In 20th Portuguese Conference on Pattern Recognition (RecPad 2014)
Covilhã, Portugal, October 2014

Multi-Target Tracking Based in Meanshift and Particle Filters

Pedro Silva, Ricardo Ferreira, José Gaspar
http://www.isr.ist.utl.pt

Institute for Systems and Robotics / IST
Lisbon, Portugal

Abstract

In this work we propose a hybrid particle filter and meanshift tracker to follow multiple targets using one camera. The number of targets is assumed to be known but allowed to vary in time.

1 Introduction

The kernel based-tracking algorithm was first proposed in [1], and despite its simplicity it proved to give good results even in the presence of motion blur and clutter. However it had the drawback of assuming that the target did not move outside of the region of interest between consecutive frames. Years later, Maggio et al. [2] proposes an hybrid target tracking algorithm that by using a particle filter and the kernel based tracking algorithm was able to increase the robustness to the mentioned problem. Still it was designed for single target tracking. The algorithm proposed in this paper, therefore intends to improve the work of [2] by addressing the multi-target tracking problem.

2 Camera Model

We use the pinhole camera model [3] to derive the perspective transformation of the points on a world referential frame to the ones in the camera image plane,

$$m \approx PM = K\left[{}^{C}R_{W} \mid {}^{C}t_{W}\right] M, \tag{1}$$

where M is a vector containing the coordinates of the point in the world referential frame, m are the pixel coordinates, P is the projection matrix, K and ${}^{C}R_{W}$ are the intrinsic and extrinsic parameters matrices respectively and finally ${}^{C}t_{W}$ is the translation vector. Given the projection matrix and a 3D plane an image pixel m can also be back-projected to a point $M = [x\ y\ z\ 1]^{\mathsf{T}}$ in the plane.

3 Target Description

The state of each target to track is described, at the time step t, by the vector $\mathbf{x}_t = (x_t, y_t, \theta_t, v_t)$, where x_t, y_t are the target spatial coordinates in the world referential frame and θ_t, v_t the target orientation and linear velocity respectively. The targets are assumed to be moving in a plane, where their z coordinate is constant and computed at the moment they are detected. Furthermore the number of targets being tracked, at a given time step, is assumed to be known.

In order to make predictions of a target next state the kinematic model $\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t$ is used with

$$f(\mathbf{x}_t) = \begin{cases} x_t = x_{t-1} + T v_{t-1}\cos(\theta_{t-1}) \\ y_t = y_{t-1} + T v_{t-1}\sin(\theta_{t-1}) \\ \theta_{t-1} = \theta_{t-1} \\ v_t = v_{t-1} \end{cases}, \tag{2}$$

T is the time step value and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, Q)$ an additive noise component drawn from the multivariate Gaussian distribution $\mathcal{N}$, with mean equal to the null vector $\mathbf{0}$ and covariance matrix Q. The function $f(\mathbf{x}_t)$ is derived from the car model used in [4], by setting to zero the wheel orientation angle ϕ. In addition to the kinematic model a target has an appearance model, under the form of a RGB color histogram, which allows to distinguish the target in the image data. This model is used to produce estimations of the targets position and orientation, by searching the image data. It is computed and assigned when the target is first detected.

4 Optimal Proposal Particle Filter

The particle filter is a Monte Carlo [5] based tracking filter used to recursively estimate the posterior probability density function $p(\mathbf{x}_t|\mathbf{z}_{1:t})$, where $\mathbf{z}_{1:t}$ are the set of observation vectors $\mathbf{z}$ up to time t. In the case of this tracking filter the measurement and kinematic models do not need to be linear, and the noise can be non additive and non-Gaussian.

The particle filter algorithm starts by drawing a set of N samples $\mathbf{x}_t^i$, $i = 1, \ldots, N$ from an importance distribution $\mathbf{x}_t^i \sim q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t)$. To each of these samples a weight ω_t^i is assigned through the formula:

$$\omega_t^i = C\omega_{t-1}^i \frac{p(\mathbf{z}_t|\mathbf{x}_t^i)\, p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t)}, \tag{3}$$

where $p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$ denotes the prior distribution, $p(\mathbf{z}_t|\mathbf{x}_t^i)$ is the likelihood and C a normalization constant to make the sum of the weights equal to 1. These weights can then be used to make the following discrete approximation of the posterior distribution:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \sum_{i=1}^{N} \omega_t^i \delta(\mathbf{x}_t - \mathbf{x}_t^i) \tag{4}$$

One commonly used estimator of the state at time t is

$$\mathbb{E}(p(\mathbf{x}_t|\mathbf{z}_{1:t})) \approx \frac{1}{N} \sum_{i=1}^{N} \omega_t^i \mathbf{x}_t^i, \tag{5}$$

where $\mathbb{E}$ denotes the expectation operator. Resampling, an additional step usually included in particle filters, replaces the actual samples by a set of new ones drawn from the discrete approximation of the posterior, hence propagating to the next iteration a set of samples that more likely represent the true system state.

In one of the simplest particle filter implementations, commonly called the bootstrap particle filter, the importance distribution is chosen to be the prior distribution: $q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t) = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$. This simplifies expression (3) making $\omega_t^i = Cp(\mathbf{z}_t|\mathbf{x}_t^i)$. However a common issue with this approach is a phenomenon typically known as degeneracy, that is characterized by the tendency for a few set of particles to have weights much larger than the rest. This leads to a poor approximation of the posterior distribution. One possible solution to overcome this is to choose a better importance sampling. It has been proven [5] that the choice $q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t) = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{z}_t)$ is optimal in the sense of minimizing the variance of the particles weights. The great disadvantage of this choice is that it is usually very difficult to evaluate. However for systems with models of the form

$$\begin{array}{c} \mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t,\ \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, Q) \\ \mathbf{z}_t = H\mathbf{x}_t + v_t,\ v_t \sim \mathcal{N}(\mathbf{0}, R) \end{array}, \tag{6}$$

with R being the measurement covariance matrix and H the observation matrix, it is possible to derive closed form expressions for $p(\mathbf{z}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$ which, as deduced in [5], are:

$$p(\mathbf{z}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{b}_t, S) \tag{7}$$

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t) = \mathcal{N}(\mathbf{a}_t, \Sigma) \tag{8}$$

$$\mathbf{b}_t = Hf(\mathbf{x}_{t-1}) \tag{9}$$

$$S = HQH^{\mathsf{T}} + R \tag{10}$$

$$\mathbf{a} = f(\mathbf{x}_{t-1}) + \Sigma H^{\mathsf{T}} R^{-1}(\mathbf{z}_t - b_t) \tag{11}$$

$$\Sigma = Q - QH^{\mathsf{T}} S^{-1} HQ \tag{12}$$

5 Multi-target Hybrid Particle Filter

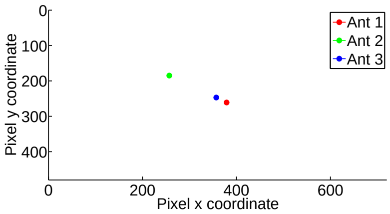
The first step in the tracking algorithm proposed is similar to the standard *prior* update of a bootstrap particle filter where the kinematic model is used to update the particles state. The difference lies in the fact that the noise is not added, with only the function $f(\cdot)$ being employed.

Then a procedure inspired by the work of Maggio et al. [2] is applied to each target, in order to get observations for its position. It starts by selecting a smaller number of particles, picked at random from the target particle set with the probability of each being drawn equal to its respective weight. If there are any duplicates they are removed in order to only have

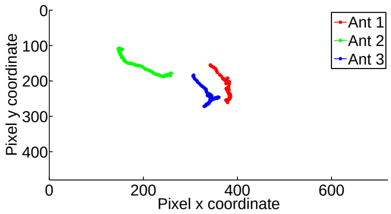
(a) Frame 1

(b) Frame 97

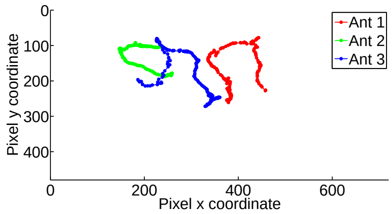
(c) Frame 197



(d) Frame 1



(e) Tracking till frame 97



(f) Tracking till frame 197

Figure 1: Tracking twenty ants. Displaying just three for clarity.

unique particle states. The full set of particles is not used only to decrease the computational burden.

The x and y hypothesis contained in each particle, together with the constant z coordinate of the target, are then used as the center of an ellipse, whose rotation around the z axis is set equal to the target orientation estimated in the previous iteration. The ellipse is then used for the application of the kernel based tracking method of Comaniciu et al. [1], together with the target appearance model. This is a method that, using the mean-shift algorithm, iteratively moves the elliptical region to a location in the image, where the RGB histogram of the pixels inside it more closely resembles the target RGB histogram. The similarity measure used is the Bhattacharyya coefficient. Finally the convergence points obtained are back-projected into the world referential frame and stored in the set $\mathcal{O}_t$, together with those of the targets that have already been iterated.

The points contained in $\mathcal{O}_t$ are then clustered using a mixture of Gaussians, with the number of clusters being equal to the number of targets, N_T. The mixture of Gaussians means are then stored to be used as observations for the posterior update of the optimal proposal particle filter. It is however necessary to associate these observations to their respective target. For that a Global Nearest Neighbours (GNN) [6] approach is employed by using the Hungarian algorithm [7], where each element of the cost matrix is equal to the distance between the observation, associated to each row, and the previous estimation of a targets position, associated to the columns. This process of clustering intends to add robustness to the problem that occurs when two similar targets are close to each other and the elliptical region of one target is attracted to the other one since this happens to have a higher Bhattacharyya coefficient.

With the observations associated to their respective targets a last step is done to obtain observations for the targets orientation. This is done through an exhaustive search in which an ellipse centered at each of the observations is placed in different rotation values. The one which gives a greater Bhattacharyya coefficient is used as the orientation for that particular observation. With the observations finally obtained, the likelihood is computed for each particle through the measurement equation $\mathbf{z}_t = \begin{bmatrix} I \mid & \mathbf{0} \end{bmatrix} \mathbf{x}_t + v_t, v_t \sim \mathcal{N}(\mathbf{0}, R)$, where I is the identity matrix.

6 Experimental Results

The proposed algorithm was applied to a video were twenty ants are moving in a dish [8]. This video has a dimension of 720×480 pixels and a duration of about eight seconds. The initial positions of the ants are set manually and sent to the algorithm.

In order to analyze the quality of the results a comparison is made with a multi-target tracker using background subtraction to detect the ants and the Multi-hypothesis tracker (MHT) algorithm for tracking and data association [9]. However as there are a considerable number of ants, only the trajectories and error values of three randomly selected ants are presented. The ground truth was manually extracted from the video.

In figures 1(a) to 1(c) three frames of the ants video are shown. Associated to them the trajectories estimated by the method proposed, named Multi-target Hybrid Particle Filter (MHPF), to the respective frame. The table 1 shows the root mean square error (RMSE), the total number of tracks lost (TL) and the number of tracks recovered (TR), for the several ants and the two algorithms.

	MHT			MHPF		
	Ant1	Ant2	Ant3	Ant1	Ant2	Ant3
RMSE	6.35	2.54	4.39	4.14	1.95	2.88
TL	152			0		
TR	152			-		

Table 1: Performance measures between the two algorithms

A track is considered lost when it changes the label that it uses to identify a target. As can be seen the MHPF showed better results in comparison with the other algorithm. Not only the RMSE is lower for all the ants but it also did not lose any track.

7 Conclusion and Future Work

The proposed tracking algorithm proved to give good results, even when the targets being tracked have all the same appearance. However it still needs further improvement, especially in the handling of total occlusions.

Acknowledgments

This work has been partially supported by the FCT project PEst-OE / EEI / LA0009 / 2013, by the QREN project 34063 SEAGULL, and by the EU project POETICON++ FP7-ICT-288382.

References

[1] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

[2] Emilio Maggio and Andrea Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *ICASSP (2)*, pages 221–224, 2005.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[4] Tiago Castanheira. Multitasking of smart cameras. *Dissertação para a obtenção do Grau de Mestre em Engenharia Electrotécnica e de Computadores*, 2013.

[5] Anton J Haug. *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley & Sons, 2012.

[6] P. Konstantinova, A. Udvarev, and T. Semerdjiev. A study of a target tracking algorithm using global nearest neighbor approach. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech)*, 2003.

[7] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[8] Zia Khan, Tucker Balch, and Frank Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *Computer Vision-ECCV 2004*, pages 279–290. Springer, 2004.

[9] David Miguel Antunes, David Martins de Matos, and José Gaspar. A library for implementing the multiple hypothesis tracking algorithm. *arXiv preprint arXiv:1106.2263*, 2011.