Optimizing Multi-Target Detection in Stochastic Environments with Active Smart Camera Networks

Christos Kyrkou
KIOS Research and Innovation Center of Excellence
University of Cyprus
Nicosia, Cyprus
kyrkou.christos@ucy.ac.cy

Stelios Timotheou
KIOS Research and Innovation Center of Excellence
University of Cyprus
Nicosia, Cyprus
timotheou.stelios@ucy.ac.cy

Theocharis Theocharides
KIOS Research and Innovation Center of Excellence
University of Cyprus
Nicosia, Cyprus
ttheocharides@ucy.ac.cy

Christos Panayiotou
KIOS Research and Innovation Center of Excellence
University of Cyprus
Nicosia, Cyprus
christosp@ucy.ac.cy

Marios Polycarpou
KIOS Research and Innovation Center of Excellence
University of Cyprus
Nicosia, Cyprus
mpolycar@ucy.ac.cy

ABSTRACT

Active Smart Camera Networks (SCNs) have a wide range of applications in areas such as banks, airports and underground terminals, where it is necessary to monitor open plan spaces reliably and robustly. However, the monitoring efficiency in such environments is affected by various factors which are stochastic in nature, such as the target movement and arrival times, as well as the capabilities of the on-board camera detection module. This paper presents a modelling framework and a subsequent optimization-based solution towards the optimal reconfiguration of the camera network in order to minimize the expected number of undetected targets under uncertainties. The proposed solution is applicable to a variety of scenarios and does not require to have a full view of the area. Simulation results indicate that the proposed solution is robust to varying conditions and is able to achieve good monitoring performance with only a few cameras.

Keywords

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

1. INTRODUCTION

Smart Camera Networks (SCNs) of actively controlled pan-tilt cameras with advanced sensing and processing capabilities, are progressively being deployed in the monitoring of public and private open plan spaces such as airport and underground terminals, government and military premises [12],[2]. As the size and complexity of camera networks increase, there is a need for improved robustness, reliability, scalability, adaptability, and reduced human intervention especially with regards to actively controlled cameras (e.g., pan-tilt capabilities) [7]. One of the most important challenges faced towards achieving these goals is related to the stochastic nature of the monitored environment such as the arrival time and trajectory of targets of interest. In addition, there are also uncertainties in the camera object detection modules, which if not considered may result in undetected targets [1] especially when considering low-cost embedded smart cameras with limited processing capabilities. Hence, it is necessary to develop intelligent algorithms to actively control the cameras to collaboratively and autonomously monitor the whole area and to detect as many targets as possible under uncertainties. Furthermore, it is often the case that cameras may not be able to offer adequate coverage of the whole monitored area at once. Hence, it is also important to develop exploration methods for the observation of new targets that enter the monitored area.

This work presents a novel modelling framework that considers the stochastic nature of target movement and arrival times, to represent the expected number of targets over space in the monitored area. Using this framework, a novel optimization-based approach is further developed to obtain the configuration of each camera (i.e., pan and tilt angles) that minimizes the miss–detection probability of each target over time, hence, maximizing the expected number of targets detected at least once. The proposed solution leverages on a probabilistic model that captures the detection performance of cameras as a function of distance that we proposed in [1]. The modelling framework and corresponding solution approach are evaluated through simulations based on the specifications derived from a Raspberry-Pi-based pan-

[1] In this research we consider a target *observed* by the network if it has been detected by the camera network at least once.

tilt camera, and is compared to the ideal case where the target positions are *a-priori* known, demonstrating comparable detection performance and better execution time.

The rest of this paper is structured as follows. Section 2 outlines some key areas of emerging research related to this work. In Section 3 we formally introduce the problem as well as assumptions for the visual sensors, targets, and the proposed framework. In Section 4 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 5 we present the evaluation results for the proposed framework through simulations. Finally, Section 6 provides concluding remarks and directions for future work.

2. RELATED WORK

A large body of research in SCNs relevant to our work concerns area and target coverage [7]. In the former case the main objective is to fully cover an observed area and monitor the activity within it. In the latter case the main objective is to cover some particular PoI that correspond to targets with known positions [9],[4],[13]. In such works the cameras are assumed to be able to monitor a spherical sector and change their horizontal orientation to observe different areas by selecting a pan angle from a discrete set. In this setting a target is either covered or not covered by a camera, hence, the notion of the camera detecting and extracting target information is not present. To the best of our knowledge, there is no relevant study that addresses the same problem of maximizing the detection performance of SCNs for moving targets under uncertainty. However, to highlight our contributions we provide an overview of related works and discuss the main differences with our approach.

In [6] a particle swarm optimization technique is used to optimize the coverage both locally for targets and globally for the area. The local coverage is based on a quality of view metric which is characterized as a Gaussian distribution that depends on the distance of a target from the camera, while only zoom and pan are controllable parameters. We go beyond this work by accounting for the actual detection performance of cameras, which can vary for different targets and distances, instead of static targets. In addition, we identify activity areas where new targets may appear and hence, we do not need to constantly view the whole area. The work in [8], addresses the problem of covering each target by a minimum number of cameras in a balanced way as to ensure adequate monitoring in the presence of faults or blocked line-of-sight, however, they do not examine detection aspects. In our case we impose no limits on the number of cameras, as we use the actual detection performance of the available cameras (which can also encapsulate the presence of occlusions). In [9] a Centralized Force-Directed Algorithm is proposed to maximize the number of targets covered. This work solves a basic instance of the problem with static targets so that the movement and discovery of new targets are not considered. A study on full-view coverage, where the targets need to be viewed with the least number of cameras but from different viewpoints, is presented in [5]. Even though it is a very important problem the work considers only cameras which can change their pan orientation and do not investigate the computer vision parameters. Natarajan et al. [10], tackle the problem of target tracking by a hybrid network of active and static cameras, through a Markov Decision Process. In contrast, no static cameras are used to provide target positions in our work. Instead, we exploit the stochastic nature of target generation and motion to build an activity map [11] within the area, and then use the probabilistic camera model to maximize the detection performance over time as new targets enter the area.

3. MODELLING

In this paper we investigate how to maximize the detection of multiple targets in the area using only active cameras, without any additional sensors or static cameras to provide their location. To tackle this problem we model various aspects in the environment such as entrance/exit points, target mobility and arrival rates, the area under surveillance, and camera sensing and detection algorithms. We utilize the modeling framework to develop an exploration algorithm and a suitable optimization algorithm that together can direct the active cameras in monitoring locations where undetected targets are most probable to appear.

3.1 Entrance/Exit Points and Paths

We assume that the monitored area has a total of N^E entrance/exit points. Each entrance/exit point r is located at position (x_r^E, y_r^E), where $r \in \mathcal{E} = \{1, \dots, N^E\}$. A target can enter from point r_1 and exit from point r_2, $r_1 \neq r_2$; such that this pair of points defines path p_l, with $\mathcal{P}(l) = \{r1, r2\}$. The rate at which targets arrive at point r_1 and subsequently exit from point r_2 is characterized by a Poisson distribution $P(\mu)$ with parameter $\mu = \lambda_l \times \tau$ where λ_l is the mean number of arrivals per time step for path l, and τ is an interval of time.

3.2 Targets

Each target $j \in \mathcal{T}$ that enters the area moves with a constant speed u_j generated from a Gaussian distribution with mean μ_u and variance σ_u^2, $u_j \sim Gaussian(\mu_u, \sigma_u^2)$. In our modelling framework we assume that each target moves along the shortest path between its entrance and exit point, which is a straight line formed by the points $r1, r2\}$ when there are no obstacles and is characterized by an inclination angle θ_l. However, the true target trajectory can deviate from the straight line. Each target is characterized by a position vector X_j in the field (x_j^T, y_j^T) and has a distance d_{ij} from each camera i. It is assumed that a target enters the area at time t_j^a and departs at time t_j^d, hence, the total time that the target is present within the field is $t_j = t_j^d - t_j^a$. We assume that target j is traversing the field according to the dynamic discrete-time linear model

$$X_j(t) = X_j(t-1) + u_j \Delta t[\sin(\theta_l); \cos(\theta_l)] + \omega_j(t), \qquad (1)$$

where $\omega_j(t) \in \mathbb{R}^{2\times1} \sim \mathcal{N}(0, \sigma_j^2)$, is the motion noise vector, and Δt is the time-step of the discrete-time model.

3.3 Camera and Sensing Model

We consider a set of active camera nodes $i \in \mathcal{C}$ with two degrees of-freedom (DoF), the pan and tilt angles. The position of each camera i in an area is denoted by (x_i^C, y_i^C), whereas its configuration and possible FoVs that it can view are determined by the tuple $(\Theta_i^P, \Theta_i^T, \Theta_i^V, \Theta_i^H, H_i)$ which are the pan and tilt angles, the vertical and horizontal viewing angles, and the height that it is located at. We assume that H_i,θ_i^V, and θ_i^H are fixed, so that the local FoV

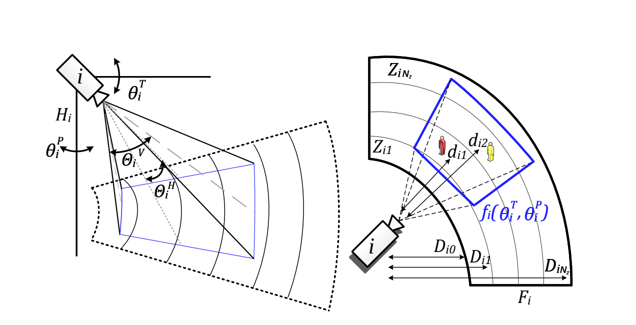

Figure 1: Sensing and detection model for downwards looking cameras.

f_i is determined using only θ_i^P and θ_i^T. The whole area that a camera can monitor with its whole range of angles is F_i. These angles are adjusted by the corresponding motorized pan-tilt stage on which the camera is mounted and are bounded as follows: $0 \leq \Theta_i^{P_{min}} \leq \Theta_i^P \leq \Theta_i^{P_{max}} \leq 360$ and $0 \leq \Theta_i^{T_{min}} \leq \Theta_i^T \leq \Theta_i^{T_{max}} \leq 90$. By projecting the 3D visual sensing camera model of camera i onto the 2D field, the camera FOV can be approximated by a trapezoid shape, as shown in Fig. 1, determined by (θ_i^P, θ_i^T). Each pair of angles defines a different configuration k from all possible configurations $\mathcal{K}_{it}$ that camera i can have at time t, which can change over time depending on the activity in the area.

3.3.1 Sensing Model

The sensing model, first introduced in [1], attempts to capture how different parameters such as the proximity of a point of interest that we want to monitor from the cameras, the noise, viewpoint of each camera, and the probabilistic nature of the underlying machine learning algorithms impact the performance of object detection algorithms and subsequently the monitoring efficiency. In this description we assume that the point of interest is the actual target but the approach is also applicable when we want to monitor locations of interest for potential targets, as discussed in Sec. 3.4. The model can be summarized as follows: The total area F_i that can be monitored by camera i is segmented into N^Z zones Z_{im},where $m = 1, \cdots, N^Z$ where N^Z is the last zone that is located further away from the camera. Each Zone m is the locus of points within the global FoV of camera i with distance between $D^Z_{i(m-1)}$ and D^Z_{im} from the camera's origin, as shown in 1. For each configuration that a camera can have it views a subset of the m zones which belong to its local FoV $f_i(\theta_i^T, \theta_i^P)$. Each zone is characterized by different detection probabilities that depend on the sensitivity of the machine learning algorithm to the visibility of the features, illumination changes, and noise. However, it is assumed that each zone is characterized by a single constant detection probability which is the average of the different probabilities within that zone. Hence, when a target is in zone Z_{im} of camera i it is assumed that on average is detected with probability P^Z_{im}. Note that the detection probability P^Z_{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.

In each time step t, camera i monitors targets located in its FoV based on the aforementioned model. We can evaluate how well a camera performs as follows: The FoV corresponding to its current configuration $k \in \mathcal{K}_{it}$ contains a set of targets $\mathcal{S}_{ikt}$, where each target j in the set is detected with non–zero detection probability $P_{ijkt} = P^Z_{im}$, where $D^Z_{im-1} \leq d_{ij} \leq D^Z_{im}, m = 1, \ldots, N^Z$. It follows that the miss–detection probability of target j from camera i using configuration k can be defined as $Q_{ijkt} = 1 - P_{ijkt}$. When a target j is observed by multiple cameras the overall probability P^o_j with which j is detected at least once in the entire time horizon H is given by[2]:

$$P^o_j = 1 - \prod_{t \in \mathcal{H}} \prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_{it}} Q_{ijkt} \tag{2}$$

3.3.2 Camera Configurations

Given a set of PoI (e.g. targets or locations in an area), we are interested in finding the best possible configuration k for each camera i that meets the given constraints for detection performance. To do so, firstly all possible configurations $\mathcal{K}_{it}$ for a camera with corresponding detection probabilities for each point must be found. To avoid the brute force approach, a systematic procedure for each camera i is performed that involves generating a finite number of configurations. This procedure is executed by each camera, and takes as input the position coordinates of the points to be monitored. The output of this process is a list of possible configurations for each camera with the respective miss–detection probabilities for each point. This list will be used as input to the optimization algorithm in order to find the overall best configuration for each camera in order to minimize the miss–detection probabilities. In case of moving PoI then the set of possible configurations $\mathcal{K}_{it}$ will change over time; however, when the points are static then it is only necessary to find it once. A camera can determine the corresponding detection probabilities by forming different FoVs around *anchor points* which determine unique configurations and a corresponding FoV. This procedure is repeated for each target/point in the area to generate the configurations for a specific camera.The reader is referred to [3] for more details.

3.4 Area and Path Segments

In order to model the expected number of undetected targets in the area we first segment the monitored area into a grid of cells $g \in \mathcal{G} = \{1, \cdots, N_g\}$, where N_g is the amount of cells, with cell g centered at (x^G_g, y^G_g). Furthermore, given that a target is moving with mean velocity $\overline{u}_j$ we can estimate its position in the path at future time-steps. Hence, we divide each path l into a total of N^s_l segments, such that each cell contains a number of segments for a particular path. We denote the index of the cell that each segment s of path l is located in as $g_l(s)$. We assume, without loss of generality, that the miss–detection probability of a target at time t located anywhere in cell g is equal to that of the cell center denoted as Q_{gt}. It follows that the miss–detection probability Q^s_{lst} of a target on segment s of path l at time t is equal to $Q_{g_l(s)t}$, as segment s of path l is in cell $g_l(s)$[3].

[2] Assuming that detections from multiple cameras are uncorrelated.

[3] Miss–detection probabilities can be modelled at a finer

Algorithm 1 Estimation of area activity and Cell miss-detection performance

```
 1: At $t = 0$ initialize $E_{ls0} = 0;\ \forall\ l, s$
    (1) Propagate the expected number of targets
 2: for ($l = 1 : N_l$) do
 3:     for ($s = 2 : N_l^s$) do
 4:         $E_{lst} = E_{l(s-1)t}$
 5:     end for
 6:     $E_{l1t} = \lambda_l$
 7: end for
    (2) Update cell weights
 8: for ($l = 1 : N_l$) do
 9:     for ($s = 1 : N_l^s$) do
10:         $W_{g_l(s)t} = W_{g_l(s)t} + E_{lst}$
11:     end for
12: end for
    (3) Update cell miss–detection probabilities
13: for ($g \in \mathcal{G}$) do
14:     $Q_{gt} = \prod_{c \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} (Q_{ikt})^{\zeta_{ikt}}$
15: end for
    (4) Update segment Miss–detection probabilities
16: for ($l = 1 : N_l$) do
17:     for ($s = 1 : N_l^s$) do
18:         $Q_{lst}^s = Q_{g_l(s)t}$
19:     end for
20: end for
    (5) Update the expected number of detected targets
21: for ($l = 1 : N_l$) do
22:     for ($s = 2 : N_l^s$) do
23:         $E_{lst} = E_{l(s-1)t} \times Q_{lst}^s$
24:     end for
25: end for
```

4. OPTIMIZATION AND RECONFIGURATION ALGORITHM

We utilize the modeling framework in Section 3 to develop a solution to the problem of maximizing the number of targets to be detected at least once. The solution relies on predicting the likelihood of undetected targets occurring at a specific region in the area which depends on the paths followed by the targets, the target arrival rates and the subsequent camera detection performance. Through appropriate optimization the cameras can then be reconfigured to increase the prospect of detecting those targets.

4.1 Area Activity Estimation

The expected activity within a cell is determined by the activity propagating from all paths that go through that cell. We model this via a data vector for each path that maintains parameter E_{lst} representing the expected number of undetected targets along segment s of path l at time t. This structure can be thought as a First-In-First-Out (FIFO) queue, with length equal to the number of segments in the path N_l^s, that propagates the expected number of undetected targets per time step along each path. By aggregating the contribution from different E_{lst} values associated with cell g we can find a weight for that cell, W_{gt}, which represents the expected number of undetected targets. Combining this with the miss–detection probabilities at those cells we can quantify the potential for the detection of new targets within different cells, which is used to decide on the best reconfiguration of the cameras.

Algorithm 1 shows how to update the expected number of undetected target at each cell segment E_{lst} and weight cell W_{gt}. At time *0* in *line 1* E_{lst} is initialized to zero. The first step is to simulate the propagation of undetected targets within the area by shifting the values of E_{lst} as shown in *line 4*. At the same time we initialize the first segment of each path with the arrival rate per time step for that path λ_l as shown in *line 6*. The second step is to update the cell weights by summing the contributions of each cell segment as shown in *line 10* to give the total number of estimated undetected targets per cell. After this step the optimization algorithm outlined in Sec. 4.2 is executed to reconfigure the cameras with the new cell weights. The new configurations result in updated miss–detection probabilities for each cell which are computed in *line 14*. Accordingly, in *line 18*, we also update the per segment miss–detection probabilities which correspond to the cell $g_l(s)$ that the segment s belongs to. The final step is to also update the expected number of undetected targets per segment in *line 23*. This is done by multiplying the expected number of targets in that cell with the cell overall miss–detection probability, resulting in a new expected number of undetected targets.

4.2 Optimization problem

In order to maximize the expected number of detected targets we want to direct the cameras to monitor cells with a high probability of containing undetected targets which is expressed by W_{gt}. This problem is equivalent to minimizing the weighted overall miss-detection probability for each cell. Recall from Section 3 that the overall miss-detection probability for a point of interest is equal to the product of miss-detection probability of all cameras that cover it. Hence, we formulate the following optimization problem to find a solution.

$$\min \quad \sum_{g \in \mathcal{G}} W_{gt} \prod_{t \in \mathcal{H}} \prod_{i \in \mathcal{C}} \prod_{k \in \mathcal{K}_i} Q_{gkt}^{\zeta_{ikt}} \tag{3a}$$

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

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

Notice that the objective of (3) is nonlinear and solution with standard solvers is not possible. Nonetheless, we are able to transform the problem into a mixed-integer linear program, which can be solved with standard optimization software, using an appropriate problem reformulation and subsequent piece-wise-linear approximation approach. More details can be found in [3], where the approach was first introduced for the case that $W_{gt} = 1, \forall\ g,\ t$.

5. EVALUATION RESULTS

To evaluate the proposed framework we developed a simulation environment in MATLAB which incorporates all the modelling aspects introduced in Section 3. We simulate a square area of $20 \times 20m^2$ where 12 entrance/exit points are generated. Four cameras are employed to monitor the area one at the center of each side. Different paths are formed between the points and targets enter from each point with different rates. The camera runs a pedestrian detection module

granularity within a cell.

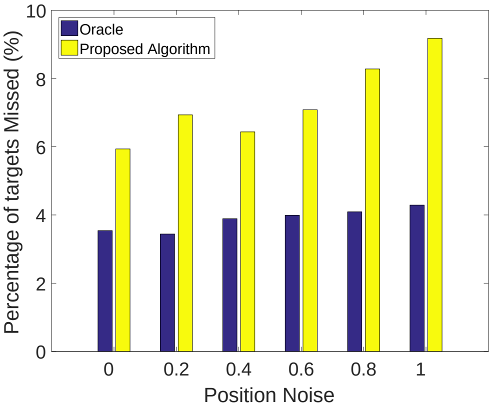

Figure 2: Percentage of missed targets versus noise in target position.

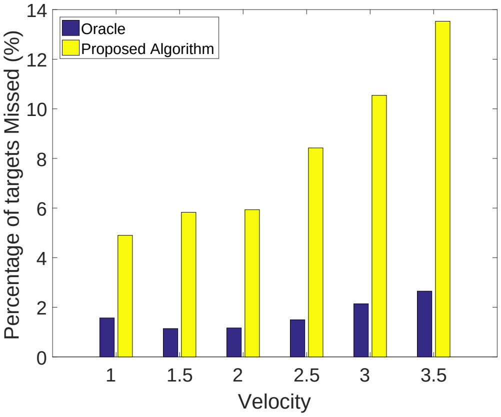

Figure 3: Percentage of missed targets versus the target velocity.

for which we use the model established in [3], which is 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.

We run a total of 200 different simulation scenarios and record the number of targets missed, and how performance is affected from various parameters such as the noise in target position, arrival rate, number of paths, and target velocity. In addition, since there are no related works that tackle the same problem we compare the performance of our approach to an ideal case scenario, labeled as *oracle*, where the target positions are known and the cameras reconfigure through optimization to improve the detection performance. Since the *oracle* knows all target locations and does not have to discover new targets as in our case it is expected to perform better. However, it serves as a benchmark which indicates how close is the developed approach to the optimal solution.

The results are illustrated in Fig. 2-6. First, Fig. 2 shows the percentage of missed targets with respect to noise on the position, or in other words the deviation of their position from their intended path. As expected the *oracle* performs better, having a value of around 3% compared to 6-9% for the proposed algorithm. However, notice that our approach maintains steady performance even if the target trajectories deviate significantly from a straight line. Fig. 3, indicates that as the velocity of targets increases the performance of our approach deteriorates; this is because as the speed of targets increases it reduces the time units that a camera views the targer and so it has less chance to detect a new target. This can be improved by adding more cameras to increase the chances of detection. On the contrary, there is only a slight deterioration on the performance of the *oracle* algorithm as the target positions are always known. In Fig. 4, we can see that as the rate of target arrivals increases, the performance of the algorithm remains steady at 5-6%. This is because through the proposed approach the cameras keep covering important hot-spots in the area and thus keep up with the increasing arrival rate. On the other hand, the "oracle" performance deteriorates in this case as the algorithm has to choose between covering different targets. Finally, in Fig. 5 we observe that as the number of paths in the area increase the performance of the *oracle* deteriorates while for the proposed approach the performance maintains some regularity at 5-6%. This is again because in the *oracle* case the cameras have to choose between targets while in our case we monitor areas with high probability of undetected targets passing through. We also perform experiments with

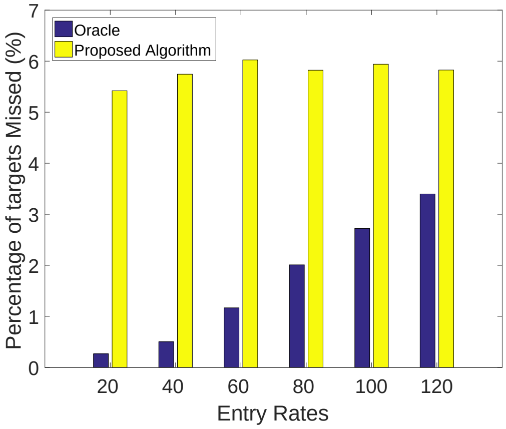

Figure 4: Percentage of missed targets versus the rate of target arrivals.

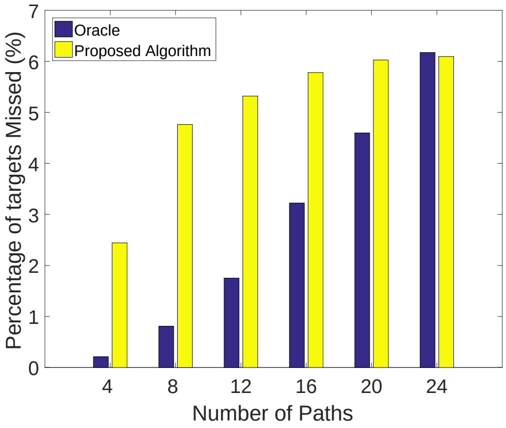

Figure 5: Percentage of missed targets versus number of paths in the monitored area.

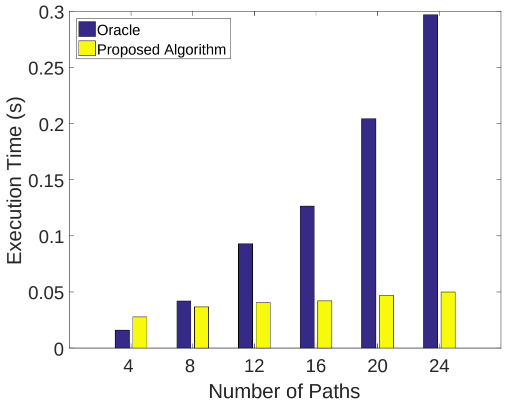

Figure 6: Execution time versus the number of paths in the monitored area.

regards to the execution time of the optimization for the two approaches with regards to the number of paths. As it is illustrated in Fig. 6, the proposed approach is almost constant at 0.05 *sec* since it depends on the number of cells which is constant, while the *oracle* execution time increases when more targets need to be considered in the optimization.

6. CONCLUDING REMARKS

This paper introduced a framework that considers stochastic parameters in the environment such as the camera detection performance and target movement to dynamically adjust the pan and tilt angles of active cameras to cover different regions, thus maximizing the expected number of unique targets detected over a time horizon. Overall, one of the main conclusions from the simulation results is that by incorporating known statistical information such as entry rates and known path trajectories can greatly improve the monitoring efficiency of SCNs. The proposed approach manages to perform close to the optimal solution and at the same time remain computationally independent from the number of targets, without having to keep a view of the whole area. Future research will utilize the actual camera detections rather than theoretical values as feedback, to improve the estimation of the location of undetected targets. Within our immediate plans is to also model more complex scenarios (target trajectories, no knowledge of arrival rates, etc.). Another direction of work is to experimentally evaluate the system under real-world conditions to gain more insight into the overall performance and implementation challenges.

7. ACKNOWLEDGMENTS

This work was supported in part by the ERC Advanced Grant "Fault-Adaptive", ERC grant agreement No 291508. The dissemination of this work is being supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE).

8. REFERENCES

[1] A Camera Uncertainty Model for Collaborative Visual Sensor Network Applications. 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, 2015. ACM.

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

[3] Christos Kyrkou, Eftychios Christoforou, Stelios Timotheou, Theocharis Theocharides, Christos Panayiotou, Marios Polycarpou. Optimizing the Detection Performance of Smart Camera Networks Through a Probabilistic Image-Based Model. *IEEE Transactions on Circuits and Systems for Video Technology*, 1, 2017.

[4] D. G. Costa, I. Silva, L. A. Guedes, P. Portugal, and F. Vasques. Enhancing redundancy in wireless visual sensor networks for target coverage. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, WebMedia '14, pages 31–38, New York, NY, USA, 2014. ACM.

[5] S. He, D. H. Shin, J. Zhang, J. Chen, and Y. Sun. Full-view area coverage in camera sensor networks: Dimension reduction and near-optimal solutions. *IEEE Transactions on Vehicular Technology*, 65(9):7448–7461, Sept 2016.

[6] K. R. Konda and N. Conci. Global and local coverage maximization in multi-camera networks by stochastic optimization. *Infocommunications Journal*, 5(1):1–8, 2013.

[7] J. Liu, S. Sridharan, and C. Fookes. Recent advances in camera planning for large area surveillance: A comprehensive review. *ACM Comput. Surv.*, 49(1):6:1–6:37, May 2016.

[8] S. M. B. Malek, M. M. Sadik, and A. Rahman. On balanced k-coverage in visual sensor networks. *J. Netw. Comput. Appl.*, 72(C):72–86, Sept. 2016.

[9] V. P. Munishwar and N. B. Abu-Ghazaleh. Scalable target coverage in smart camera networks. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*, ICDSC '10, pages 206–213, New York, NY, USA, 2010. ACM.

[10] 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.

[11] 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.

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

[13] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recogn. Lett.*, 34(1):3–19, Jan. 2013.