Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Experimental Validation of Patrolling Strategies in an Automated Surveillance Environment ⋆

Stephan M. Huck * **Nikolaos Kariotoglou** *
Michael Dahinden * **John Lygeros** *

* *Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8092, Switzerland (e-mail: {huck, karioto, lygeros}@control.ee.ethz.ch, dahmicha@ethz.ch)*

Abstract:
The Autonomous Robotic Patrolling and Surveillance environment (AuRoPaS) is a testbed at the Automatic Control Laboratory of ETH Zurich to experimentally validate tracking, observation, and monitoring strategies for security systems. The setup comprises two high performance closed-circuit television (CCTV) cameras and mobile robots to simulate different types of surveillance scenarios. We propose a velocity based model predictive control scheme for the camera movements, which allows us to generate smooth trajectories and acquire stable images from targets. Experimental results demonstrate the successful reference tracking of the camera controller. We illustrate the integration of high level algorithms into the testbed by applying two stochastic patrolling strategies. The patrolling performances are evaluated on a scenario with moving targets visiting prioritized regions.

Keywords: Cameras, Predictive Control, Autonomous Mobile Robots, Surveillance Systems, Patrolling, Implementation, Validation

1. INTRODUCTION

The scale and complexity of network systems has risen significantly during the past few decades as a result of technological advancement. This effect is clearly noticeable in the area of surveillance systems where the number of closed-circuit television (CCTV) cameras is constantly growing, especially around safety critical areas like stadiums, airports, road networks and railway stations. Using these systems, it is possible to quickly react or even predict situations where crowd safety might be compromised. However, in order to do so effectively, it is usually the job of a human operator to identify such situations and notify the relevant authorities. Due to the limitation of human operators in analyzing multiple images simultaneously, it is of interest to automate this process and have the CCTV cameras automatically detect and report threats. Towards this end we have developed a testbed to benchmark autonomous patrolling strategies where the goal is to capture a smart evader.

Researchers have been addressing the broad field of surveillance systems from quite different perspectives. Tasks like object identification and detection (see Lowe (1999), and Comaniciu et al. (2003)) or face recognition (see Li and Jain (2005)) are mainly addressed by the computer vision community. An approach based on pursuit-evasion scenarios has partially been addressed from the game theoretical side for example in Vidal et al. (2002), and Meng (2008). A survey on different methods can be found in Chung et al. (2011). In the control community some algorithms have been proposed (see for example Raimondo et al. (2010), and Avni et al. (2008)) where the goal is to exploit the dynamics of pan-tilt-zoom cameras (PTZ) to either extend the range of surveillance or improve the quality of the obtained images. Another related field of research involves the objective of patrolling areas to detect intruders, e.g. Baseggio et al. (2010) and Basilico et al. (2009).

The wide range of applications, available systems and scenarios make it difficult to evaluate and compare different approaches in a systematic way. A few self contained testbeds have been proposed so far. In order to evaluate the tracking algorithm presented in Raimondo et al. (2010), an early version of the AuRoPAS testbed was used consisting of a single camera and small scale race cars as moving targets. The target movement was not automated, thus posing limitations on the reproducibility of experiments. The authors in Salvagnini et al. (2011) utilize a single PTZ camera and a projector to emulate moving targets, achieve reproducibility of scenarios and evaluate tracking algorithms.

In this work, we present an automated testbed to analyze different kinds of surveillance tasks, with the focus on PTZ cameras, in a reproducible fashion. To the best of our knowledge we propose the first dual PTZ camera setup utilizing real autonomous mobile robots as moving targets. The major advantage is the possibility to run fully automated and reproducible long times experiments. We further propose a model based velocity controller for PTZ cameras in order to obtain stable image acquisition which is crucial in most image processing tasks. The capabilities of the testbed are demonstrated with the comparison of two patrolling strategies.

The remainder of this paper is organized as follows: the experimental setup, consisting of a surveillance system and automated mobile robots, is described in Section 2. In Section 3 we present the camera models and the predictive velocity controller. The two tested patrolling strategies are introduced in Section 4 and all experimental results are

⋆ This work was partially supported by the HYCON2 Network of Excellence and by the SNF grant number 2000211_37876

provided in Section 5. We conclude the paper by pointing out some final remarks and potential applications.

2. TEST BED ENVIRONMENT

The surveillance environment consists of a flat platform with a 3.0 m × 2.5 m area and two PTZ CCTV cameras mounted in 2 m distance to the surveillance space at a height of 2.5 m where a "Logitech HD Pro Webcam C920" is mounted, acting as a global eye. Images captured by the global eye are used as feedback for the control of the mobile robots and as a tool to verify the operation of the PTZ camera controllers and patrolling strategies.

The computational tasks are split between an Intel Core i5-760 2,8 GHz Windows PC with 8GB RAM, running all camera related programs and surveillance algorithms, and an Intel Core i5-2400 3.1 GHz Windows 7 PC with 8GB RAM used in the communication, detection and path-planning algorithms for the mobile robots. All developed algorithms are programmed in C++ with a graphical user interface developed with C# for the robot-management, e.g., insertion and removal. A schematic of the system with the software architecture is presented in Figure 1.

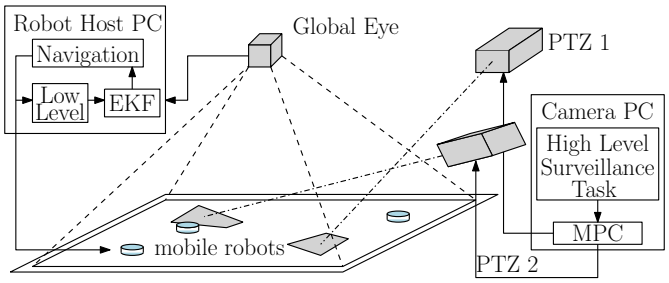

Fig. 1. Schematic of the testbed setup with global eye, PTZ cameras and mobile robots. The solid arrows depict communications links. The gray polygons on the ground plane are the projected FOVs.

2.1 E-Puck Robot

One of the main goals of the testbed is the repeatability of experiments. Towards this end we are employing several low cost differential wheel robots (e-puck model Mondada et al. (2009)) that are simple to control given position feedback. The used robot model has a diameter of 70 mm and is equipped with a variety of passive and active sensors. It can move with a maximal speed of 0.13 m/ s using two individually controlled stepper motors with a resolution of 1000 steps per wheel rotation. The motor control loop is run at a very high frequency making the input-to-velocity relationship almost linear. Using these robots, it is possible to run experiments over long time by making them follow predefined trajectories. Moreover, for the robots to follow state dependent randomized strategies, closed loop control can be used to generate their corresponding random movements.

We modeled the robot as a discrete time unicycle model of the form

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \alpha_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \\ \alpha_k \end{pmatrix} + \begin{pmatrix} v_k \cos(\alpha_k) \\ v_k \sin(\alpha_k) \\ \omega_k \end{pmatrix} T_r \qquad (1)$$

where T_r is the sampling time, $(x_k,\, y_k)$ the position on the plane and α_k the orientation at time step k. The inputs to the model v_k, ω_k are the translational and angular velocity respectively. These can be mapped to left and right wheel speeds v_L,v_R using standard relations described in, e.g., Borenstein et al. (1996)

$$v_k = \frac{v_R + v_L}{2}, \quad \omega_k = \frac{v_R - v_L}{d}, \quad v_k = \omega_k r, \qquad (2)$$

where r is the radius of the circular path followed when $\omega_k \neq 0$ and $d = 4.21$ cm is the distance between the wheels.

2.2 Detection Algorithms

The position and orientation of the robots is measured using the global eye camera and the state information is extracted from the images with a blob-detection algorithm. The algorithm uses a a two-colored black and blue scheme, where a smaller black dot is used to determine the robot heading angle. Whenever multiple robots are deployed, we use a variation of a probabilistic measurement association approach (see e.g., Bar-Shalom et al., 2009) to correctly assign robots and measurements. In addition, a standard extended Kalman filter (EKF), based on the model (1), was implemented to further improve the quality of the measurements. The process is assumed to be affected by zero-mean additive Gaussian noise. The details for the implemented version can be found in Huck et al. (2014).

2.3 Communication and Control

The control structure, as shown in Figure 1, comprises a generic high level navigation function running on the PC and low-level controls on-board the e-pucks. The navigation function is running in closed loop, with a sampling period of T_{nav}. It uses the EKF filtered measurements of the robot states and sends the estimated position along with the next reference position to the robot via Bluetooth. A trajectory planner, generating a trajectory between the actual position and the reference position, and the wheel speed controller are implemented as low level controls. Based on odometry, using model (1) and relations (2), the wheel speed controller applies the velocity commands in open loop. It can be seen from Figure 2 that the proposed approach yields small errors and the on-board open loop point tracking is sufficient for the e-puck robot. As a result, we can treat the robot as a black box and switch between different high level planning algorithms. Furthermore the open loop approach allows us to limit the communication requirements. Note that in this regard, an exact copy of the low level controls, running at a sampling time of $T_r < T_{nav}$, is implemented on the PC as well, to keep the communications at a minimum. An example of a high level navigation function is described later in Section 5.3.

3. CAMERA CONTROL

AuRoPaS uses two PTZ CCTV cameras of the type Ulisse Compact from Videotec (2013) with optical zooms of x10 and x32. The minimal zoom covers the whole testbed and the field of view (FOV) using maximal zoom approximately captures a single robot, which allows us to simulate a broad range of small and large scale surveillance scenarios from indoor and outdoor spaces. Both cameras are connected through a serial link to the surveillance PC and come with on-board velocity tracking controllers.

We use a standard pinhole camera model (e.g., Spong et al., 2005, Chap. 11), to project the camera field of view on the surveillance space. Using this projection map we can translate the camera PTZ state to the positions on the surveillance space that are covered. Controlling the position of the FOV on the surveillance space (for a fixed zoom level) is equivalent to controlling the pan and tilt angles of the motors. As a result we have developed a

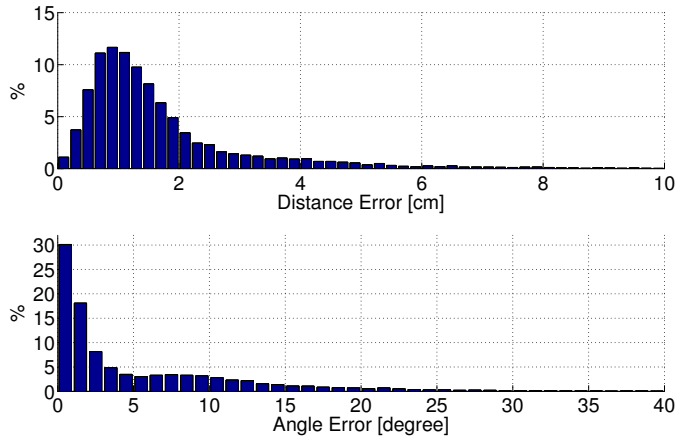

Fig. 2. Position and orientation errors (in %) between reference and measurement of an 80 min experiment with 6 robots, executing a random walk with sampling time of $T_{nav} = 3.125$ s. The trajectories generated onboard were Dubins paths. The mean distance error is 1.7 cm and the mean angular error is 5.4 degrees.

model to describe the evolution of PT angles as a function of the velocity input given to the on-board controllers. In this work we propose a predictive control scheme to achieve position reference tracking under constraints on the smoothness of the velocity profile.

3.1 MPC Trajectory Tracking

The camera motion system consists of two independent motors and a built-in controller, tracking the specified pan and tilt motor velocities. Based on identification experiments, we use a discrete linear single-integrator model

$$\boldsymbol{p}_{k+1} = \mathbf{A}\boldsymbol{p}_k + \mathbf{B}\boldsymbol{u}_k \tag{3}$$

to capture the dynamics. The state $\boldsymbol{p}_k = (\theta_k, \psi_k) \in [0, 2\pi]^2$ consists of the pan and tilt angles of the camera at time step k and the input $\boldsymbol{u}_k = (v_{\theta,k}, v_{\psi,k})$ consists of the desired pan and tilt velocities. The system and input matrices are given by $\mathbf{A} = \mathbb{I}$ and $\mathbf{B} = \mathbb{I} \cdot T_m$, where $\mathbb{I}$ is the identity matrix of appropriate dimension and T_m the sampling time of the system.

We apply a standard MPC scheme for reference tracking (see e.g., Rawlings and Mayne, 2009) and solve the optimization problem given by

$$\begin{aligned}
\min_{\boldsymbol{u}_k} \quad & \sum_{k=1}^{N} \Delta\boldsymbol{p}_{k+1}^T \mathbf{Q} \Delta\boldsymbol{p}_{k+1} + \boldsymbol{u}_k^T \mathbf{R} \boldsymbol{u}_k + \\
& + \sigma(\delta_{\theta,k} + \delta_{\psi,k}) + \rho(\delta_{\theta,k}^2 + \delta_{\psi,k}^2) \\
\text{subject to:} \quad & \boldsymbol{p}_1 = \boldsymbol{p} \\
& \boldsymbol{p}_{k+1} = \mathbf{A}\boldsymbol{p}_k + \mathbf{B}\boldsymbol{u}_k \quad k = 1, \dots, N \\
& \|\boldsymbol{u}_k - \boldsymbol{u}_{k-1}\|_2 \leq c \quad k = 1, \dots, N \\
& |\Delta\theta_k| \leq b_\theta + \delta_{\theta,k} \quad k = 1, \dots, N \\
& |\Delta\psi_k| \leq b_\psi + \delta_{\psi,k} \quad k = 1, \dots, N \\
& \delta_{\theta,k} \geq 0, \quad \delta_{\psi,k} \geq 0 \quad k = 1, \dots, N
\end{aligned} \tag{4}$$

where $\mathbf{Q} \succ 0$ and $\mathbf{R} \succ 0$ are weight matrices to penalize the deviation from the reference angles $\Delta\boldsymbol{p}_k = \boldsymbol{p}_{ref,k} - \boldsymbol{p}_k$ and the inputs respectively. A rate constraint c is imposed on the inputs to enforce a smooth velocity profile. Furthermore, we enforce box-constraints b_θ, b_ψ on the individual angles. While the rate constraints are always to be met, a linear penalty term involving the relaxations δ_θ and δ_ψ is introduced in the cost function, to take care of potential infeasibility due to the box constraints on the angles. Consider $(\lambda_{\theta,k}^*, \lambda_{\psi,k}^*) \in \mathbb{R}_+^2$ to be the Lagrange multiplier vector for the state constraints of each step. If the penalty parameter is chosen to be $\sigma \geq \| (\lambda_{\theta,k}^*, \lambda_{\psi,k}^*) \|_\infty$ for all $k = 1, \dots, N$, it is known from (Bertsekas, 1999, Chap. 5), that (4) recovers the same solution u^* of the problem without penalty term, if one exists for the latter. The quadratic penalty term with $\rho > 0$ assures strict convexity, such that the problem is well conditioned and can be directly solved via a first order method solver such as FiOrdOs (Ullmann, 2011). The parameter ρ is a design choice and can be selected, e.g., in the range of the weight matrices $\mathbf{Q}, \mathbf{R}$. The values for all used parameters are given in the results section.

4. PATROLLING STRATEGIES

Previous results (Raimondo et al., 2011) have illustrated the benefit of randomized patrolling strategies as opposed to deterministic strategies, where a smart evader can avoid detection almost surely. As a result we focus on the implementation and evaluation of stochastic patrolling strategies and two different methods to demonstrate the applicability on the testbed. The first strategy, referred to as *stochastic localization patrolling* (SLP), was presented in the earlier work of Huck et al. (2012), where knowledge about the areas to be patrolled is incorporated via an *importance map*, representing the priority or expected importance of regions. A similar concept for mobile robots on finite spaces can be found in Srivastava et al. (2009). We compared this to another algorithm, called *random line patrolling* (RLP), relying on information about the important regions of the surveillance space.

The SLP strategy consists of three phases: a specification phase, formalizing the patrolling objectives, a construction phase, distributing the objectives to the cameras and the final patrolling phase that uses a stochastic search algorithm (Algorithm 1 below), to provide set-points to the camera trajectory MPC introduced above. Let $\mathcal{X} \subset \mathbb{R}^2$ denote the patrolling area and the pair (θ, ψ) the position of the camera center in the local pan-tilt space $\mathcal{Y}_i \subset [0, 2\pi]^2$ of camera $i = 1, ..., N$. From the specification phase we assume to be given a scalar function $C : \mathcal{X} \to [0, 1]$ called the importance map. This is translated during the second phase into probability distributions ξ_i on the local spaces $\mathcal{Y}_i$. The distributions ξ_i indicate how often the camera i should be at each angle configuration according to the desired coverage of the corresponding part of $\mathcal{X}$. For the construction of the distributions ξ_i see Huck et al. (2012), where two methods are proposed such that the coverage of the global space resembles the given importance map. In the third phase, a discrete time Markov Chain determines the target position for the camera given the current angle measurement. The transition probability of the chain depends on the distribution ξ_i, such that the camera FOV moves with constant velocity on straight lines through the regions of low importance and changes direction whenever the importance of the seen area is high. Results from recent work (Huck and Lygeros, 2013) show that the applied Markov chain is guaranteed to converge to a distribution which qualitatively reflects the given distributions ξ_i.

The RLP strategy is also based on the knowledge of important areas. In essence, we directly utilize the importance map C constructed above, to determine movements of the camera between regions of high priority in the patrolling space $\mathcal{X}$. The camera moves deterministically between

Algorithm 1 Stochastic Localization Patrolling (SLP)

Require: Initial position p_0 and camera heading angle ϕ_0, distribution ξ, velocity $\bar{v}$

for $k = 0 \rightarrow N_{ref}$ **do**

update $p_{k+1} = p_k + \bar{v} \begin{pmatrix} \cos\phi_k \\ \sin\phi_k \end{pmatrix} T_p$

generate proposal angle $\tilde{\phi}_{k+1}$

calculate the acceptance probability $\alpha(\xi(\boldsymbol{p}_k), \phi_k)$

update: $\theta_{k+1} = \begin{cases} \tilde{\theta}_{k+1} & \text{w. p. } \alpha(\xi(\boldsymbol{p}_k), \phi_k) \\ \theta_k & \text{w. p. } 1 - \alpha(\xi(\boldsymbol{p}_k), \phi_k) \end{cases}$

set $\boldsymbol{p}_{ref}(k) = \boldsymbol{p}_k$

set $k = k + 1$

end for

return $\boldsymbol{p}_{ref}$

important areas but chooses randomly which area to visit next. Algorithm 2 describes the steps of applying the RLP method on the AuRoPas testbed when generating the reference trajectory online. We first create a list of the centers of important regions according to their assigned priorities and normalize them. Second, the resulting distribution is used to generate a trajectory of target positions $\{\boldsymbol{p}_{t,m=1,\dots,N_{ref}}\}$ via inverse transform sampling (e.g. Devroye, 1986). In contrast to the SLP strategy, this method switches to the next target position only if the camera is sufficiently close to the current set-point. We implemented a rolling horizon-type approach by appending set-points to the current trajectory. Consider at time k the N-step solution $(\boldsymbol{u_i}^*)_{|k}$ and the associated trajectory $(\boldsymbol{p_i}^*)_{|k}$ to problem (4) for given target positions $\boldsymbol{p}_{ref,i} = \boldsymbol{p}_{t,m}$, where $i = k, ..., k+N$. Similarly to the state constraints of the MPC formulation, we consider a state of the solution to be sufficiently close, if the conditions $|\Delta\theta_i^*| \leq b_\theta$ and $|\Delta\psi_i^*| \leq b_\psi$ hold. Assume the condition is fulfilled for state $\boldsymbol{p}_j$ for some $j \in \{k, ..., k+N\}$, then the prediction horizon is shifted and all reference states $\boldsymbol{p}_{ref,j}, \dots, \boldsymbol{p}_{ref,k+N}$ are set to the next target position $\boldsymbol{p}_{t,m+1}$. This set point change is illustrated in Figure 3.

Algorithm 2 Random Line Patrolling (RLP)

Require: Initial state $\boldsymbol{p}_0 = (\theta_0, \psi_0)$, initial target positions $\boldsymbol{p}_{t,0}$, cumulative distribution F over prioritized positions

set $k = 0$, $m = 0$

set $\boldsymbol{p}_{ref}(i) = \boldsymbol{p}_{t,m} \quad \forall i = 1, ..., N$

loop

solve problem (4) $\rightarrow (\boldsymbol{p_i}^*, \boldsymbol{u_i}^*)_{|k}$

if for any j: $|\Delta\theta_j^*| \leq b_\theta$ and $|\Delta\psi_j^*| \leq b_\psi$ **then**

set $m = m + 1$

sample $\boldsymbol{p_{t,m}} = F^{-1}(U)$, U uniform on $[0, 1]$

end if

shift reference vector $\boldsymbol{p}_{ref}(i) \leftarrow \boldsymbol{p}_{ref}(i+1)$ for $i = 1, ..., N-1$

set $(\boldsymbol{p}_{ref}(j), \dots, \boldsymbol{p}_{ref}(N)) \leftarrow \boldsymbol{p}_{t,m}$

apply $\boldsymbol{u}_{1|k}^*$ to the camera

set $k = k + 1$

end loop

Note that the presented patrolling strategies can determine the next target position for the cameras online. However, for reasons of computational efficiency we generated a list of reference positions in advance in both cases. In contrast to a deterministic strategy, a smart evader with full knowledge of the importance map and the applied strategies can observe the movement of the camera and anticipate a possible next target but will never know the actual realization of the followed path.

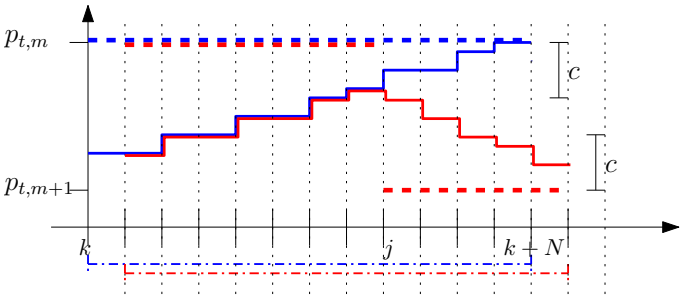

Fig. 3. Camera trajectory (solid lines) for two MPC iterations (blue, red), with dynamic augmentation of the references (dashed lines). In the first iteration, the state at time step j ends up in the specified distance to $\boldsymbol{p_{t,m}}$, such that the reference states $\boldsymbol{p}_{ref}(j), \dots, \boldsymbol{p}_{ref}(N)$ are set to the new setpoint $\boldsymbol{p}_{t,m+1}$ for the second iteration. The dashed-dotted lines at the bottom illustrate the corresponding time horizons.

5. RESULTS

5.1 Camera Identification & Controller Settings

The intrinsic and extrinsic parameters of the cameras were estimated according to Raimondo et al. (2010); Table 1 shows the most relevant, with camera height H, offset D with respect to the pan rotational axis and spatial offsets x_{off}, y_{off}, z_{off} of the optical center to the tilt coordinate system. For this work, we focus on the pan-tilt control and assume a constant field of view. To this end, the given focal length λ corresponds to fixed zoom levels of the cameras, which were chosen as a trade-off between FOV size and image resolution, over the whole pan-tilt range.

	H	D	x_{off}	y_{off}	z_{off}	λ
Camera 1	2.5	0.124	0.1485	-0.0275	0.02	0.0297
Camera 2	2.5	0.125	-0.01	0.0437	0.0917	0.0448

Table 1. Parameters of the deployed Ulisse Compact cameras.

The sampling time used in (3) was conservatively chosen to be $T_p = 0.25\,\text{s}$, based on the limits of the serial link to the cameras. We found a period of about 160 ms to result in a conflict free communication of read-out requests and control commands. On average, 11 ms were required to solve problem (4) and about 70 ms for the read out of the angle encoders.

The weight matrices for the trajectory controller were chosen to be

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.8 \end{bmatrix},$$

the penalty parameter chosen as $\sigma = 500, \rho = 1$ and the constraint limits were set to

$$b_\theta = 0.5 \quad b_\psi = 0.3, \quad c = 0.3/T_m.$$

Note, that c is a constraint on the discrete changes of the inputs and is therefore depending on the sampling time. Based on qualitative studies the velocity constraints were chosen to be $1°/\,\text{s}$ to ensure smooth image acquisition over the whole range of camera movements.

5.2 Tracking Performance

To asses the tracking performance of the model predictive controller, independent of the patrolling strategies, we tested several trajectories defined on the global space $\mathcal{X}$. Figure 4 shows the movement calculated by the MPC and the tracking achieved on the actual setup for the curved reference path. To carry out this experiment a line was drawn onto the global space, recorded with the global eye camera and imported into the simulation.

The achieved trajectory was assessed by first mounting a laser pointer onto the camera and adjusting it to the center of the camera image and then recording the movement (green circled path) of the laser point with the global eye camera. The main source of errors is the quantization of the mapping $\mathcal{X} \to \mathcal{Y}$ given by the pinhole camera projection model. Note that the quantization is conceptually not necessary, but allows us to use a look-up table for the FOV, which has clear computational advantages.

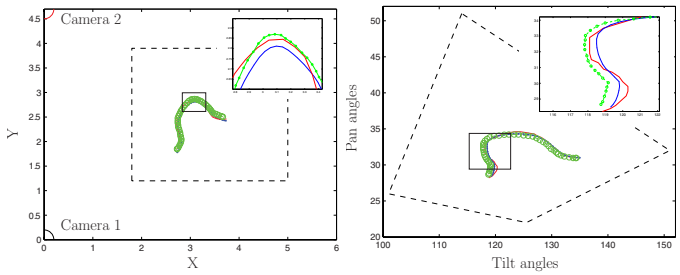

Fig. 4. Tracking results of the MPC controller. The red connected line indicates the reference trajectory on the relevant space. The blue dotted line indicates the trajectory achieved by the model predictive controller and the green circled line the actual movement of the image center. *Left:* Results recorded on the surveillance space for camera 2. *Right:* Results recorded on the pan-tilt space for camera 2.

The maximum and mean 2-norm errors of the tracking controller are reported in Table 2 for the camera angle readouts (subscript r) in the cameras pan-tilt space and the laser measurements from the world coordinate frame.

error \ path	straight	curved	edged
mean_r	0.57	0.54	0.55
max_r	0.66	0.72	0.64
mean	5.6	7.2	6.6
max	7.4	12.9	9.1

Table 2. Tracking errors for a straight line, a curved path and an edged path. The top rows show the pan-tilt errors in degrees and lower rows the errors on the patrolling space in cm.

5.3 Patrolling Scenario

To evaluate the performance of a patrolling strategy, we have assumed the existence of an object identification algorithm that determines whether a target or evader has appeared on the FOV. We employ again a standard OpenCV based blob detection algorithm directly on the images acquired by the pan-tilt cameras. We consider detections all situations where a large enough blue object is observed on the image.

We consider the following scenario to compare the performance of the two patrolling strategies described in Section 4. The targets move through the surveillance space between positions of interest in a random order according to an assigned likelihood. Using this list, we generate the importance map for the cameras representing the priorities expected by the system designer which (in general) differ in shape and value from the truth, as illustrated in Figure 5.

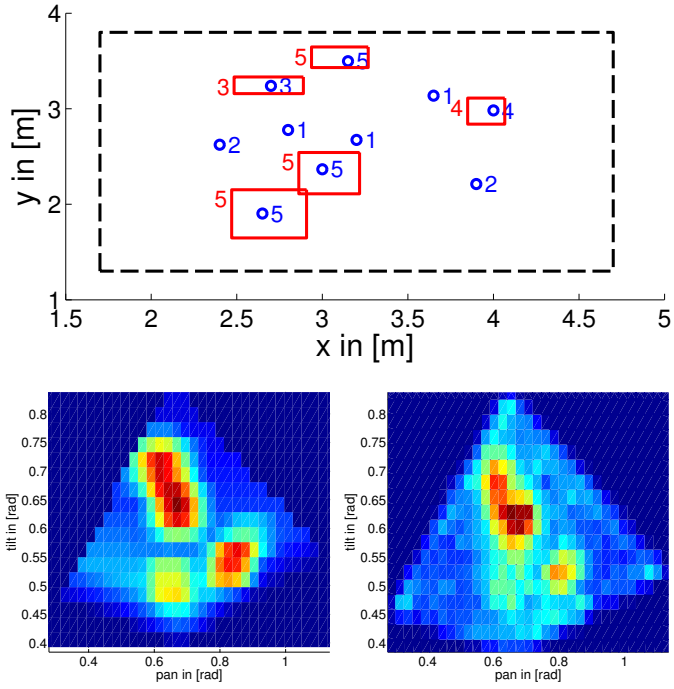

Fig. 5. *Top:* Waypoints of the robots on $\mathcal{X}$ (dashed black square) shown as circles with blue numbers indicating their importance to the robot. The designed importance map is depicted by red squares with red numbers indicating the expected importance. The RLP will move between the centers of the squares. *Bottom left:* SLP guidance distribution ξ_2 for camera 2 where red color indicates high probability and dark blue the non-accessible part of the space. *Bottom right:* Actual distribution of pan-tilt positions obtained after 30k steps of the SLP.

The patrolling strategies were run for 15 min, while the detections of the robot were counted online. For the SLP the same design choices were used as reported in Huck et al. (2012). Table 3 summarizes the performance results of both strategies and an additional static scenario, where each camera was fixed to one important location. A demonstration video of the experiments with RLP and SLP strategies can be found on the web (http://www.youtube.com/usr/ETHZurichIfA).

Strategy	RLP	SLP	Static
Camera 1 average	59.3	21.7	17.3
Camera 1 maximum	65	27	18
Camera 2 average	67.7	20.3	13
Camera 2 maximum	71	24	15

Table 3. Average number of detections for RLP and SLP as well as for the cameras kept on the topmost and the lowermost position with importance 5 from Figure 5. All three experiments were repeated three times for 900 s.

In the particular patrolling scenario considered we expected the RLP strategy to perform better since it mimics the movement of the evader between important regions. However, knowing the RLP strategy, it is straightforward

to devise an evading strategy that will avoid detection. In contrast, the SLP strategy deviates from the important regions and randomly scans the whole area making it more difficult for a smart evader to visit all important regions while staying undetected.

6. CONCLUSION

We presented a dual camera testbed with the capabilities to simulate a wide range of surveillance tasks. One of the key characteristics of the setup is that all processes are automated in order to obtain accurate statistical data to evaluate the performance of randomized algorithms. The proposed framework and test bed can be extended to multiple cameras in a straightforward way. In order to ease image processing tasks we suggested a model based approach for the control of pan-tilt cameras that achieves smooth image acquisition. The controller was evaluated experimentally and demonstrated excellent performance in tracking pan-tilt positions while satisfying predefined motor acceleration constraints. Part of our future work will be directed towards including the zoom variable in the presented optimization problem.

We used the proposed testbed to evaluate the performance of two randomized patrolling algorithms against an evader utilizing the same type of information about the surveillance space. Our next step will be to design smart evaders assuming they can observe the camera position at every decision step to design feedback evading strategies. Finally, we are interested in the design of cooperative patrolling strategies where the cameras exchange and exploit information about their current positions when making a move in the pan-tilt space.

REFERENCES

Avni, O., Borrelli, F., Katzir, G., Rivlin, E., and Rotstein, H. (2008). Scanning and tracking with independent cameras - a biologically motivated approach based on model predictive control. *Autonomous Robots*, 24(3), 285–302. doi:10.1007/s10514-007-9057-4.

Bar-Shalom, Y., Daum, F., and Huang, J. (2009). The probabilistic data association filter. *IEEE Control Systems*, 29(6), 82–100. doi:10.1109/MCS.2009.934469.

Baseggio, M., Cenedese, A., Merlo, P., Pozzi, M., and Schenato, L. (2010). Distributed perimeter patrolling and tracking for camera networks. In *2010 49th IEEE Conference on Decision and Control (CDC)*, 2093–2098. doi:10.1109/CDC.2010.5717883.

Basilico, N., Gatti, N., and Amigoni, F. (2009). Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, 5764. Richland, SC.

Bertsekas, D.P. (1999). *Nonlinear programming*. Athena Scientific.

Borenstein, J., Everett, H.R., and Feng, L. (1996). *Navigating mobile robots: systems and techniques*. A K Peters.

Chung, T.H., Hollinger, G.A., and Isler, V. (2011). Search and pursuit-evasion in mobile robotics. *Autonomous Robots*, 31(4), 299–316. doi:10.1007/s10514-011-9241-4.

Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), 564–577. doi:10.1109/TPAMI.2003.1195991.

Devroye, L. (1986). *Non-uniform random variate generation*. Springer-Verlag.

Huck, S.M., Kariotoglou, N., Summers, S., Raimondo, D.M., and Lygeros, J. (2012). Design of importance-map based randomized patrolling strategies. In *Complexity in Engineering (COMPENG), 2012*, 1 –6. doi:10.1109/CompEng.2012.6242945.

Huck, S.M., Kariotoglou, N., Dahinden, M., and Lygeros, J. (2014). Experimental validation of patrolling strategies in an automated surveillance environment. Technical report, Automatic Control Laboratory, ETH Zurich, Switzerland. URL `http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=4710`.

Huck, S.M. and Lygeros, J. (2013). Stochastic localization of sources with convergence guarantees. In *Proceedings of 12th European Control Conference*, 6. Zurich, Switzerland.

Li, S.Z. and Jain, A.K. (2005). *Handbook of Face Recognition*. Springer.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999*, volume 2, 1150–1157 vol.2. doi:10.1109/ICCV.1999.790410.

Meng, Y. (2008). Multi-robot searching using game- theory based approach. *International Journal of Advanced Robotic Systems*, 1. doi:10.5772/6232.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.c., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *In Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 5965.

Raimondo, D.M., Gasparella, S., Sturzenegger, D., and Lygeros, J. (2010). A tracking algorithm for PTZ cameras. In *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'10)*, 61–66. doi:10.3182/20100913-2-FR-4014.00060.

Raimondo, D., Kariotoglou, N., Summers, S., and Lygeros, J. (2011). Probabilistic certification of pan-tilt-zoom camera surveillance systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2064–2069. doi:10.1109/CDC.2011.6161534.

Rawlings, J. and Mayne, D. (2009). *Model predictive control: theory and design*. Nob Hill Pub., Madison, Wis.

Salvagnini, P., Cristani, M., Bue, A.D., and Murino, V. (2011). An experimental framework for evaluating PTZ tracking algorithms. In J.L. Crowley, B.A. Draper, and M. Thonnat (eds.), *Computer Vision Systems*, number 6962 in Lecture Notes in Computer Science, 81–90. Springer Berlin Heidelberg.

Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2005). *Robot Modeling and Control*. Wiley.

Srivastava, K., Stipanovic, D., and Spong, M. (2009). On a stochastic robotic surveillance problem. In *Proceedings of the 48th IEEE Conference on Decision and Control 2009. CDC/CCC 2009.*, 8567–8574. doi:10.1109/CDC.2009.5400569.

Ullmann, F. (2011). *FiOrdOs: A Matlab Toolbox for C-Code Generation for First Order Methods*. Master thesis, Automatic Control Laboratory, ETH Zurich, Zurich, Switzerland.

Vidal, R., Shakernia, O., Kim, H., Shim, D., and Sastry, S. (2002). Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18(5), 662–669. doi:10.1109/TRA.2002.804040.

Videotec (2013). Ulisse compact [online]. URL `http://www.videotec.com.html`.