Towards Improving the Detection Performance in Collaborative Visual Sensor Networks

Christos Kyrkou, Stelios Timotheou, Eftychios Christoforou, Theocharis Theocharides, Christos Panayiotou, Marios Polycarpou
KIOS Research Center for Intelligent Systems and Networks
University of Cyprus
Nicosia, Cyprus
{kyrkou.christos, timotheou.stelios, e.christoforou, ttheocharides, christosp, mpolycar}@ucy.ac.cy

ABSTRACT

Visual Sensor Networks (VSNs) exploit the processing and communication capabilities of modern smart cameras to handle a variety of applications such as security and surveillance and critical infrastructure protection. The performance of various tasks in such applications, such as activity recognition, tracking, etc., can be severely affected by the detection module especially when considering low-cost embedded smart cameras with limited processing capabilities. Hence, this paper presents research towards the development of optimization algorithms and decision making solutions to improve the detection performance of such VSNs. Specifically, it introduces a probabilistic detection model that can be used to characterize the detection capabilities of cameras, and shows how it can be used to reconfigure VSNs. Experimental as well as simulation results indicate that the proposed solution is able to effectively improve the robustness and overall detection performance of VSNs.

Keywords

Active Vision; Self-Configuring Camera Networks; Visual Sensor Networks; Camera Sensing Model; Smart Cameras

1. INTRODUCTION

Visual Sensor Networks (VSNs) consist of networked cameras that can communicate to collaboratively monitor an area by detecting, recognizing, and tracking targets, while observing a scene [2]. Modern smart cameras offer advanced sensing and processing capabilities and collaboration capabilities that facilitate their deployment in a wide range of applications ranging from security and surveillance to industrial monitoring and personalized healthcare [14], [15]. The overall performance in these applications relates directly to the detection module capabilities of each camera in the network, since tasks such as recognition and tracking require capturing multiple instances of each target to update its state (speed, velocity, etc.). Hence, it is of key importance to develop models and algorithms that are able to characterize the behaviour of a camera detection module and reconfigure the VSN in order to improve its detection performance.

The majority of existing works assume that the detection module of the cameras in VSNs operate under perfect conditions, and do not take into account the possibility that a camera may not detect a target that is located within in its Field-of-View (FoV). However, in real-world applications even cameras featuring sophisticated visual sensors and onboard processors for decision-making, are inherently error prone due to the probabilistic nature of the detection algorithms that rely on machine-learning. This becomes more apparent for low-power and low-cost camera systems [16],[6] that can be integrated into ubiquitous cyber physical systems but do not have the necessary resources to run demanding state-of-the-art object detection algorithms. Hence, such embedded systems either lower the resolution or run less demanding algorithms both of which compromise the performance of the object detection module. However, there is limited research in dealing with this issue in VSNs.

This paper presents research towards incorporating detection performance as a key metric that can be used to reconfigure VSNs in order to improve their efficiency. To this end, the contribution of this work is twofold. First, it proposes a flexible probabilistic model that can be used to study the impact of degrading detection performance in VSN applications, and also characterize the detection capabilities of each camera in the network. Second, an optimization algorithm is formulated that utilizes the respective detection performance achieved by each camera per target, in order to set a new pan and tilt angle for each camera that results in maximizing the overall detection performance of the network. The optimization algorithm allows to maximize the detection performance for multiple targets rather than only a single target. We show the application of the model and optimization algorithm in an active network of Raspberry-Pi-based pan-tilt smart cameras that monitor targets in the field. Also, we show how the results are affected through simulations for varying number of cameras and targets.

The rest of this paper is structured as follows. Section 2 outlines some key areas of emerging research in VSNs. In Section 3 we formally introduce the problem as well as assumptions for the visual sensors, targets, and the proposed detection model. Also, in this section we formulate an optimization algorithm that utilizes detection probability information in order to identify new camera configurations that maximize the overall target detection probability. In Section 4 we present the evaluation results for the proposed model and optimization algorithm both experimentally and through simulations. Finally, Section 5 provides concluding remarks and discusses directions for future work.

2. RELATED WORK

There has been an increasing amount of emerging research in VSNs towards developing collaborative and distributed vision algorithms, with an emphasis on PTZ cameras [5] and networks [9], as well as dynamic network reconfiguration [17]. For example, [7] and [8] deal with the problem of naivety in static VSNs where not all cameras observe all targets, but need to maintain a state estimate for each target. To address this problem the authors introduce a multi-target information consensus algorithm that handles the issues of naivety as well as estimation errors in tracking and data association. The work in [10] formulates a game-theoretic approach, so that cameras can opportunistically identify time instances where the network can reconfigure in order to meet tracking requirements of targets. The work in [18] investigates how to model the probability of targets entering or exiting from certain areas in order to steer the cameras towards monitoring those areas, whereas in [13], the authors consider a 3D environment where the height of targets plays an important role in the application, and reconfigure the network of cameras based on an activity relevance map. Finally, the authors in [12] propose a reconfiguration scheme for SCNs in order to reduce the uncertainty in the targets location and movement. As such, existing models and approaches used in VSNs assume that a visual sensor will always detect a target that is present in its FoV, and do not consider the uncertainty in the detection module of a smart camera. In this paper, we present research towards the development of a probabilistic detection model that captures the behaviour of the detection module and can thus provide additional input to decision-making and dynamic configuration algorithms. Additionally, we formulate an optimization algorithm that can take advantage of probabilistic detection information, such as the one provided by the proposed model, in order maximize the overall detection performance.

3. PROBLEM DESCRIPTION AND SOLUTION OVERVIEW

We consider an active network consisting of N_C smart camera nodes i that belong in the set $\mathcal{C}$ and N_T targets j in the set $\mathcal{T}$ that are present in the area that is monitored. The objective is to configure the pan and tilt angles of the network cameras so that the overall cumulative detection probability of all targets is maximized (or equivalently minimize the miss–detection probability), thus effectively maximizing the expected number of detected targets.

3.1 Visual Sensor Model

Cameras in the network are considered to be active, in which case they have some degrees of freedom and can adjust their point of view based on collective or local information. For instance, it is possible that they can move in space, thus changing their location (x_i^C, y_i^C), or they can remain in a specific location but change their pan Θ_i^P and tilt Θ_i^T parameters (Fig. 1). All cameras i have a sensing range R_i, and are located at a height H_i. The camera monitors a specific area which is denoted as F_i^L and represents its local (current) FoV, and is a subset of the total area that a camera is able to monitor, denoted as F_i^G. We assume that a camera can change the pan and tilt angles by a fixed step and so the set of all configurations is finite. Specific values of these parameters correspond to a single configuration k that camera i can have from all possible finite configurations K_i. The active configuration for each camera i is specified by a binary variable x_{ik} which is equal to one if the camera employs configuration $k \in \mathcal{K}_i$ and zero otherwise. We also assume that the cameras in the network are calibrated so that associations between them can be established, and that geometric information is available so that cameras can localize targets. It is assumed that the location (x_j^T, y_j^T), and distance d_{ij} of target j from camera i can be determined on the scale size and resolution that it the target is detected. Each camera uses the above information to coordinate with other cameras regarding common target views. We assume that targets can move within the monitored area with a speed that permits their detection from the cameras.

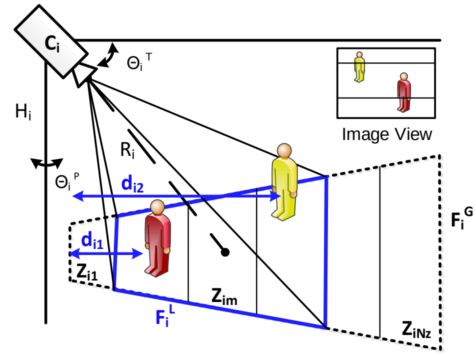


Figure 1: Camera Detection Model

VSN applications are associated with a moving target that can change its position and viewpoint orientation, and thus may affect the detection performance of a camera especially as its distance from the camera increases which decreases the its pixel resolution. Hence, camera i can detect the targets that are inside its local FoV F_i^L primarily depending on how far they are from it. Hence, the probability of detection is based on the sensing range R_i and local FoV of each camera.

Through the following model we attempt to capture how the resolution of the target in the camera image affects the detection probability. The main characteristics of the model are also shown in Fig. 1. First, the global FoV of 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 location. Depending on its current local FoV F_i^L a camera views a subset of the m zones as shown in Fig. 1. Within each zone, a target can be detected with a range of probabilities. Thus we characterize each zone with an average probability and assume a uniform constant detection probability for simplicity. Hence, when a target is in zone Z_{im} of camera i it is assumed that on average is detected with probability P_{im}. A zone Z_{im} has a higher detection probability if it is closer to camera location $(x_i^C,\ y_i^C)$, $P_{im} > P_{in}$, for $m < n$. A camera can establish the zone Z_{im} that a target j is detected through trigonometry using the pan and tilt angles of current configuration k. Hence, we define the detection probability of camera i for target j using configuration k as $p_{ijk} = P_{im}$, and subsequently the miss–detection probability is $q_{ijk} = 1 - p_{ijk}$.

3.2 Optimization Algorithm for Configuration Assignment

In order to improve the detection capabilities of a VSN we formulate an appropriate optimization algorithm that utilizes the information from the aforementioned camera detection model in Section 3.1 to select an appropriate configuration for each camera i in the network in order to maximize the overall detection probability (and also maximize the expected number of observed targets). A key step in this process is to identify all possible configurations and subsequent targets that can be monitored with non-zero probability. This can be done through a systematic process where given that all target j positions are known (which can be achieved through wide-view static cameras [12]), each camera i generates all possible configurations $k \in K_i$ and determines the set of target located within each F_i^L. We will not focus on the process of identifying camera configurations and target sets as it goes beyond the scope of this paper and since the optimization algorithm can operate either on an exhaustive list of configurations, or one where only a significant subset is present.

The above problem is equivalent to the minimization of the overall miss–detection probability. Hence, when multiple cameras cover the same target j then the overall detection probability for that target can be found using the product of the miss–detection probabilities as $1 - \prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} q_{ijk}^{x_{ik}}$ when the detections are uncorrelated. Hence the algorithm can be formulated as:

$$\min \quad \sum_{j \in \mathcal{T}} \prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} q_{ijk}^{x_{ik}} \tag{1a}$$

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

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

Notice that the objective of (1) 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 [3]. Let $2^{-z_j} = \prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} q_{ijk}^{x_{ik}}$. Taking the logarithm of both sides gives $z_j = -\sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{K}_i} x_{ik} \log_2(q_{ijk})$, $z_j \geq 0$, and the formulation becomes:

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

$$\text{s.t.} \quad \text{Constraints (1b) - (1c)}, \tag{2b}$$

$$z_j = -\sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{K}_i} x_{ik} \log_2(q_{ijk}), j \in \mathcal{T}, \tag{2c}$$

$$z_j \geq 0, j \in \mathcal{T} \tag{2d}$$

The new formulation (2) is an integer programming problem with the objective function composed of separable monotonically increasing convex terms 2^{-z_j}. Following the analysis from [11], 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 (2) becomes:

$$\min \quad \sum_{j \in \mathcal{T}} \sum_{l=1}^{L_j} \alpha_{l,j}\xi_{lj} \tag{3a}$$

$$\text{s.t.} \quad \text{Constraints (1b) - (1c)}, \tag{3b}$$

$$\sum_{l=1}^{L_j} \xi_{lj} = -\sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{K}_i} x_{ik} \log_2(q_{ijk}), j \in \mathcal{T}, \tag{3c}$$

$$0 \leq \xi_{lj} \leq \beta_{l+1,j} - \beta_{l,j},\ l = 1, ..., L_j,\ j \in \mathcal{T} \tag{3d}$$

Formulation (3) is a MILP optimization problem that can be solved with standard solvers. 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 as proposed in [19].

4. EVALUATION RESULTS

4.1 Experimental Setup

To evaluate the proposed model and optimization algorithm we have developed a network of smart cameras based on the Raspberry Pi single-board computer [6]. Each Raspberry Pi is connected with a webcam that is mounted on a motorized two degrees-of-freedom (DoF) pan-tilt stage, as shown in Fig. 3. The servo motors are controlled by the Raspberry Pi and the control electronics using a pulse width modulation (PWM) approach. Communication between the camera stations is realized via a dedicated local Wi-Fi network. Each camera station is also fitted with programmable LEDs that indicate the status of the system. The cameras were also able to calculate an estimate of the targets position in a global reference system using geometric information and the current angle configurations. The target objects were remote controlled cars. For this reason, we trained an image classifier capable of detecting cars using the Cascade Object Detection Algorithm [20] with Local binary Pattern (LBP) features which is available in the OpenCV computer vision library [4]. The training set was constructed using the database from [1] and was enhanced with additional sample images with a total of 800 positive and 3200 negative samples. The experiments were conducted in non-controlled environments with ambient light. For the camera detection model we employed a 3-zone approach and the detection probabilities for each zone were 90% for the proximal zone, 50% for the intermediate zone and 20% for the distant one. A different number of zones can be employed with respective detection probabilities, however, depending on the application scenario and operating environment.

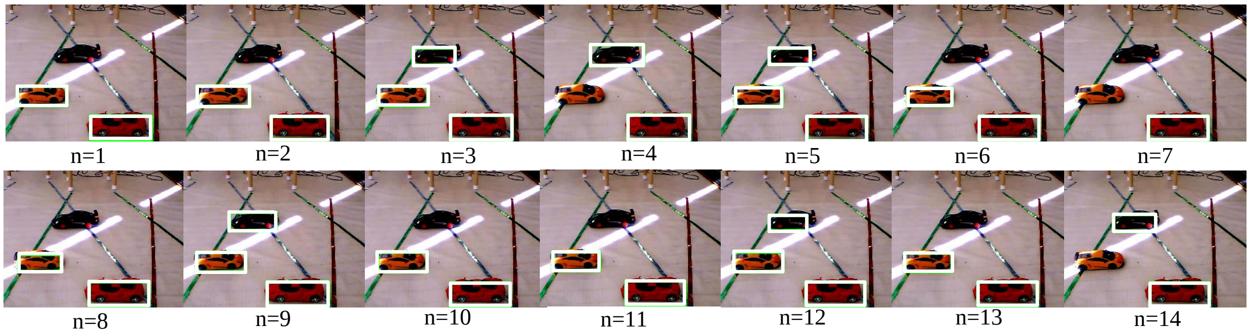


Figure 2: Detection results from a single camera for successive frames (n is number of frame). Notice that the detection performance deteriorates (less bounding boxes) for targets located further from the camera.

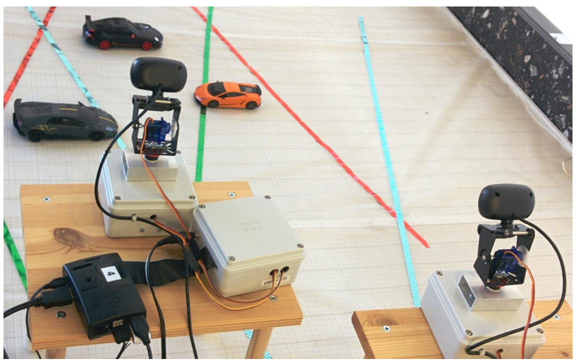


Figure 3: Setup of Raspberry-Pi Cameras in the network monitoring the targets

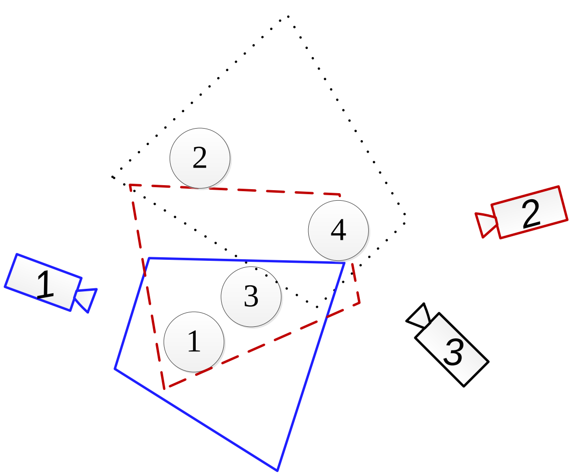


Figure 4: New configuration applied by the optimization algorithm

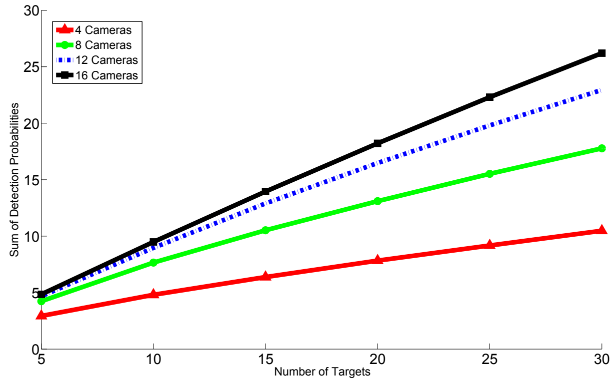
4.2 Experimental Results

The extracted values and model were used to configure the camera stations for the experiments. Each station runs a supervised learning machine trained to detect the object of interest, and communicates wirelessly with a central server that runs the optimization algorithm and transmits back the new configuration parameters. Information exchanged between the stations includes a notification with the camera's ID each time a target was detected, the target's coordinates (derived from its position in the image and the joint rotations of the pan-tilt stage) as well as the detection probability for the target (corresponding to the spatial zone in which it was detected). The targets were placed in various positions within the field and the cameras proceed to calculate the corresponding detection probabilities and available configurations depending on the targets. The central server received all the data and computed new pan and tilt angles for each camera that maximized the network detection performance based on the outcome of the optimization algorithm. Following, we calculated the corresponding detection probabilities achieved for each target.

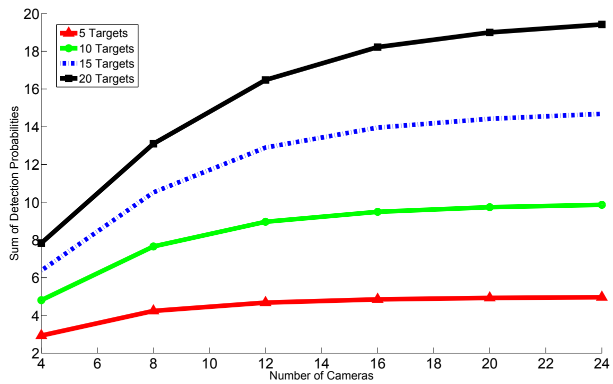
Table 1: Per Camera and Overall Detection probabilities for the configuration produced by the optimization algorithm

Camera ID	Outcome	Target ID 1	2	3	4
Camera 1	Expected	0.9	0	0.5	0
	Experimental	0.85	0	0.51	0
Camera 2	Expected	0.2	0	0.5	0
	Experimental	0.17	0	0.65	0
Camera 3	Expected	0	0.2	0	0.9
	Experimental	0	0.16	0	0.88
Overall	Expected	0.92	0.2	0.75	0.9
	Experimental	0.97	0.17	0.89	0.88

The optimization algorithm outcomes were verified using the three camera setup and up to four car targets in the monitored area. In Table 1 the detection probabilities achieved by individual cameras as well as the overall combined probabilities for each target are shown for a specific scenario. Furthermore, it also shows the expected value using the detection model from Section 3.1 and the actual measured detection probability. In this particular example we can observe how the optimization algorithm operates in order to produce a solution. Cameras 1 and 2 are configured to focus on targets 1 and 3 as they add more value towards maximizing the detection probability. Also notice that the measured



(a)



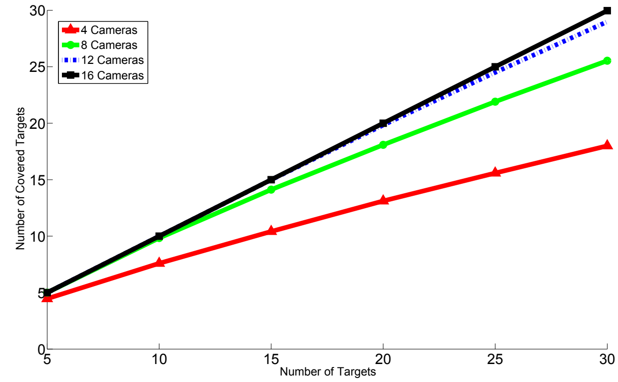
(b)

Figure 5: Combined Detection Probability as (a) the number of cameras increases. (b) the number of targets increases

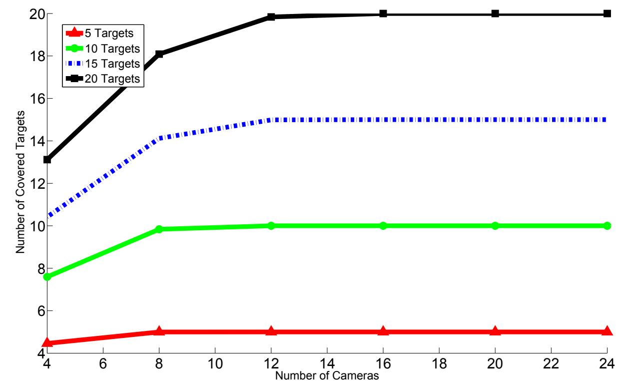
values are indeed close to the expected ones. Minor discrepancies are due to the fact that the cameras may detect the target at the same time instance, in which case the combined values will be lower that the sum, or at different times, in which case the value will approach their sum.

4.3 Simulation Results

The simulation scenarios involved a square area where targets were generated at random positions and moved at predetermined structured paths. An equal number of cameras are placed at each side of the square field and we assume that there are no obstacles in the area. We performed simulation studies for different number of targets (ranging from 5 to 20, with a step of 5) and cameras (ranging from 4 to 16, with a step of 4). For each combination of targets and cameras we run 1000 different scenarios and averaged the results across all runs. First Fig. 5 shows the combined detection probability (sum of all combined target probabilities) for all network cameras. Second, Fig. 6 shows the effective number of targets that are covered by the network of cameras (i.e. targets detected with a probability greater that zero). Together these two figures illustrate how the algorithm behaves with the increasing number of cameras and targets. As expected with a few cameras and high number of targets, the network is not able to fully cover all the targets. In such a case the optimization algorithm will configure the cameras to focus on the targets with a high detection probability. As



(a)



(b)

Figure 6: Effective number of monitored targets as (a) the number of cameras increases and (b) as the number of targets increases

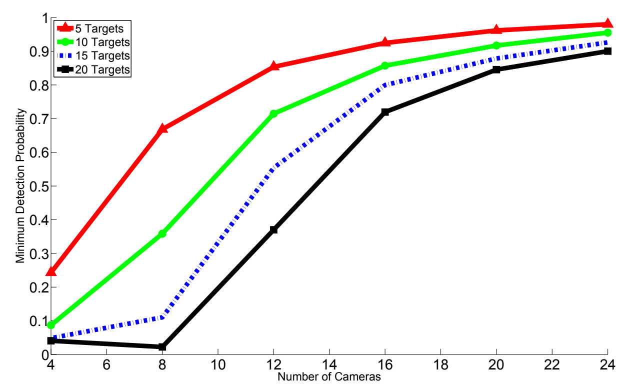


Figure 7: Minimum detection probability that the targets are monitored with

the number of cameras increases the optimization approach manages to find solutions which maximize the overall detection performance and all the targets can be covered, also it tries to increase overlaps between cameras so that the miss–detection probability is reduced. Another important result that stems from Fig. 7, that shows the minimum detection probability out of all targets, is that through this analysis we can determine how many cameras must be placed in an area in order to guarantee that a targets will be detected with a given probability. Again notice that as the number of cameras increases and the targets covered by multiple cameras are increased, the minimum detection probability with which a target can be detected with is also increased.

5. CONCLUDING REMARKS

This paper presented research towards improving the detection performance of VSNs consisting of low-cost embedded smart cameras. Through the utilization of a model to characterize the detection behaviour of smart cameras and an optimization algorithm that can make use of such information, we were able to improve the overall detection performance by reconfiguring the cameras in the network. We have evaluated the proposed model and optimization algorithm though experiments using real smart cameras, as well as through simulations studies.

The effort going forward will be on identifying possible improvements on the proposed solution. Some issues that require further research concern the efficient and optimal identification of the possible configurations that a camera can have, and how to achieve fairness so that no target remains uncovered. Finally, it is worth exploring a distributed implementation of the optimization algorithm so that it can be run on the cameras themselves and thus reduce communication overheads.

6. ACKNOWLEDGMENTS

This work was supported in part by the ERC Advanced Grant "Fault-Adaptive", ERC grant agreement no 291508.

7. REFERENCES

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, Nov. 2004.

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

[3] R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin. Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research*, 55(6):1136–1146, 2007.

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

[5] 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.*, pages 3786–3793, Oct 2009.

[6] W. Gay. *Raspberry Pi Hardware Reference*. Apress, Berkely, CA, USA, 1st edition, 2014.

[7] 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)*, pages 2403–2410, June 2013.

[8] A. Kamal, J. Farrell, and A. Roy-Chowdhury. Information weighted consensus filters and their application in distributed camera networks. *IEEE Transactions on Automatic Control*, 58(12):3112–3125, Dec 2013.

[9] C. Micheloni, B. Rinner, and G. Foresti. Video analysis in pan-tilt-zoom camera networks. *IEEE Signal Processing Magazine*, 27(5):78–90, Sept 2010.

[10] A. Morye, C. Ding, A. Roy-Chowdhury, and J. Farrell. Distributed constrained optimization for bayesian opportunistic visual sensing. *IEEE Transactions on Control Systems Technology*, 22(6):2302–2318, Nov 2014.

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

[12] 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*, AAMAS '12, pages 155–162, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[13] C. Piciarelli, C. Micheloni, and G. L. Foresti. Automatic reconfiguration of video sensor networks for optimal 3d coverage. In *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, Aug 2011.

[14] F. Z. Qureshi and D. Terzopoulos. Proactive ptz camera control. In *Distributed Video Sensor Networks*, pages 273–287. Springer London, 2011.

[15] B. Rinner and W. Wolf. An introduction to distributed smart cameras. *Proceedings of the IEEE*, 96(10):1565–1575, Oct 2008.

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

[17] J. SanMiguel, C. Micheloni, K. Shoop, G. Foresti, and A. Cavallaro. Self-reconfigurable smart camera networks. *Computer*, 47(5):67–73, May 2014.

[18] Z. Shuai, S. Yoon, S. Oh, and M.-H. Yang. Traffic modeling and prediction using sensor networks: Who will go where and when? *ACM Trans. Sen. Netw.*, 9(1):6:1–6:28, Nov. 2012.

[19] S. Timotheou. Asset-task assignment algorithms in the presence of execution uncertainty. *The Computer Journal*, 54(9):1514–1525, 2011.

[20] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.