A tracking algorithm for PTZ cameras

Davide M. Raimondo[1], Simone Gasparella[2]
David Sturzenegger[1], John Lygeros[1], Manfred Morari[1]
[1]Automatic Control Laboratory ETH Zurich, Switzerland
[2]Videotec S.p.A., Italy

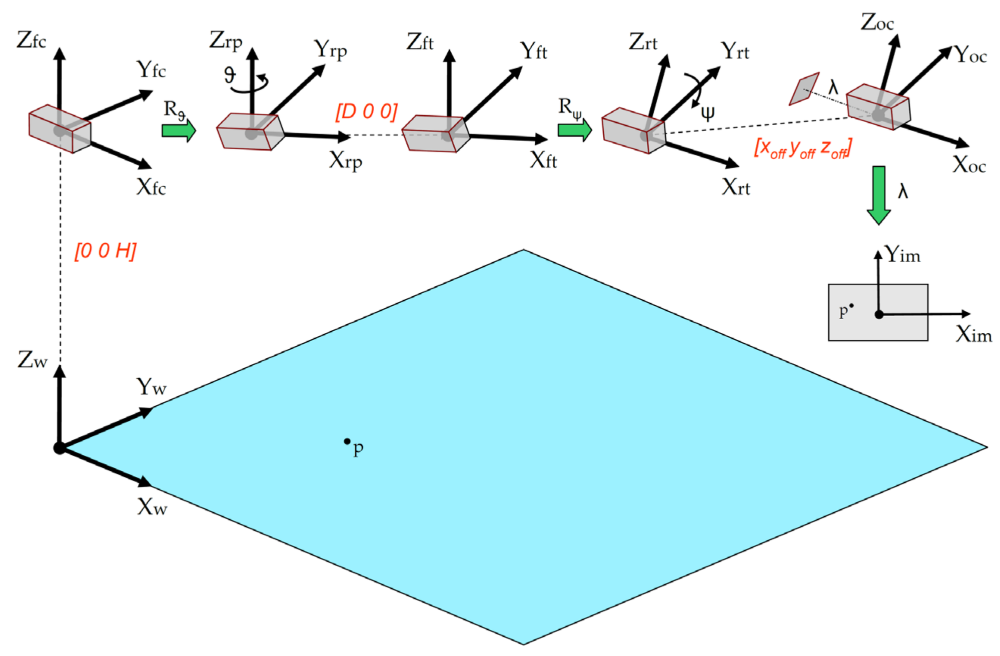
FeedNetback co-design for networked control systems

ABSTRACT

This paper presents a tracking algorithm for PTZ (pan-tilt-zoom) cameras. The tracked objects are Mini 1:43 scale RC cars that have been described by a unicycle model. The algorithm is based on the combination of EKF (Extended Kalman Filter) and PF (Particle Filters). A scanning procedure is used to explore the environment. Once targets are detected, EKF is used to predict their future position. PTZ are then moved in order to guarantee a certain probability of targets detection at the next time instant. If a target is lost, particle filters is exploited. If target is found again EKF is restored. If this does not happen in a predefined number of steps, the scanning procedure restarts.

CAMERAS

In collaboration with Videotec S.p.A., ETH has set up in its laboratory a test bed comprising pan-tilt-zoom Ulisse compact cameras,. The testbed is used to design an automated surveillance system able to detect and track targets as they move through the monitored site.



Cameras dynamics are considered constrained as follows:

$$\begin{aligned} \theta_{min} &\le \theta \le \theta_{max}, & |\theta(k+1)-\theta(k)| &\le \Delta\theta \\ \psi_{min} &\le \psi \le \psi_{max}, & |\psi(k+1)-\psi(k)| &\le \Delta\psi \\ \zeta_{min} &\le \zeta \le \zeta_{max}, & |\zeta(k+1)-\zeta(k)| &\le \Delta\zeta \end{aligned}$$

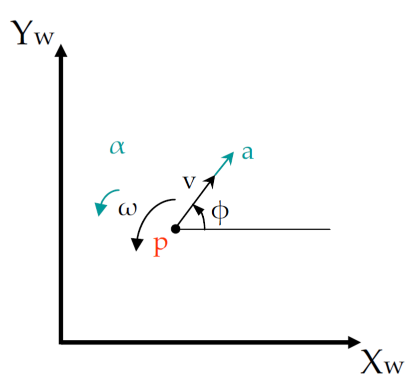
TARGETS

In our experiments so far Mini 1:43 scale RC cars Kyosho dNano have been considered as targets. Tracking these race cars can be quite challenging since they can achieve speeds of up to 5m/s. Target measurements are 3D position and orientation.

Unicycle model

Targets dynamics have been modeled as follows

$$\begin{cases} x(k+1) &= x(k) + \cos(\phi(k))v(k)\Delta T \\ y(k+1) &= y(k) + \sin(\phi(k))v(k)\Delta T \\ \phi(k+1) &= \phi(k) + \omega(k)\Delta T \\ v(k+1) &= v(k) + a(k)\Delta T \\ \omega(k+1) &= \omega(k) + \alpha(k)\Delta T \end{cases}$$



The accelerations, inputs of the model, are unknown but bounded

$$|a| \le a_{max} \qquad |\alpha| \le \alpha_{max}$$

Targets velocities are also bounded

$$|v| \le v_{max} \qquad |\omega| \le \omega_{max}$$

Such bounds are available from cars manufacturer.

Stochastic model

Since accelerations are unknown, they can be modeled as process noise. Moreover, measurements are also corrupted by noise.
If we denote with **x = [x y ϕ v ω]** the state vector and with **h(x) = [x y ϕ]** the observation function, we can write the following stochastic model

$$\begin{cases} \mathbf{x}(k+1) &= f(\mathbf{x}(k)) + \mathbf{w}(k) \\ \mathbf{z}(k) &= h(\mathbf{x}(k)) + \mathbf{v}(k) \end{cases} \qquad \mathbf{w}(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ a(k)\Delta T \\ \alpha(k)\Delta T \end{bmatrix} \qquad \mathbf{v}(k) = \begin{bmatrix} v_x(k) \\ v_y(k) \\ v_\phi(k) \end{bmatrix}$$

ACKNOWLEDGEMENTS

The project was supported by the European Commission under the Project *Feednetback* FP7-ICT-223866 (www.feednetback.eu).

TRACKING ALGORITHM

Extended Kalman Filter

Both process noise and observation noise are supposed to be gaussian.

$$a \sim \mathcal{N}(0,\sigma_1^2) \qquad \alpha \sim \mathcal{N}(0,\sigma_2^2)$$

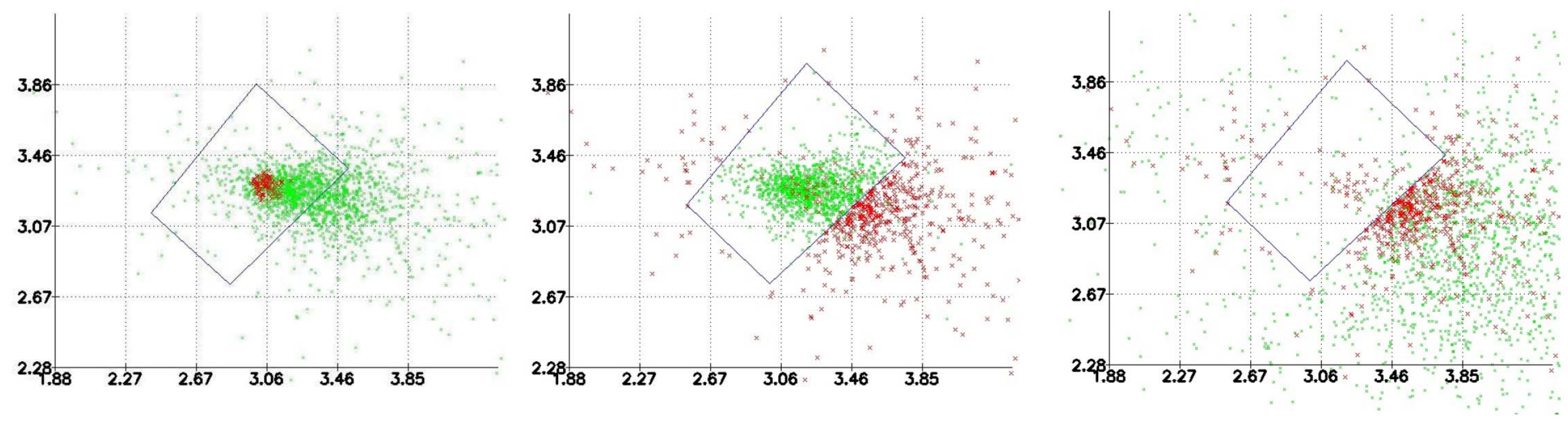
$$v_x \sim \mathcal{N}(0,\sigma_x^2) \qquad v_y \sim \mathcal{N}(0,\sigma_y^2) \qquad v_\phi \sim \mathcal{N}(0,\sigma_\phi^2)$$

Note that, observation noises are related to cameras accuracy, i.e. it is reasonable to assume they are gaussian. The initial state x(0) is assumed to follow a known Gaussian distribution x(0) ~ $N(\hat{\mathbf{x}}(0), P(0))$

Particle Filters

When a target is lost, EKF in open loop prediction could provide poor performances. For this reason, in that case, PF is exploited instead.

Sequential Importance Resampling for PTZ Cameras



(a) (b) (c)

SIR algorithm: example. Figure (a) shows in red samples of the probability distribution $p(\mathbf{x}(k-1))$ and in green the N particles $\mathbf{x}^j(k)$, $j=1,\ldots,N$. Figure (b) shows time k+1. The target was not detected In green we have again the N particles $\mathbf{x}^j(k)$ while in red the resampled N particles $\tilde{\mathbf{x}}^j(k)$. Figure (c) shows in red $\tilde{\mathbf{x}}^j(k)$ and in green $\mathbf{x}^j(k+1)$

OPTIMIZATION PROBLEM

$$\min_{\theta_i(k+1),\psi_i(k+1),\zeta_i(k+1)} \sum_{i=1}^{n} \kappa_1\Delta\theta_i^2 + \kappa_2\Delta\psi_i^2 + \kappa_3\Delta\zeta_i^2 + \frac{\kappa_4}{\zeta_i(k+1)^2}$$

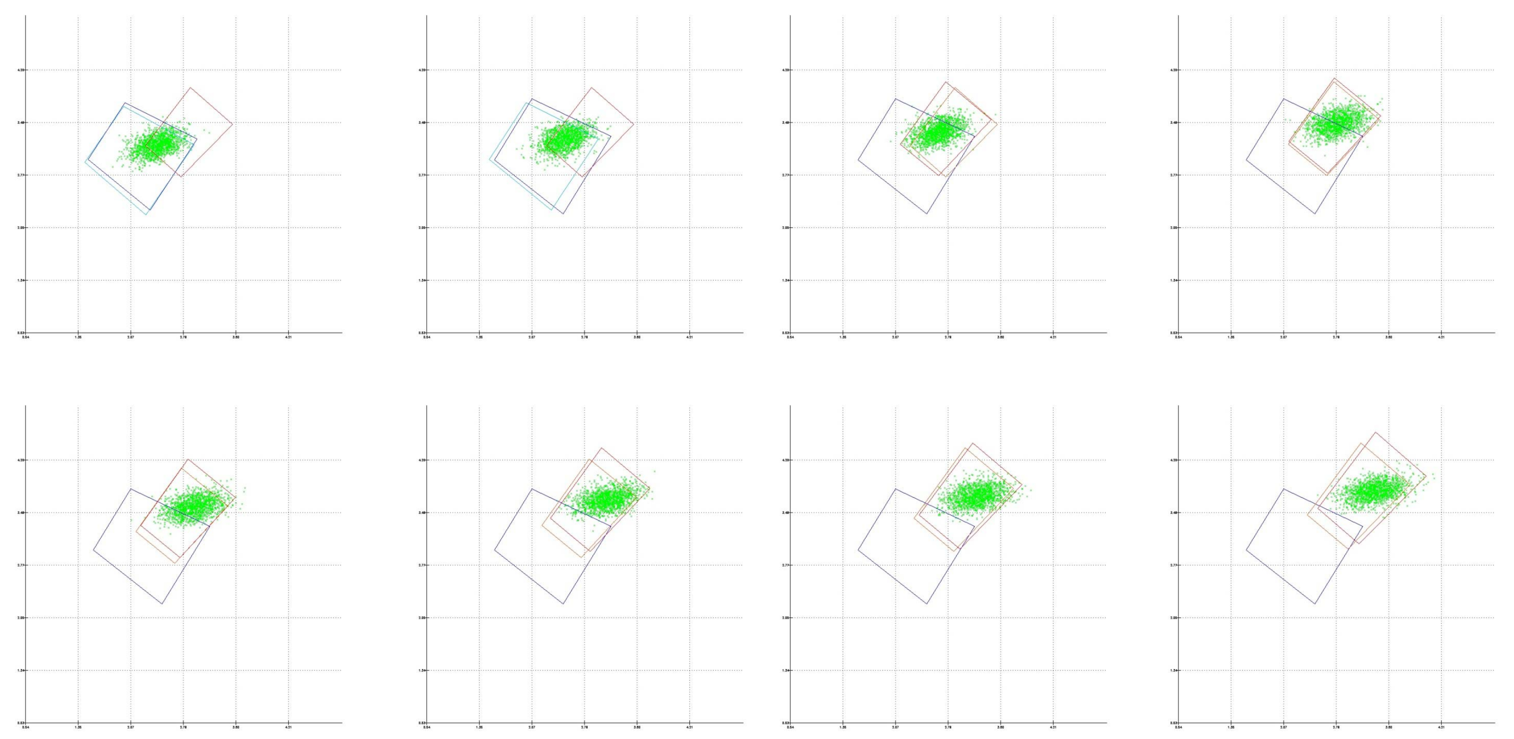
subject to cameras dynamics (2)
$P(target_j \in FOV_{\cup}(k+1)) >= \alpha$
$\forall j = 1,\ldots,m$

The optimization is performed by using Simulated Annealing (SA), see e.g. [1], with time limit equal to the sampling time.

Parallel Simulated annealing

One of the major drawbacks of simulated annealing is its very slow convergence, especially for problems with large search space. Under the assumption that each camera has a dedicated processor on board and communication between cameras is available, a parallel versions of SA can be implemented. Given *n*, number of cameras and *a* number of cameras we want to move

- start the procedure with $a = 1$
- solve in parallel all possible combinations $\binom{n}{a}$ for a fixed number of iterations
- initialize all $\binom{n}{a}$ SA optimization problems with the minimum cost feasible solution of $a-1$
- iterate the procedure till time limit is exceeded.



REFERENCES

[1] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies", Journal of Statistical Physics, vol. 34, no. 5, pp. 975-986, 1984.