A Review on Particle Filter for a Video Based Tracking Problem

Report Author - Luca Marson
Groupt Partner - Alfredo Reichlin

Abstract—Tracking of object moving along highly non linear trajectory is a challenging task and the Particle Filter is a valid choice for solving this problem because it is able to deal efficiently with multimodal distributions and non linear motion models.
This project summarizes different ways of integrating color feature and edge detections to implement a robust estimator for the tracking problem. Color distribution are robust to partial occlusions and are rotational and scale invariant, but they perform poorly when the background is similar to the object to be tracked. Edge detection on the other hand is robust to this kind of problem, but it can be easily tricked by shapes similar to the reference one, like the object shadow.

1 INTRODUCTION

The tracking of an object is a challenging task which can have many applications, such as video surveillance, video compression as well as application in the entertainment market, like virtual reality. The Particle Filter is an ideal estimator for this kind of problem because it can provide a robust framework for highly non linear systems, in which multiple hypotesis are to be taken into account [1].
Implementing the Particle Filter to solve the tracking problem means finding a suitable motion model, as well as different osservation models. We followed two different approach: one which consists in considering a cloud of particle representing single pixel on the video frame, and another which assing a subregion of the frame to each particle. We experimented with both color and edge detection as suggested in [3].

2 RELATED WORK

Many in the scientific literature tried to solve the tracking problem. One popular solution is a deterministic approach known as Mean Shift Tracker by Comaniciu et al. [7], which iteratively search for a local maxima of a color similarity measure between a target reference and the current video frame.
A different approach is using probabilistic methods, like the Particle Filter or the Kalman Filter: the object to be tracked is modeled by developing a state-space representation which mirrors its dynamics. One of the first example developed by the computer vision community was the Condensation algorithm [4], which relies on real time tracking of an object edge features. Another implementation of the Particle Filter for tracking was proposed in [1], where colour features are exploited for robustness and adequately low computational costs.

3 IMPLEMENTATION

3.1 Particle Filter

According to [8], the Particle Filter is a non-parametric implementation of the Bayes filter. The state X_t is represented by a set of random state samples drawn from the posterior which approximates it by associating a weight w_t to each variable. The state evolution is described by a state model $p(x_t|x_{t-1})$ from which the particles are drawn. Their weight is evaluation according to the observation model $p(z_t|x_t)$, where z_t denotes an observation. The particles are then resampled by drawing with replacement from the particle set with a probability proportional to the weights.
As stated in [4], the Particle Filter is valid

choice for the tracking problem because it is able to represent multimodal states and to take into account non linear motion models.
During resampling, samples with a high weight may be chosen several times, leading to identical copies, while others with relatively low weights may not be chosen at all
In the following the main aspects of our implementation of a Particle Filter for object tracking based on color and edge features are outlined.

3.2 Our Solution

We developed two different approaches to solve the tracking problem using the Particle Filter, based on what we found in the field literature. Both are based on color and edge feature extraction, but they differ on how the state is represented.
For every solution, we represented the frame using both the pixel RGB values and HSV values. According to [1], using the HSV domain for this purpose allows more robustness to brightness change by damping the V value of the color representation.

Fig. 1: Particle Cloud and Histogram models

3.3 Initialization

We developed solution for both the tracking problem and the global localization problem.

- For the tracking problem, the starting point is initalized manually so that the estimator is able to trace its future trajectory. The initial particle set is sampled from a gaussian with mean equal to the object center and variance set so that the particles covers most of it. The same set is used as reference for the rest of the tracking operation of the Particle Cloud model, while for the Histogram Model just one particle is used as reference.
- For the global localization problem, the reference is set as before, while the initial particle set is sampled so that the particles are spread uniformly all around the frame. The estimator is able to localize the object after a few frames as showed in the results section.

3.4 State Model

3.4.1 Particle Cloud Model

The state model we deployed in order to describe the tracked object motion is based on the one proposed in [2]:

$$\mathbf{x_t} = \mathbf{x_{t-1}} + \mathbf{v_t}$$

The state x_t is given by:

$$\mathbf{x_t} = \{x, y\}$$

where x,y are the particle coordinates in the frame. Each particle therefore represent a single pixel RGB/HSV value.
v_t is drawn from a uniform distribution whose support is proportional to the distance of the current mean state to the previous one.
Even if extremely simple, this model is able to represent object moving in all direction, which is often the case in this kind of problem.

3.4.2 Histogram Model

For this model the state representation is the same, despite for the fact that a particle represents a square subregion of the frame whose side length is fixed for sake of simplicity.

In order to recover from loss due to total occlusion, we implemented a mutation mechanism which randomly reassign a particle coordinates with a certain probability p.

3.5 Observation Model

3.5.1 Particle Cloud Model

For each particle, the weight is proportional to the euclidean distance between the RGB value of the particle and the relative reference pixel.

3.5.2 Histogram Model

For each particle at a given timestep, a color histogram is built as described in [5], which represents the particle color distribution. This approach provides robustness against non-rigidity, rotation and partial occlusion. The histograms are calculated in the RGB space using 8x8x8 bins.
Each pixel of the particle subregion are passed trough a kernel which penalizes those far away from the center in order to increase the reliability of the color distribution [1].
The particle weight is evaluated by computing the Bhattacharyya distance between its histogram p and the one of the reference subregion, q.
The Bhattacharyya distance is evaluated as follows[1]:

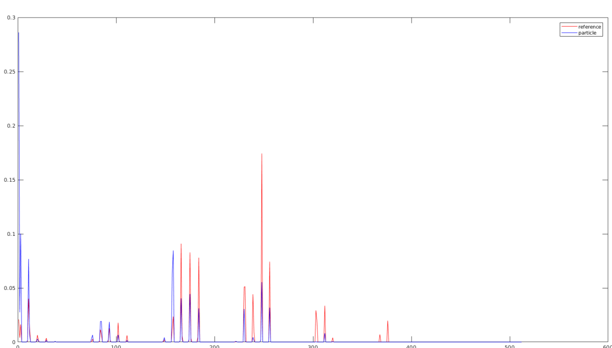
$$\rho[p,q] = \sum_{u=1}^{m} \sqrt{p(u)q(u)}$$

$$d = \sqrt{1 - \rho[p,q]}$$

where m is the number of bins.
Since we want the distance to be as small as possible, the particle weight is specified by a gaussian with variance determined empirically.

$$w_t = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{d^2}{2\sigma^2}}$$

Figure 2 compares the color distributions of a particle and of the reference picture.



We deployed the same observation model for the edge detection Particle Filter as suggested in [2], by building a histogram of 8x8 containing the magnitude and phase of each pixel of the subregion costructed with the sobel operator.

Fig. 2: RGB histograms

We also merged the color and edge features detection in the following way:
the particle weight is first calculated using the Particle Cloud observation model. A fixed number of particles are then selected randomly from the particle set and a subregion around them is used to perfrom the edge detection as in the Histogram Model. If the weight of the selected particles is higher than a certain threshold, the weight of other particle laying inside the subregion is increased.

For all the models, the current state is evaluated by calculating the particles position average, weighted by their measurement probability [1].

3.6 Resampling

We used Low Variance resampling and Multinomial resampling as described in [8].

4 Experimental Results

We recorded a video to test the strong and weak points of our algorithm. The object to be tracked is a yellow glove which moves along a highly non linear trajectory. The video resolution is 640x480, it lasts 16 seconds and its framerate is 15frames/s.
From frame 100 a total occlusion occurs, which allows us to test the loss recovery system.
The background color is similar to the object one and from frame 190 the light conditions change drastically so that we can test the robustness of the color cue detection. Light is switched on again after frame 220.
The glove is illuminated from above and it projects a sharp shadow on the floor, so that we can test the robustness of the edge feature detection.

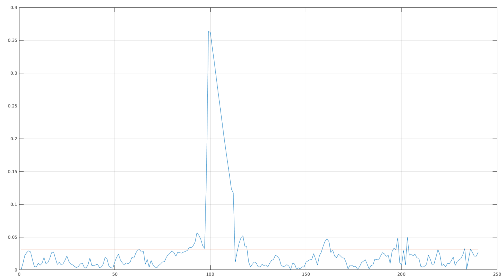

0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
50
100
150
200
250

Fig. 3

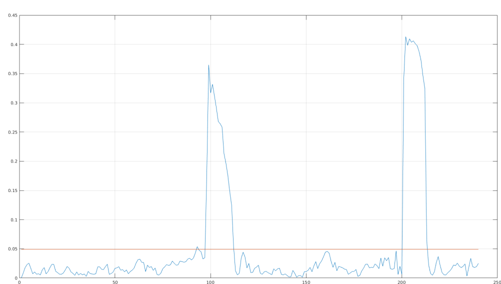

0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
50
100
150
200
250

Fig. 4

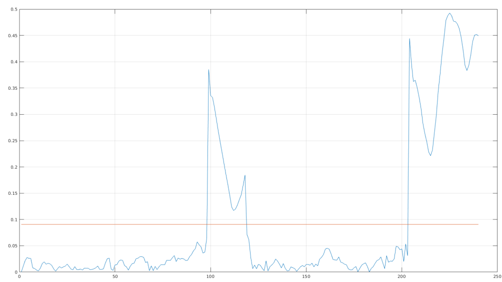


Fig. 5

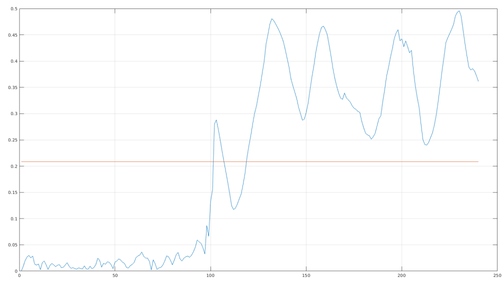


Fig. 6

The plots above show the normalized distance with respect to the frame diagonal from the filter state to the ground truth (manually evaluated) for several runs of the Particle Cloud model. The straight line is the average distance.

- Figure 3
  In this experiment, the estimator looses the object position only when it is completely occluded at fram 100, but it is able to recover it as soon as the object become visible again thanks to the mutation mechanism. It is not affected at all by the change of illumination at frame 190.
- Figure 4
  In this experiment, the estimator behaves like before excepts that it looses the position also after the light change, but it is able to recover after a few frames thanks to the mutation mechanism.
- Figure 5
  The estimator is not able to recover the position after the change of light.
- Figure 6
  The estimator does not recover the position at all after the object total occlusion.

4.1 Total occlusion

Fig. 7: Frame 96

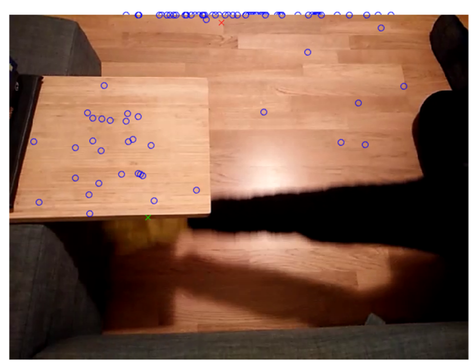


Fig. 8: frame 100

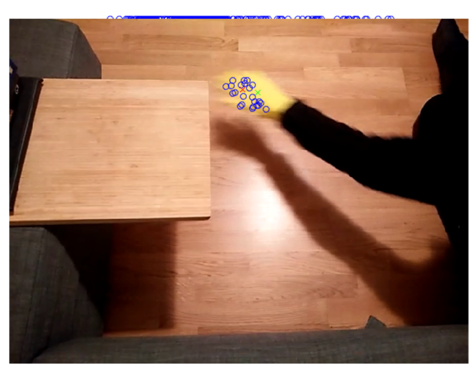


Fig. 9: frame 116

These pictures show how the estimator behaves when total occlusion occurs.
During occlusion (Figure 8) the particle spreads uniformly because of the diffusion step and

because the observation model assigns an approximately uniform weigth.
A few frame after the total occlusion (Figure 9), the estimator recover the object position because some of the diffused particles fall onto it.

4.2 Light change

Fig. 10: Frame 186

Fig. 11: frame 191

This experiment show the limits of the tracking based on color features detection.
The observation model is tricked by the light change and the estimated state is far from the ground truth when the light goes off (Figure 11).
Even when the light is switched on again (Figure 12), the estimator is not able to recover the real object location because most of the particles are clustered in the wrong position. This is also due to the fact that the camera

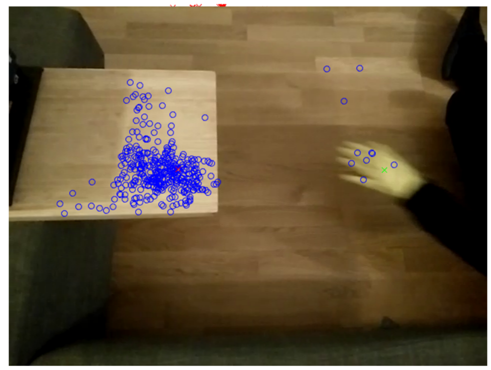


Fig. 12: frame 221

shutter reacts slowly to the light change and the colors are rendered differently with respect to the beginning of the video.

4.3 Mutation rate

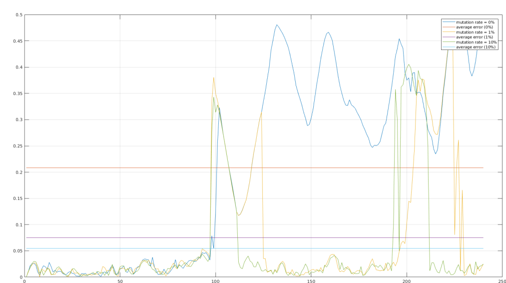


Fig. 13: Mutation rate

In this experiment we tested different values for the probability of particle mutation, to understand how they impact on the estimator performance. In Figure 13 it can be seen that the higher is the mutation rate, the higher is the probability for the system to recover the object position after occlusion and change of light. This can be noticed from the width of the peaks after frame 100 (occlusion), and frame 190 (light change).

4.4 Particle set cardinality

In this experiment we tested different dimensions for the cardinality of the particle set for both models. In both plots it can be seen that the higher is the number of particles, the better are the filter performances.

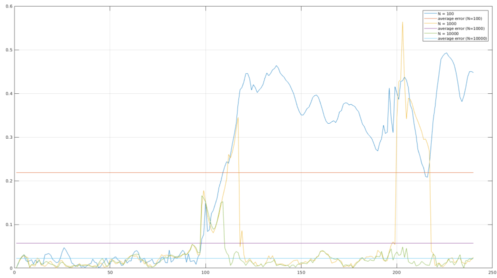


Fig. 14: Particle Cloud Model

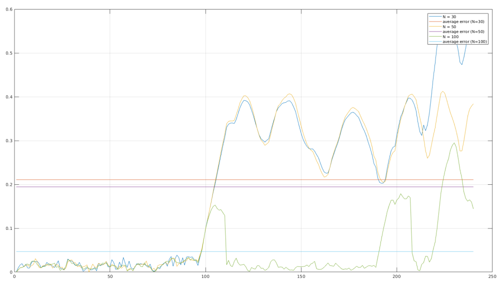


Fig. 15: Histogram Model

4.5 Edge detection

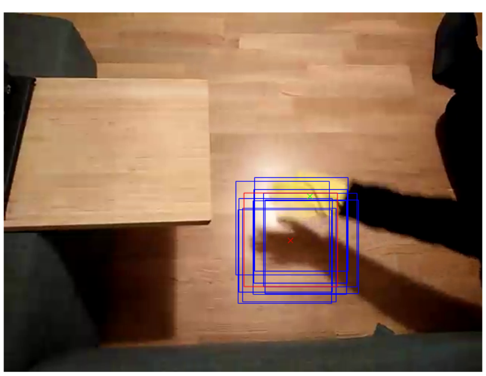


Fig. 16: Mutation rate

In Figure 16 it can be seen that the Histogram Model with edge detection can be easily tricked by shapes looking like the one of the object to be tracked, like a shadow. Edge detection on its own is not robust enough, and this is why we tried to implement it together with color cue detection in the Particle Cloud model.

4.6 Models comparison

In this experiment we tested how different model configurations work.

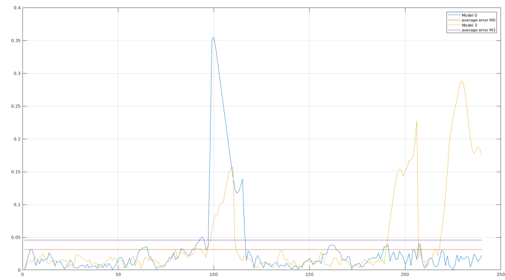


Fig. 17: Particle Cloud Model vs Histogram Model

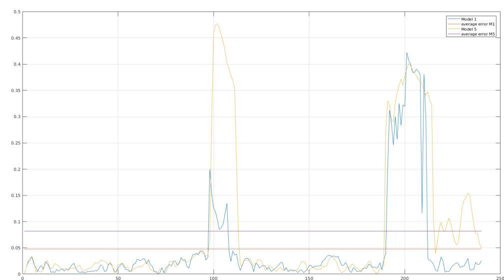


Fig. 18: Particle Cloud Model with Edge cue (RGB vs HSV

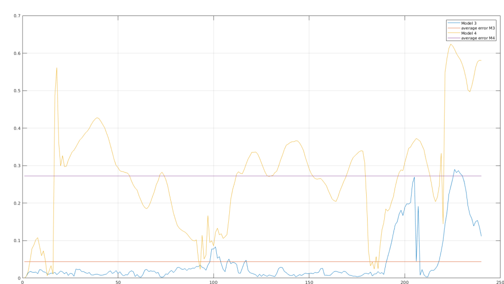


Fig. 19: Histogram Model - Color cue vs Edge cue

5 Summary and Conclusion

In this project we implemented different solution to tackle the tracking problem with Particle Filter. The Histogram Model advantage is that it is able to take into consideration a wider set

of hypotesis with respect to the Particle Cloud model, but at the expense of a much higher computation cost, because it has to read the value of a great number of pixels.
There are plenty of further steps that can be taken in order to improve our models.
As illustrated [5], we can extend the Histogram Model to track multiple objects.
We can integrate other feature detection techniques, like taking into account texture cues as suggested in [3].
We can implement advanced machine learning algorithms like Support Vector Machine to automatically initialize the estimator, so that there is no need to manually select the object(s) to be tracked[1].

References

[1] K. Nummiaro, E. Koller-Meier, L. Van Gool, *An adaptive color-based particle filter*, Image and vision computing 21 (1), 99-110, 2003.
[2] C. Yang, R. Duraiswami, L. Davis, *Fast Multiple Object Tracking via a Hierarchical Particle Filter*, Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1.
[3] L, Mihaylova, P. Brasnett, N. Canagarajah, D. Bull, *Object Tracking by Particle Filtering Techniques in Video Sequences*, Advances and challenges in multisensor data and information processing Volume 8, 260-268, 2007.
[4] M. Isard, A. Blake, *CONDENSATION - Conditional Density Propagation for Visual Tracking*, August 1998, Volume 29, Issue 1, pp 528.
[5] J. Czyz, B. Ristic, B. Macq, *A Particle Filter for Joint Detection and Tracking of Color Objects*, Image and Vision Computing, Volume 25, Number 8, August 2007.
[6] J. Deutscher, A. Blake, I. Reid, *Articulated Body Motion Capture by Annealed Particle Filtering*, Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662).
[7] D. Comaniciu, V. Ramesh, P. Meer, *Real-Time Tracking of Non- Rigid Objects using Mean Shift*, Computer Vision and Pattern Recognition (2000) 142149
[8] S. Thrun, W. Burgard, D. Fox, *Probalistic robotics*, MIT Press, (2005).