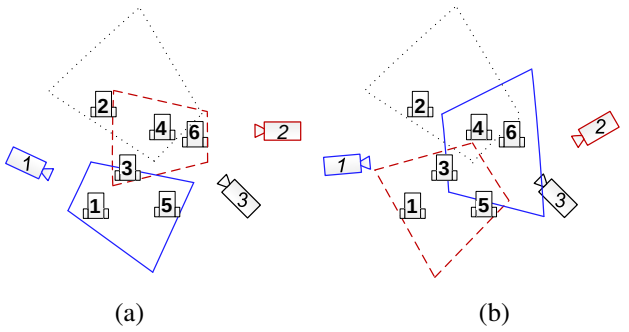


Fig. 9. FoV obtained from the experimental setup for an example scenario with 6 targets. (a) Configuration produced by the MODP algorithm. (b) Configuration produced by the MNDT and MMDP algorithm. (Camera 1 - solid line, Camera 2 - dashed line, Camera 3 - dotted line)

general processs described herein is applicable to any case and only the actual probability values and number of zones may change between different configurations as it will be shown in VII-C to accommodate the different uncertainties in various applications.

B. Experimental Evaluation of the Proposed Algorithms

The model and the extracted values of its parameters were used for further experiments in an active vision scenario. For the experiments conducted in the laboratory, we used three smart camera stations which monitored a $(1.5 \times 1.5m^2)$ area in the laboratory and programmable Lego Mindstorm robots were used as targets which are capable of executing specified paths. This allowed to perform the same experiment for different algorithms thus ensuring a fair performance comparison. The detection module was trained to recognize the Lego robots after training on positive (Lego robot images) and negative (background images) samples. The number of targets varied between 3-6 and they were placed in various positions within the field to create different scenarios. The cameras also communicate with a dedicated server PC through local Wi-Fi, in order to exchange information and coordinate their actions. The cameras were placed at the same height at arbitrary initial orientations and for every initial target placement, the cameras responded by accordingly adjusting their configurations depending on the output of the optimization algorithms. Based on the model in Section IV and experimental detection measurements we employed a 3-zone model and the assigned detection probabilities were 90% for the proximal zone, 50% for the intermediate zone and 20% for the distant one. These parameters were the same for all cameras even thought different zone models are possible when the sensing or detection algorithms are different.

1) Static Scenario: In order to validate how well the model captures the camera detection behaviour we first set up a static layout and measured the expected and experimental probability values for each target for the specific three-camera setup in Fig. 9-a. These values are collected in Table VII. The values in the table indicate that the zone-specific detection rates expected by individual cameras for specific configurations are consistent. Furthermore, we evaluate the experimental and expected results for the configurations produced by the three optimization algorithms, as shown in Fig. 9. In each case the network assumes a different configuration while trying to meet the objective of each optimization algorithm. For each algorithm we are interested in the combined detection probability for each target for all cameras that monitor it. The expected total detection performance achieved by the cameras was calculated as described in Section IV using Eq. (1). The experimental combined detection probabilities were measured by considering the detection results of the cameras for 1000 consecutive frames. Table VII shows the combined detection probabilities for each target after applying the configurations obtained from the three optimization algorithms. First, the results further validate that the expected detection probabilities assigned to each zone are in agreement with the experimental results. Second, we verify that the expected values when combining detections from multiple cameras are close to the experimental ones. For example, in Fig. 9-a cameras 2 and 3 both monitor target 4 with an expected combined detection probability $1 - (1 - 0.5) \times (1 - 0.9) = 0.95$ which agrees with the experimentally observed probability of 0.97, as shown in Table VII. The table also highlights the impact of each optimization criterion on the expected combined detection performance achieved for each target. Overall, we see that the MNDT and MMDP algorithms produce a configuration that covers all targets, while the MODP algorithm trades-off coverage for higher cumulative detection probability. These trends have also been observed in the rest of the experimental evaluation scenarios thus verifying the validity of the model and the effectiveness of the optimization algorithms. The impact of each algorithm is further illustrated in the Section VII-D where simulation results are shown for larger setups.

2) Dynamic Scenario: For this particular scenario we have extended the experimental setup with an overhead top-view camera with the sole purpose of determining the locations of all the observable targets in the area as in [21]. In this way, the *target scanning* procedure is implemented (Step 1 of Fig. 1). A top view camera is not the only option as this process can be performed by any other sensors and/or cameras [5], as outlined in Section III. It can also be performed by the cameras themselves but would require developing new strategies and approaches for coordination that go beyond the scope of this work. The server receives the target coordinates, determines the set of possible configurations for each camera (Step 2 of Fig. 1), executes the optimization algorithm under consideration (Step 3 of Fig. 1), and sends the new pan and tilt angles to the camera stations in real-time (Step 4 of Fig. 1). Based on the new configuration, the cameras execute the machine-learning-based detection algorithm and return the detection results to the server. This procedure is continuously performed to ensure that all targets are dynamically monitored. In this scenario we have used 3 smart cameras and programmed 3 Lego Mindstorm robots to execute predefined paths in order to monitor the detection performance of the solutions produced by the three optimization algorithms.

Specifically, in Fig. 10 we compare the individual camera detection probabilities with the combined detection probability for a single moving target. We observe that the combined

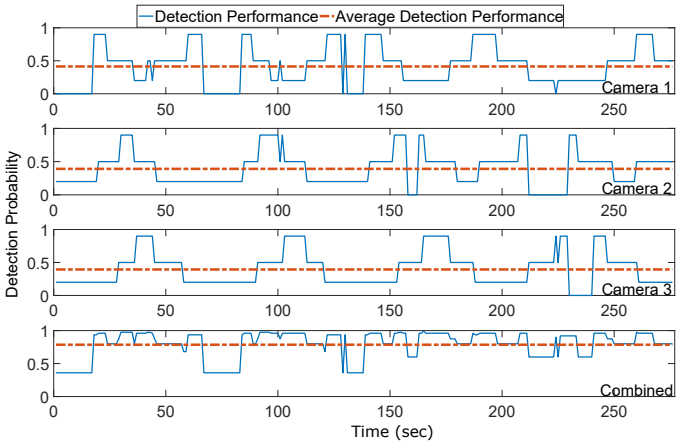


Fig. 10. Experimental results for dynamic scenario with robots: The individual detection performance of each camera as it changes with the movement of a target and the their combined detection performance.

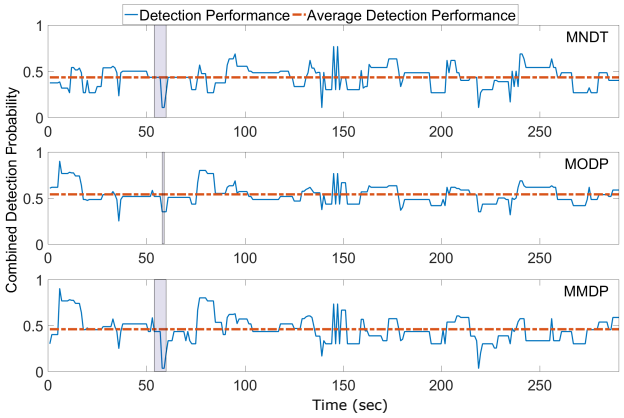


Fig. 11. Experimental results for dynamic scenario with robots:The detection performance of the network as it changes over time for the three optimization algorithms.

average detection probability is maintained at a high level (0.79) despite the individual cameras having a lower average performance (0.41, 0.38, 0.39 for cameras 1,2 and 3, respectively), as well as fluctuations in the individual camera detection probabilities. Fig. 11 illustrates the combined detection probabilities for the three optimization algorithms. Overall, we can observe that on average the two proposed optimization algorithms provide a higher combined detection performance (0.54 for MODP and 0.47 for MMDP) compared to the baseline approach (0.45 for MNDT) that does not utilize probability information. The same figure also highlights the fact that there is discrepancy between the expected model probabilities and the real observations due to environmental uncertainties (e.g., movement, lighting conditions), as there are time instances where the baseline scenario performs better than the theoretically optimal (e.g., time instances 54-60 for MNDT vs MMDP, and 58-59 for MNDT vs MODP). However, the proposed approaches outperform, on average, the baseline by up to 17% over the duration of the experiment.

With regards to the computational complexity of the proposed solutions, the total duration of the whole process to receive the target coordinates, solve the optimization problem, and set the new camera configurations in the experiments averaged around 0.018 seconds. Hence, this was not found to introduce any significant delays and permitted the smart cameras to monitor the moving targets at 15 frames-per-second, which is close to the frame-rate achieved without the networking components. Note that this execution time includes the solution of the MILP formulations as obtained through a MILP solver (Gurobi [46]), which in the worst case scenario, has exponentially increasing execution time; in future work we expect to improve execution time by developing highly effective polynomial complexity algorithms.

C. Real-Case Evaluation of proposed algorithms

To demonstrate the generalization capabilities of our framework we evaluate it under a real case scenario and a larger physical space. The smart camera network was set up in a building lobby to monitor people. The same camera stations with the previous scenario were used, this time placed at a higher position ($2.20m$) and use a different cascade detection algorithm able to detect humans, available in OpenCV. The overall setup is shown in Fig. 12. For the particular experiment, we have refined the model to reflect the fact that we are now detecting a different object (humans) in a larger operating environment ($5 \times 9m^2$). The model is suitable to handle additional challenges that were not present in the robot experiments with regards to the target (e.g., pose, appearance, size) and environment (e.g., size, lighting, reflections). For example, the full body of a person is necessary to be present in the image in order to be detectable, and also good detection modules for pedestrians often require sufficient margin around the person [48]. To deal with inherent uncertainties associated with detecting humans in a real environment, we have experimentally calibrated the model for various individuals and characteristics by measuring the average detection probability at various distances and adjusting accordingly the number of zones and corresponding probabilities. Through this procedure we established a 5-zone model with probability values $\{0.2, 0.5, 0.8, 0.5, 0.2\}$, resulting in a bell-shaped probability distribution, with an effective capturing distance of 3 to 9 metres for each camera. The existence of a non-detection region for distances below $3m$ arises because a person needs to be at a certain distance away from the camera for its full-body to be captured. This is evident in Fig. 12-c where camera 3 does not view the whole subject, and hence, is not able to detect it; in contrast to the subject in Fig. 12-d, where all cameras are able to detect it. Lower probability values in the first few zones reflect the fact that when not enough margin is present around the person it is not detected as frequently.

The detection performance of the proposed algorithms for this scenario is evaluated using 3 cameras and 4 people moving in the monitored area. People were walking with random motion patterns and different velocities. The proposed model is used to estimate the detection performance and through the optimization process reconfigure the network to meet the objective of each algorithm. The combined detection probabilities for the three optimization algorithms are illustrated

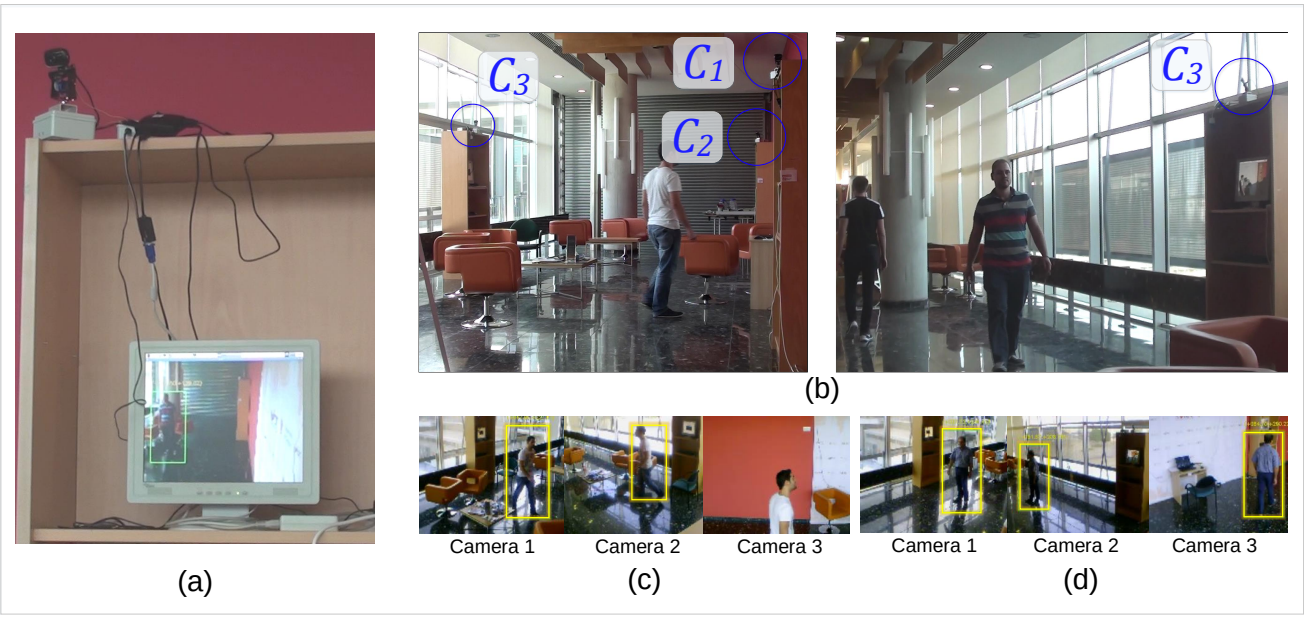


Fig. 12. Experimental setup for dynamic real-case scenario: (a) Setup for camera 1 station. (b) Two different views of the experimental setup and monitored area with different targets. (c) Detections of one subject for the 3 cameras. Notice that C_3 is not able to detect the target. (d) Detections of a different subject where all cameras are able to detect it.

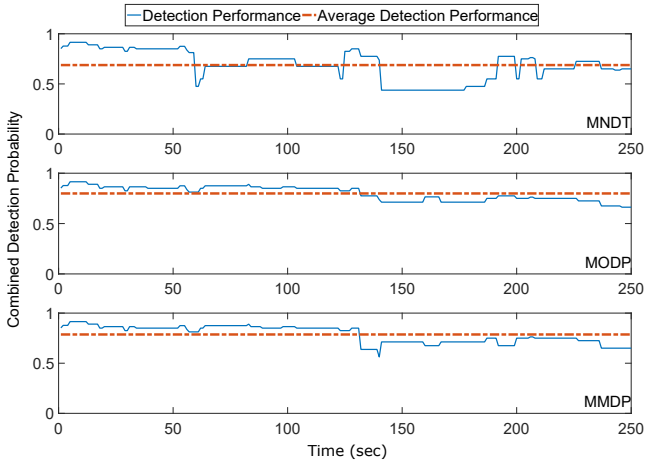


Fig. 13. Experimental results for dynamic real-case scenario: Detection performance of the network for the three optimization algorithms.

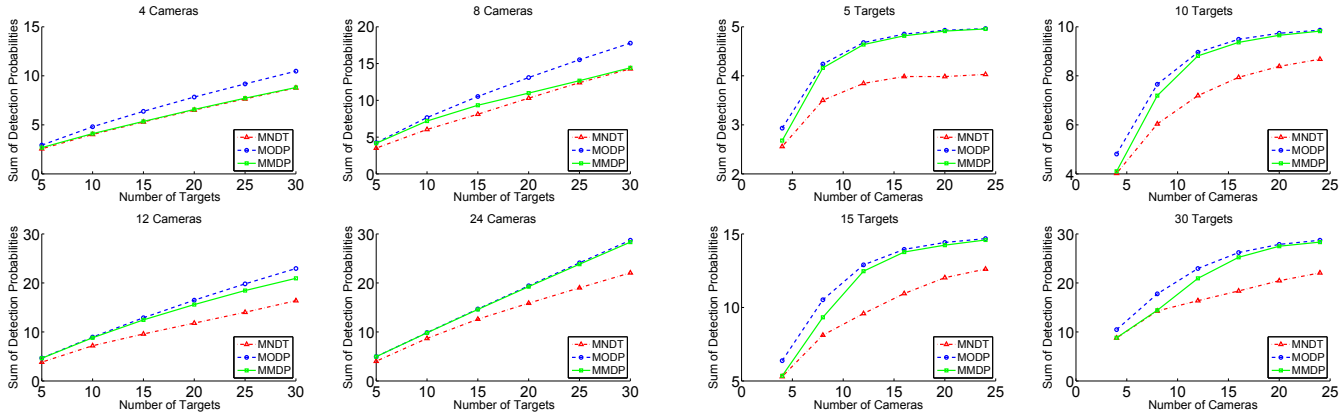
in Fig. 13. Overall, we can observe that on average the two proposed optimization algorithms provide a higher combined detection performance (0.80 for MODP and 0.78 for MMDP) compared to the baseline approach (0.68 for MNDT) that does not utilize probability information. In this scenario the proposed approaches outperformed the baseline by up to 58% for a specific time instance, and 15% on average.

D. Simulation Evaluation

Beyond the experimental evaluation which confirmed the model and the expected results we also performed simulation studies to evaluate the scalability in terms of performance of the algorithms with respect to an increasing number of cameras and targets as well as the temporal impact on overall detection of each of the optimization algorithms. We have developed a visual simulation environment in MATLAB which was very close to the real-life experimental framework while encapsulating all the aforementioned models and algorithms. Within a square field area targets were generated in random positions, while all cameras were placed in the perimeter of the area with an equal number of cameras at each side. We performed simulation experiments for different number of targets (ranging from 5 to 30, with a step of 5) and cameras (ranging from 4 to 24, with a step of 4). For each combination of targets and cameras we run 1000 different scenarios and compare the averaged outcomes for the three algorithms.

We first evaluate and compare the effectiveness of the three algorithms in terms of achieved detection performance for all targets in the area which is defined as the sum of all camera combined detection probabilities for each target $\sum_{j\in\mathcal{T}}(1-\prod_{i\in\mathcal{C}}\prod_{k\in\mathcal{K}_i} q_{ijk}^{b_{ik}})$, where remember that q_{ijk} is the miss-detection probability of camera i for target j in selected configuration k. We also examine the impact of increasing the number of cameras (Fig. 14(a)) and targets (Fig. 14(b)) respectively, in the effectiveness of the algorithm. For all experiments the MODP algorithm achieves the best overall detection probability, which is expected as it is tailored for this task. The MNDT algorithm achieves the lowest overall detection probability, as it is not aware of the detection capabilities of the cameras. Finally, the MMDP algorithm has the most interesting behaviour as it tries to balance both objectives. For the plots in Fig. 14(a) it starts by achieving the same results as the MNDT algorithm however as the number of available cameras increases, it gradually reaches the results of the MODP. Conversely, as observed by the plots in Fig. 14(b), as the number of the number of targets increases the behaviour of the MMDP drifts further away from the MODP and towards the MNDT.

Next, in Fig. 15 it is investigated how each algorithm



(a) (b)

Fig. 14. Total Detection Probability as (a) number of cameras increases. (b) number of targets increases

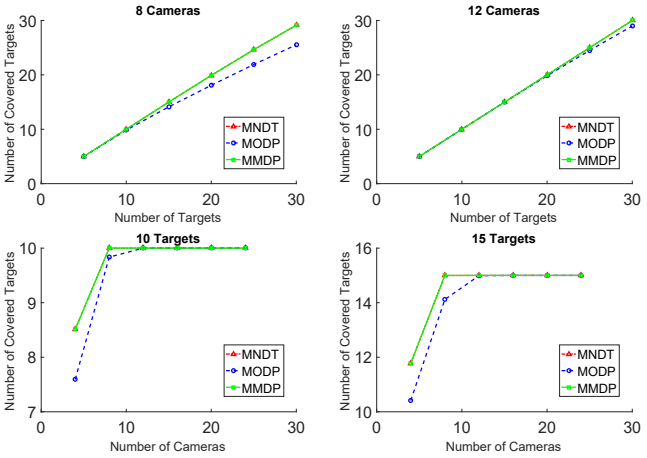


Fig. 15. Effective number of monitored targets as the number of cameras increases (top) and as the number of targets increases (bottom)

performs with regards to the total number of covered targets (targets that can be detected with a non-zero probability) given by $\sum_{i\in\mathcal{C}}\lceil\sum_{j\in\mathcal{T}}\lceil 1-\prod_{k\in\mathcal{K}_i} q_{ijk}^{b_{ik}})\rceil/N_C\rceil$. First it is observed that the MNDT and MMDP algorithms achieve the same results. On the other hand, the MODP covers fewer number of targets as it tries to maximize the overall detection probability by making cameras overlap and only covers the same number when there are sufficient number of cameras to monitor the previously uncovered targets. Notice that the MNDT and MODP algorithms do not cover all the targets either. Interestingly, these results can provide an indication on the number of cameras that need to be placed in an area to ensure maximum coverage depending on the number of targets and assignment algorithm.

Finally, we evaluate how each of the three algorithms performs in terms of the minimum achieved detection probability, for an individual target which is given by $\min_{j\in\mathcal{T}}(1-\prod_{i\in\mathcal{C}}\prod_{k\in\mathcal{K}_i} q_{ijk}^{b_{ik}})$. This is important in cases where we need to monitor targets with a certain probability which may be necessary to capture as many instances of the target as possible for better activity recognition or identification. These results are depicted in Fig. 16. As expected the MMDP algorithm demonstrates the highest minimum detection probability in all scenarios as it considers this as one of the objectives of the optimization problem. In the case of fixed number of cameras and varying number of targets (Fig. 16(a)) the MNDT algorithm outperforms the MODP algorithm up to a certain point where there are enough cameras to allow for more overlapping detections. The reason for this is that the main objective of the MODP algorithm is to maximize the overall detection probability which is achieved by seeking overlaps between targets. Hence, in some cases when the number of cameras is not sufficient enough some targets will be left with limited coverage resulting in lower detection probability. This is something that the MMDP algorithm considers and hence can be used for those types of scenarios. The opposite is true for the case of fixed number of cameras and varying number of targets (Fig. 16(b)), where the MODP outperforms MNDT for small number of targets. However, as the number of cameras increases the MODP manages to achieve better results in terms of the minimum detection probability.

The previous results focused on a static configuration where the cameras, after becoming aware of all target locations, reconfigure to maximize a specific optimization metric. To provide an indication of when the network can reconfigure again we examine how the detection probabilities change after τ time instances once a configuration has been set, which is given by $\sum_{j\in\mathcal{T}}(1-(\prod_{i\in\mathcal{C}}\prod_{k\in\mathcal{K}_i} q_{ijk}^{b_{ik}})^{\tau})$. As the detections at different time instances are independent we can combine them to obtain estimate detection probabilities over time. In this way we identify the time instances necessary for the network to confidently acquire detections from all targets. Results are obtained for different scenarios in Fig. 17. It is observed that the MODP and MMDP converge faster as the number of cameras increase, while the MNDT takes more time instances to converge. The MMDP is the fastest to converge overall in all the cases, while the MODP and MNDT change depending on the number of targets and cameras. Hence, MMDP is particularly useful when the targets are monitored with the same camera configurations for multiple time-instances.

Conclusively, the two proposed optimization algorithms demonstrate significant trends that determine their appropri-

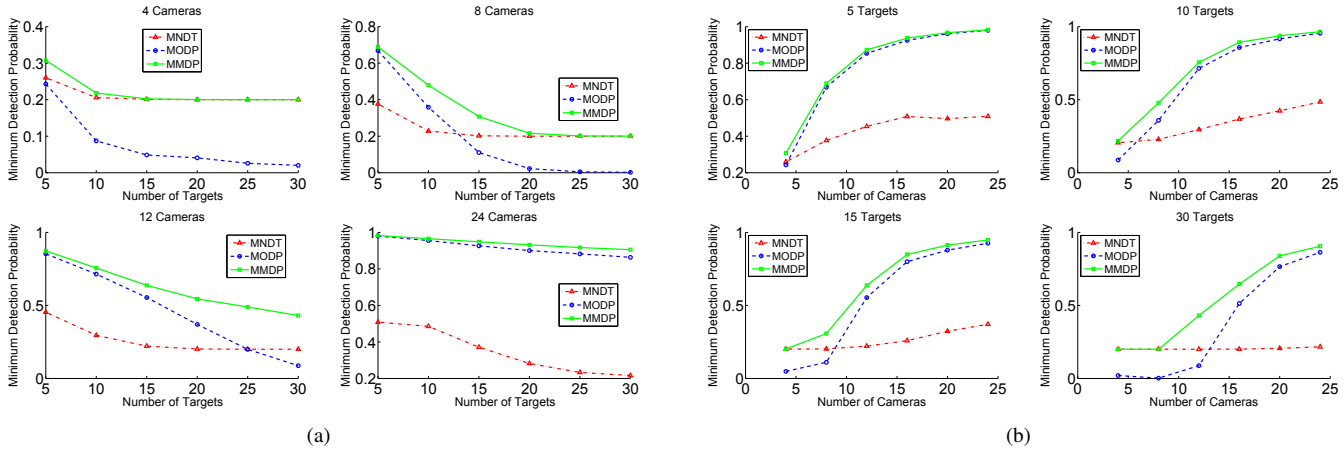


Fig. 16. Minimum Detection probability in the network as (a) the number of cameras increases and (b) as the number of targets increases

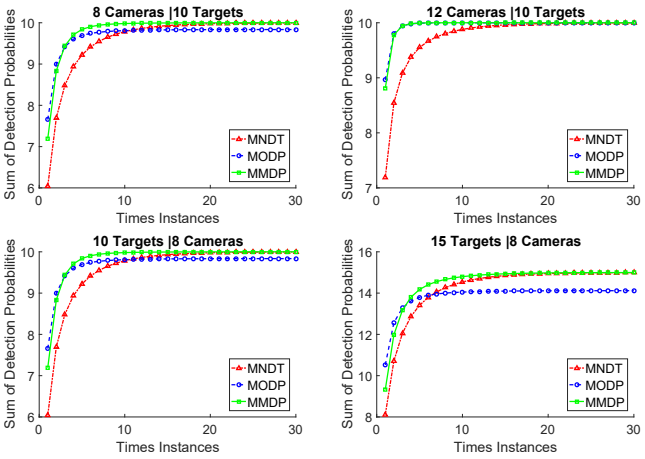


Fig. 17. Detection probability over time as the number of cameras increases (top) and as the number of targets increases (bottom)

ateness for a specific application. As expected the MODP and MMDP perform better in terms of detection performance. Overall, the MMDP algorithm offers a compromising solution which manages to increase the overall detection probabilities for the targets while also trying to maximize the coverage. It is possible to develop an adaptive approach where the MMDP algorithm can weight the two different objectives emphasizing on the one that is needed currently.

VIII. Discussion and Future Work

In this work we have developed a probabilistic image-based model that can be used to describe the detection characteristics of cameras. This model was employed to formulate optimization problems with different objectives to select the best configuration for each camera in the network in order to achieve optimal detection performance; demonstrating the importance of incorporating such a model into the configuration selection process. The proposed framework can be generalized to diverse application scenarios, in terms of the physical size of the monitored area and object, and type of detection module and object category. This generalization is achieved by appropriately adjusting the parameters of the proposed probabilistic model to fit a specific scenario. In particular, the model can scale with the size of the monitored area and object, by adjusting the number of zones as well as their physical size. In addition, the behavior of the detection module for different object categories can be captured by appropriately adjusting the probabilities in each zone. Finally, the cameras can have different model parameters and probabilities to adapt to differences in camera specifications or the area monitored by each camera. One possible drawback of our approach lies perhaps in the off-line calibration process to determine the detection probabilities which can incur additional deployment time. To address this, we are currently developing an automated approach for extracting the detection probabilities using the concept of pixels-on-target and multi-scale resolution pyramids. Another refinement is to also consider the camera zoom capabilities as an additional factor that can change the detection performance. Hence, the optimization algorithms will have to choose from a larger set of configurations for each camera. Finally, with the increasing deployment and availability of low-cost smart cameras, we are also looking to explore how additional, and perhaps mobile, cameras can be used to compensate for areas that remain unobserved despite the reconfiguration.

References

[1] H. Aghajan and A. Cavallaro, *Multi-Camera Networks: Principles and Applications*. Academic Press, 2009.

[2] F. Qureshi and D. Terzopoulos, "Smart camera networks in virtual reality," in *Proceedings of the First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDSC '07.*, Sept 2007, pp. 87–94.

[3] C. Kyrkou and T. Theocharides, "A Flexible Parallel Hardware Architecture for AdaBoost-Based Real-Time Object Detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 6, pp. 1034–1047, June 2011.

[4] ——, "Accelerating object detection via a visual-feature-directed search cascade: algorithm and field programmable gate array implementation," *Journal of Electronic Imaging*, vol. 25, no. 4, p. 041013, 2016. [Online]. Available: http://dx.doi.org/10.1117/1.JEI.25.4.041013

[5] A. Kansal, W. Kaiser, G. Pottie, M. Srivastava, and G. Sukhatme, "Reconfiguration methods for mobile sensor networks," *ACM Trans. Sen. Netw.*, vol. 3, no. 4, Oct. 2007.

[6] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Occlusion-aware multiple camera reconfiguration," in *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*, ser. ICDSC '10. New York, NY, USA: ACM, 2010, pp. 88–94.

[7] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, "Socio-economic vision graph generation and handover in distributed smart camera networks," *ACM Trans. Sen. Netw.*, vol. 10, no. 2, pp. 20:1–20:24, Jan. 2014.

[8] A. Ilie and G. Welch, "Online control of active camera networks for computer vision tasks," *ACM Trans. Sen. Netw.*, vol. 10, no. 2, pp. 25:1–25:40, Jan. 2014.

[9] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh, "CMUcam3: An Open Programmable Embedded Vision Sensor," 2007.

[10] P. Natarajan, P. K. Atrey, and M. Kankanhalli, "Multi-camera coordination and control in surveillance systems: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 4, pp. 57:1–57:30, Jun. 2015.

[11] C. Piciarelli, L. Esterle, A. Khan, B. Rinner, and G. Foresti, "Dynamic reconfiguration in camera networks: a short survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[12] C. Kyrkou, E. Christoforou, T. Theocharides, C. Panayiotou, and M. Polycarpou, "A Camera Uncertainty Model for Collaborative Visual Sensor Network Applications," in *Proceedings of the 9th International Conference on Distributed Smart Cameras (ICDSC '15)*. New York, NY, USA: ACM, 2015, pp. 86–91.

[13] S. Sunderrajan and B. Manjunath, "Multiple view discriminative appearance modeling with IMCMC for distributed tracking," in *Proceedings of the Seventh International Conference on Distributed Smart Cameras (ICDSC)*, Oct 2013, pp. 1–7.

[14] A. E. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, "Cooperative features extraction in visual sensor networks: A game-theoretic approach," in *Proceedings of Proceedings of the 9th International Conference on Distributed Smart Cameras*, ser. ICDSC '15. New York, NY, USA: ACM, 2015, pp. 80–85.

[15] C. Micheloni, B. Rinner, and G. Foresti, "Video analysis in pan-tilt-zoom camera networks," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 78–90, Sept 2010.

[16] J. SanMiguel, C. Micheloni, K. Shoop, G. Foresti, and A. Cavallaro, "Self-reconfigurable smart camera networks," *Computer*, vol. 47, no. 5, pp. 67–73, May 2014.

[17] T. Dinh, Q. Yu, and G. Medioni, "Real time tracking using an active pan-tilt-zoom network camera," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009.*, Oct 2009, pp. 3786–3793.

[18] S.-N. Lim, A. Elgammal, and L. Davis, "Image-based pan-tilt camera control in a multi-camera surveillance environment," in *Proceedings of the International Conference on Multimedia and Expo, 2003. ICME '03.*, vol. 1, July 2003, pp. I–645–8 vol.1.

[19] A. Biswas, P. Guha, A. Mukerjee, and K. Venkatesh, "Intrusion detection and tracking with pan-tilt cameras," in *Proceedings of the IET International Conference on Visual Information Engineering, 2006. VIE 2006.*, Sept 2006, pp. 565–571.

[20] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Collaborative sensing in a distributed ptz camera network," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3282–3295, July 2012.

[21] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, "Object detection, tracking and recognition for multiple smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1606–1624, Oct 2008.

[22] C. Kyrkou, T. Theocharides, C. Panayiotou, and M. Polycarpou, "Distributed Adaptive Task Allocation for Energy Conservation in Camera Sensor Networks," in *Proceedings of the 9th International Conference on Distributed Smart Cameras*, ser. ICDSC '15. New York, NY, USA: ACM, 2015, pp. 92–97.

[23] C. Ye, Y. Zheng, S. Velipasalar, and M. Gursoy, "Energy-aware and robust task (re)assignment in embedded smart camera networks," in *Proceedings of the 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2013, pp. 123–128.

[24] B. Dieber, C. Micheloni, and B. Rinner, "Resource-aware coverage and task assignment in visual sensor networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1424–1437, Oct 2011.

[25] S. Indu, S. Chaudhury, N. Mittal, and A. Bhattacharyya, "Optimal sensor placement for surveillance of large spaces," in *Proceedings of Third ACM/IEEE International Conference on Distributed Smart Cameras*, Aug 2009, pp. 1–8.

[26] F. Angella, L. Reithler, and F. Gallesio, "Optimal deployment of cameras for video surveillance systems," in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, Sept 2007, pp. 388–392.

[27] A. Kamal, J. Farrell, and A. Roy-Chowdhury, "Information consensus for distributed multi-target tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 2403–2410.

[28] ——, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, Dec 2013.

[29] A. Morye, C. Ding, A. Roy-Chowdhury, and J. Farrell, "Distributed constrained optimization for bayesian opportunistic visual sensing," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2302–2318, Nov 2014.

[30] F. Z. Qureshi and D. Terzopoulos, "Proactive ptz camera control," in *Distributed Video Sensor Networks*. Springer London, 2011, pp. 273–287.

[31] P. Natarajan, T. N. Hoang, K. H. Low, and M. Kankanhalli, "Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS '12. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 155–162.

[32] D. Karuppiah, R. Grupen, A. Hanson, and E. Riseman, "Smart resource reconfiguration by exploiting dynamics in perceptual tasks," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 1513–1519.

[33] H. Hu, Y. Wen, T. S. Chua, J. Huang, W. Zhu, and X. Li, "Joint content replication and request routing for social video distribution over cloud cdn: A community clustering method," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 7, pp. 1320–1333, July 2016.

[34] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: an analytical approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 1914–1925, Dec 2015.

[35] K. H. Lee and J. N. Hwang, "On-road pedestrian tracking across multiple driving recorders," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1429–1438, Sept 2015.

[36] M. Karakaya and H. Qi, "Collaborative localization in visual sensor networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 2, pp. 18:1–18:24, Jan. 2014.

[37] C. Laoudias, P. Tsangaridis, M. Polycarpou, C. Panayiotou, C. Kyrkou, and T. Theocharides, "Cooperative fault-tolerant target tracking in Camera Sensor Networks," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 6634–6639.

[38] B. Tavli, K. Bicakci, R. Zilan, and J. M. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools and Applications*, vol. 60, no. 3, pp. 689–726, 2012.

[39] R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin, "Exact and heuristic algorithms for the weapon-target assignment problem," *Operations Research*, vol. 55, no. 6, pp. 1136–1146, 2007.

[40] K. Murty, *Linear and Combinatorial Programming*. Krieger Publishing Company, Malabar, FL, 1976.

[41] S. Timotheou, "Asset-task assignment algorithms in the presence of execution uncertainty," *The Computer Journal*, vol. 54, no. 9, pp. 1514–1525, 2011.

[42] W. Gay, *Raspberry Pi Hardware Reference*, 1st ed. Berkely, CA, USA: Apress, 2014.

[43] M. Pietikainen, H. Abdenour, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*. Springer, 2011.

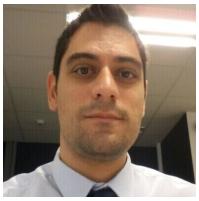
[44] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[45] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[46] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com

[47] E. Olti, T. Verbeke, G. Braeckman, V. T. Dadarlat, and A. Munteanu, "Robot tracking in low-power visual sensor networks," in *Proceedings of the 10th International Conference on Distributed Smart Camera*, ser. ICDSC '16. New York, NY, USA: ACM, 2016, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/2967413.2967420

[48] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005, pp. 886–893.



Christos Kyrkou (S'09-M'14) received the B.Sc, M.Sc. and Ph.D. degrees in Computer Engineering from the University of Cyprus in 2008, 2010, and 2014 respectively. He is currently a Research Associate at the KIOS Research Center for Intelligent Systems and Networks at the University of Cyprus, Nicosia, Cyprus. His research interests include real-time embedded systems, FPGAs and reconfigurable hardware, computer vision, machine learning, and smart camera networks. He is a technical program committee member for the IEEE International Symposium on Nanoelectronic and Information Systems (INIS) and International Conference on Pervasive and Embedded Computing (PEC). Dr. Kyrkou is a member of IEEE, ACM, and the Technical Chamber of Cyprus. He received an award for graduating top of his class during his B.Sc. studies at the University of Cyprus, and received a full scholarship for his postgraduate studies from the Department of Electrical and Computer Engineering at the University of Cyprus.

Eftychios G. Christoforou is with the University of Cyprus as a Special Scientist at the Department of Electrical and Computer Engineering and as a Research Associate at the KIOS Research Center for Intelligent Systems and Networks. He obtained a Diploma in Mechanical Engineering from the National Technical University of Athens, Greece in 1994, and a Postgraduate Diploma in Management from the Mediterranean Institute of Management, Cyprus in 1996. He received his Ph.D. in Mechanical Engineering (Robotics) from the University of Canterbury, New Zealand in 2000. Prior to his current appointment he worked as Adjunct Professor at Washington University in St. Louis, Department of Electrical and Systems Engineering, Missouri from 2004 until 2007. During the same period he also held an appointment as a Research Fellow with the Medical School, Department of Radiology at the same university. He has also held various industrial positions focusing on research and development of mechatronic systems as well as process dynamics modeling. His research interests include nonlinear and adaptive control, design/analysis and control of robotic systems, medical robotics, applications of robotics in Architecture, smart camera networks, and biomechanics.

Stelios Timotheou (S'04-M'10) received a B.Sc. from the Electrical and Computer Engineering (ECE) School of the National Technical University of Athens, and an M.Sc. and Ph.D. from the Electrical and Electronic Engineering Department of Imperial College London. He is currently a Research Associate at the KIOS Research Center for Intelligent Systems and Networks of the University of Cyprus (UCY). In previous appointments, he was a Visiting Lecturer at the ECE Department of UCY, a Research Associate at the Computer Laboratory of the University of Cambridge and a Visiting Scholar at the Intelligent Transportation Systems Center & Testbed, University of Toronto. His research focuses on the modeling and system-wide solution of problems in complex and uncertain environments that require real-time and close to optimal decisions by developing optimisation, machine learning and computational intelligence techniques. Application areas of his work include intelligent transportation systems, communication systems, disaster management and smart-camera networks.

Theocharis Theocharides (S01-M05-SM11) is currently an Assistant Professor at the Department of Electrical and Computer Engineering, at the University of Cyprus. His research focuses on the broad areas of intelligent embedded systems design, with emphasis on domain-specific architectures, evolvable and reconfigurable hardware, and design of reliable and low power embedded and application specific processors and circuits. He has authored/co-authored more than 80 papers in internationally acclaimed scientific journals and conferences. He is a senior member of the IEEE and the IEEE Computer Society, a member of the HiPEAC Network of Excellence, and currently serves on the editorial board of the IEEE Design and Test magazine, and on several Organizational and Technical Program Committee boards of various IEEE Conferences. His present work focuses on the development of hardware-friendly machine learning algorithms for big-data processing and pattern recognition in CPS, visual information extraction, distributed embedded computer vision and sensing applications, and vision-based robotic collaboration for various monitoring and visual information extraction algorithms.

Christos Panayiotou has received a B.Sc. and a Ph.D. degree in Electrical and Computer Engineering from the University of Massachusetts, Amherst, in 1994 and 1999 respectively. From 1999 to 2002 he was a Research Associate at the Center for Information and System Engineering (CISE) and the Manufacturing Engineering Department at Boston University. Since 2002-2003 he has been with the Department of Electrical and Computer Engineering, University of Cyprus where he is currently an Associate Professor. He is also a founding member of the KIOS Research Center for Intelligent Systems and Networks. His research interests include wireless, ad hoc and sensor networks, distributed control systems, fault diagnosis and fault tolerant systems, intelligent transportation systems, computer communication networks, optimization and control of discrete-event systems, resource allocation, simulation. Christos is an Associate Editor for the Conference Editorial Board of the IEEE Control Systems Society, the IEEE Transactions of Control Systems Technology, the Journal of Discrete-Event Dynamical Systems, and the European Journal of Control and has served in the program committee of numerous international conferences.

Marios Polycarpou (M'92-SM'98-F'06) is a Professor of Electrical and Computer Engineering and the Director of the KIOS Research Center for Intelligent Systems and Networks at the University of Cyprus. He received the B.A degree in Computer Science and the B.Sc. in Electrical Engineering, both from Rice University, USA in 1987, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, in 1989 and 1992 respectively. His teaching and research interests are in intelligent systems and networks, adaptive and cooperative control systems, computational intelligence, fault diagnosis and distributed agents. Dr. Polycarpou has published more than 300 articles in refereed journals, edited books and refereed conference proceedings, and co-authored 7 books. He is also the holder of 6 patents. Prof. Polycarpou is a Fellow of IEEE and IFAC. He is the recipient of the 2016 IEEE Neural Networks Pioneer Award and the 2014 Best Paper Award for the journal Building and Environment (Elsevier). He has served as the President of the IEEE Computational Intelligence Society (2012-2013), and as the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems (2004-2010). He is currently the Vice President of the European Control Association (EUCA). Prof. Polycarpou has participated in more than 60 research projects/grants, funded by several agencies and industry in Europe and the United States, including the prestigious European Research Council (ERC) Advanced Grant.