An efficient Visual Object Tracking Method by hybrid approach

Thien Bao Bui
Condorcet university center
University of Burgundy
Le-Creusot, France
Thien-Bao_Bui@etu.u-bourgogne.fr

***Abstract*—Visual Object Tracking remains one of the most difficult problem to solve in Computer Vision field. Many algorithms, many approaches have been made however the task is still far from being achieved due to various constrainsts such as illumination variation, occlusion, non-rigid object, etc. In this paper, we are going to adopt a hybrid approach which combines the deterministic and probabilistic methods together, namely Mean-shift and Particle-Kalman filter, to overcome the problem. We expect these algorithms will cover each other drawbacks in order to achieve the task. Indeed, the result is good and will be discussed in the last section.**

I. Introduction

Over the years, tracking remains an open research discussion in computer vision. Difficulties encountered until now mainly come from the environment and the characteristic of the object itself. The problem from the environment can be listed as cluttered background, illumination variation. On the other hand, the limitation related to the attribute of the moving object is usually the variation of the target object appearance over time (or frames). In case of rigid object, we can take advantage of the known shape of the object to build its accurate model. Otherwise, no accurate model can be built and the tracking task can't be performed, not to mention the case of fast motion object or non-linear motion. In addition, when the target make an interaction with another object, it can cause, either partial or full, occlusion. The background frames computation will be greatly affected. Moreover, the situation where two or more objects have the same color distribution as our target always exist in video scene. These issues can be characterized in three main concepts: deformation, illumination and occlusion [1].

So far, all the algorithms invented for visual tracking can be classified in two groups: deterministic and probabilistic method [insert citation here]. The classical problem of visual tracking can be stated as: given the past and the current measurements (raw of filtered images, low-level features, high-level detection) $z_{1:t} := (z_1, ..., z_t)$, and the single target state x_t, output an estimate of current hidden state

$$\hat{x}_t = f(z_{1:t})$$

Deterministic method would try to optimize ad-hoc objective function

$$\hat{x}_t = \arg\min E(x_t; \hat{x}_{t-1}, z_t)$$

, or minimize the function $E(x_t; z_t)$ "around" $\hat{x}_{t-1}$. On the other hand, probabilistic method would perform the computation of the filtering probability density function $p(x_t|z_{1:t})$ and point estimation

$$\hat{x}_t = \arg\max p(x_t|z_{1:t}) \text{ or } E[x_t|z_{1:t}]$$

Among the deterministic tracking method, mean-shift is widely used even though it can not track fast-motion object and perform badly in case of occlusion. However, it is quite robust against partial occlusion, not to mention its low complexity algorithms which is suitable for real time tracking. Mean-shift is considered as non-parametric estimation for density which repeatedly finds the highest degree of resemblance between the distribution patterns of candidate region in current frame with target models distribution pattern [2]. On the other hand, Kalman filter and Particle filter are typical examples of probabilistic methods. Similar to mean-shift, Kalman filter is easy to implement real time tracking. But unlike mean-shift, Kalman filter performs well, in case of occlusion, the task of predicting the location of linear motion object. However, for non linear motion object, its performance is pretty poor. In contrast to Kalman filter, Particle filter can deal with objects which changes their attitude and velocity abruptly. To define the probability distribution, Particle filter employ unsystematic samples called as "particle", where each particle is associated with a certain weight. Then the computation will be computed based on these samples and weights. In addition, Particle filter also shows the ability to recover from losing track due to full occlusion. Besides these positive points, it is worth noticing the drawback of this filter, that is the need of huge amount of samples or particles, even it is necessary since every single particle has low contribution and efficiency, increases the complexity of the algorithm. Hence, it is not possible to implement this filter for real time tracking task.

In order to overcome all the challenges mentioned above, this paper is created with a robust tracking algorithm which is a combination of different algorithms as a mean to cover each algorithm limitations.

II. Related work

Since the use of single tracking algorithm is not efficient, several papers nowadays have combined different techniques to achieve the visual tracking task. The combination of mean-shift and kalman filter algorithm is proposed to track object under occlusion [3]. Kalman filter is used to predict the target location when occlusion occured. And the result shows that the algorithm is robust against occlusion. The only drawback is the limitation in dealing with non linear motion. Similarly, to deal with occlusion, Hongxia Chu et al [4] came up with another advance Mean-shift tracking algorithm whose experimental results confirm its ability to track the object when both illumination variation and occlusion occured. Instead of utilizing common histogram to represent the color of each object, Hongxia utilize weighted histogram with spatial corrected background and provide a complete derivation of it. To deal with non linear motion, Chen [5] has proposed the combination of mean-shift and particle filter. This algorithm performs well not only with non linear motion, but also with occlusion despite using only few particles. Nonetheless, the tracking speed is not improved much because of the required iterations of mean-shift for each particle. Moreover, the high density of the particle set due to the impact of mean-shift on each particle may decrease the quality of tracking process, especially when the target recovers from any partial/full occlusion. Another approach is to combine particle and kalman filter [6]. It is worth noticing that the non linear motion usually happens in local view only, while the linear one is recognized in global view. Therefore, the idea of combining particle and kalman filter is to perform local motion and global motion estimation respectively. Compared to the combination of mean-shift and particle filter, this proposed algorithm can reduce greatly the number of particles without adding any iterations. Therefore, the speed of the system can be improved significantly.

Extending the idea of combining different techniques together, this paper propose a combination of mean-shift and particle-kalman filter. Mean-shift will be used as master tracker when there is no occlusion. Particle-kalman filter will be used when the result from mean-shift is not convincing or the occlusion occured.

III. Methodology and implementation

Before going to the detail of the implementation, let's go through the overview of mean-shift, kalman and particle filter. Then the implementation detail will be explained in the following sections.

A. Mean-shift

The basic idea of mean-shift is stated as follows: given a set of data points, the algorithm iteratively assigns each data point towards the real/closest centroid of the cluster. The direction of to the closest cluster centroid is determined by where most of the data points nearby are at. On figure 1, given a small window (initially C_1), move the window to the are of maximum pixel density. The initial window is C_1, centered at the blue square C_{1_o}. Then find the real centroid of the window C_1 and it is C_{1_r}. Since they are not matched, move the window C_1 to the new center C_{1_r}. Repeat the process again until the new centroid found is matched with the current centroid or the error between the new and current centroid is smaller than a predefined threshold, the algorithm stops.

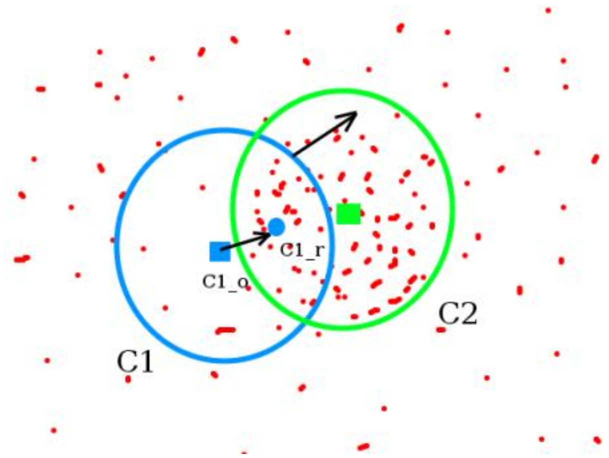

Fig. 1. Mean-shift algorithm illustration. The image is taken from [7]

B. Kalman filter

The famous Kalman filter algorithm can be summarized as follows:

Algorithm 1: Kalman filter Algorithm

Input: x_{k-1}, P_{k-1}, u_k, z_k
Output: $\hat{x_k}$, $\hat{P_k}$

1) Set $k = 1$
2) Make prediction

Predicted state:

$$\hat{x}_{k|k-1} = F_{k-1}x_{k-\hat{1}|k-1}$$

Predicted error covariance:

$$P_{x,k|k-1} = F_{k-1}P_{x,k-1|k-1}F'_{k-1} + Q_k$$

3) Predicted measurement: $y_{k|k-1} = H_k\hat{x}_{k|k-1}$
4) Innovation covariance matrix: $P_{y,k|k-1} = H_kP_{k|k-1}H'_k + R_k$
5) Kalman gain: $K_k = P_{x,k|k-1}H'_kP^{-1}_{y,k|k-1}$
6) Updated state: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1})$
7) Error covariance: $P_{x,k|k} = P_{x,k|k-1} - K_kH_kP_{x,k|k-1}$
8) Set $k = k + 1$ and repeat from step 2

C. Particle filter

The idea of particle filters is to find an approximate representation of any probability density function (pdf) as a set of particles (samples). Each particle has one set of values for the state variables. This method can represent any arbitrary distribution, making it good for non-Gaussian process (non linear motion object for example), multi-modal pdf. The algorithm of this particle filter can be summarized as follows:

Algorithm 2: Particle filter Algorithm

Input: A proposal distribution $q(x_{k+1}|x_{1:k}, y_{k+1})$, a resampling strategy, the number of particles N

Output: An approximation of the trajectory posterior density $p(x_{1:k}|y_{1:k})$

Initialization: Generate $x_1^i \sim p_{x_0}$, $i = 1, ..., N$ and let $w_{1|0}^i = 1/N$;

for $k = 1, 2, ...$ **do**

1) Measurement update:

   for $i \leftarrow 1$ ***to*** N **do**

$$w_{k|k}^i = \frac{1}{c_k} w_{k|k-1}^i p(y_k|x_k^i),$$

   where the normalization weight is given by

$$c_k = \sum_{i=1}^{N} w_{k|k-1}^i p(y_k|x_k^i)$$

   end

2) Estimation: The filtering density is approximated by

$$\hat{p}(x_{1:k}|y_{1:k}) = \sum_{i=1}^{N} w_{k|k}^i \delta(x_{1:k} - x_{1:k}^i)$$

   and the mean $\hat{x}_{1:k} \approx \sum_{i=1}^{N} w_{k|k}^i x_{1:k}^i$

3) Resampling: Optionally at each time, take N samples with replacement from the set $\{x_{1:k}^i\}_{i=1}^N$ where the probability to take sample i is $w_{k|k}^i$ and let $w_{k|k}^i = 1/N$

4) Time update: Generate predictions according to the proposal distribution:

$$x_{k+1}^i \sim q(x_{k+1}|x_k^i, y_{k+1})$$

   and the compensate for the importance weight

$$w_{k+1|k}^i = w_{k|k}^i \frac{p(x_{k+1}^i|x_k^i)}{q(x_{k+1}^i|x_k^i, y_{k+1})}$$

end

D. Model initialization

Firstly, we select manually the target model in any frame of the video by cropping the region of interest to determine the initial position. The tracking process will be performed from that frame until the end of the video. From the targeted object, the color feature is extracted, using the $8x8x4$ HSV weighted histogram proposed by Commaniciu et al [8]. The equation of the target color distribution is given by:

$$q(u) = C \sum_{i=1}^{n} (\| \frac{x_c - x_i}{h} \|)^2 \delta(b(x_i) - u)$$

where x_c defines the midpoint of the ROI, C is the normalized coefficient, and h is the normalizing constant ($h = \sqrt{H_x^2 + H_y^2}$).

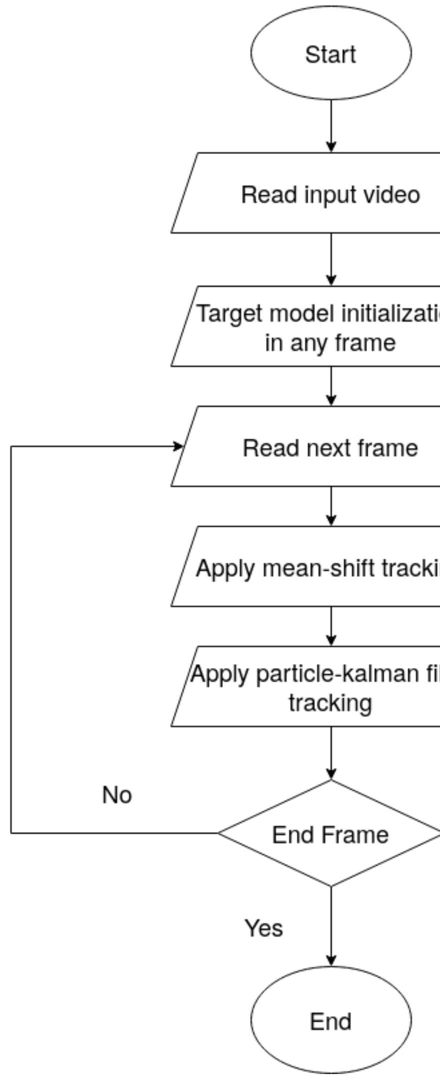

Fig. 2. Main flowchart

In HSV color space, we can deal with background illumination by thresholding the HSV histogram. We restrict the Hue values to fall between the range (0-180) and ignore the low saturation values which usually do not contain useful information. This thresholding approach is demonstrated by this equation:

$$g(t) = \begin{cases} \alpha t \; 1 < t \leq h \\ 0 \;\; \text{otherwise} \end{cases}$$

where α is the scaling factor and h is the bandwidth of the new search window.

E. Apply the combination of mean-shift and particle-kalman filter continuously

Due to the lack of time, the similarity function mentioned in the original paper can not be implemented. Therefore, the flow of algorithm is modified by applying three algorithms consecutively. The input of mean-shift is the ROI region as well as its preprocessed HSV histogram. The output tracked window of mean-shift will be the input for particle-kalman filter. Actually, the output of mean-shift will go to particle filter first to define the number of particles. Then this number will be decreased thanks to the linearity of Kalman filter. Therefore, we can achieve the task of tracking the non-linear as well as linear motion in local and global view respectively.

IV. Result

In this section, we are going to compare the effectiveness of three different combinations: mean-shift and kalman filter, mean-shift and particle filter, our proposed method.

Fig. 3. Cropping ROI and the HSV histogram corresponding

As this figure (3) shows, we can crop manually the ROI and the HSV histogram of the cropped region will also be computed and displayed in another window.

The following figures are used to illustrated the effectiveness of each algorithm in different scenario: linear motion (figure (4)), non linear motion (figure 5), occlusion (figure 6), and object with similar color distribution (figure 7). The white rectangle in each image is the result of mean-shift. The blue, red, green rectangles are results of each combination: mean-shift and kalman filter, mean-shift and particle filter, mean-shift and particle-kalman filter respectively.

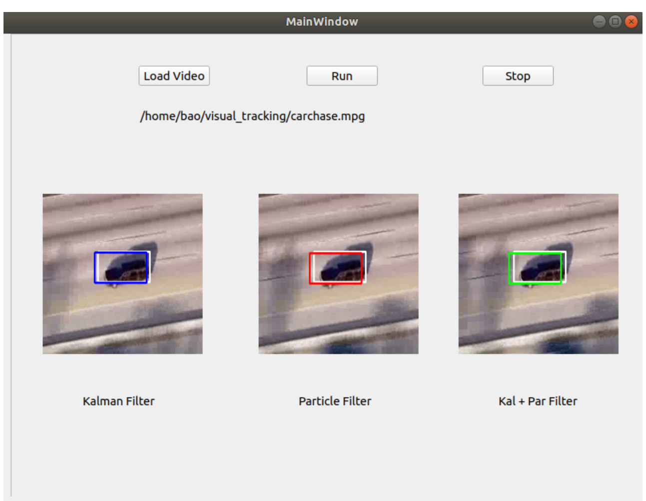

Fig. 4. Object is moving normally (linear motion)

When the car is moving normally, all algorithms show good result.

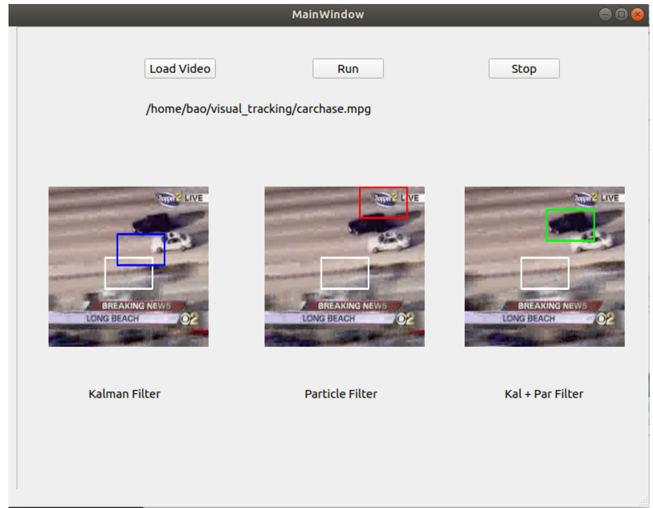

Fig. 5. Target suddenly changes its velocity

In this case, the target changes its velocity abruptly, the motion becomes non linear. As expected, the first combination can not track the object correctly since it has no ability dealing with non-Gaussian process. However, the second combination with particle filter also shows difficulty in tracking the target. While our implementation has no difficulty in tracking the target.

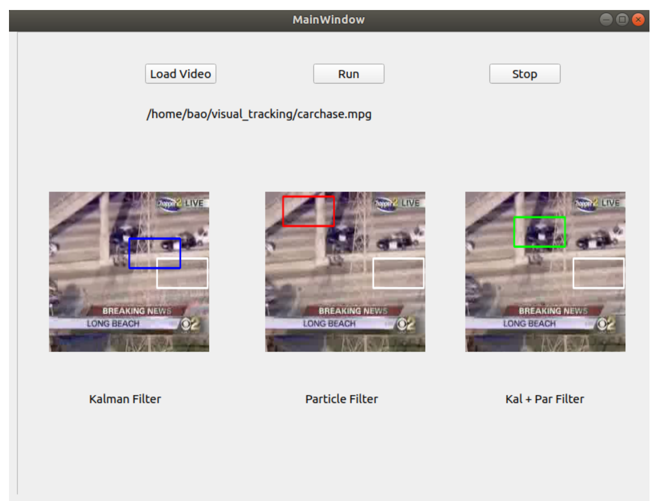

Fig. 6. Occlusion occurs

The similar result happens in case of occlusion. Our implementation achieves better result compared to the other two combination.

However, all algorithms have poor performance when there is an object with similar color distribution as our target appears next to it.

V. Conclusion and Future work

Extending the idea of combining different tracking algorithms, this method has been created with a great motivation

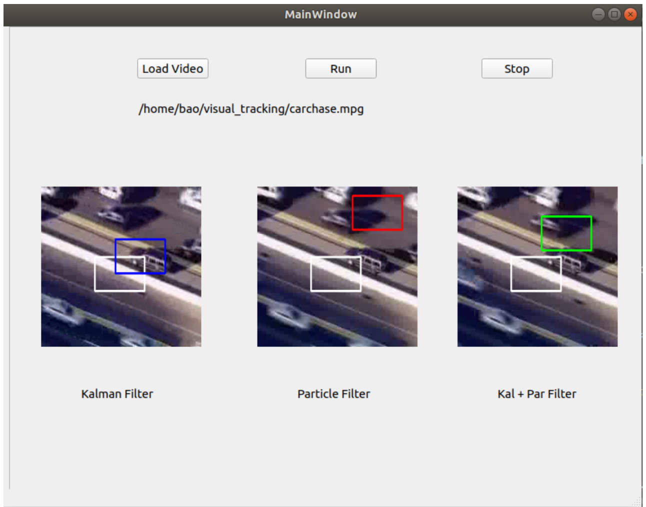

Fig. 7. Another object with similar color distribution as our target

for solving the difficulty in tracking task as well as covering each algorithms drawbacks. Indeed, the results have proved that this proposed method can deal with different problem in visual tracking such as non linear motion, occlusion and can be implemented in real time tracking.

Meanwhile, the last result shows the drawback of this proposed method in case of object with similar color distribution. One possible reason is that we just take into account the color of the target. To overcome this problem, more features of the target should be considered when developing the algorithm.

References

[1] S. H. Shaikh, K. Saeed, and N. Chaki, "Moving object detection approaches, challenges and object tracking," 2014.

[2] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[3] I. Iraei and K. Faez, "Object tracking with occlusion handling using mean shift, kalman filter and edge histogram," *2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 1–6, 2015.

[4] H. Chu, Q. Song, H. Yuan, Z. Xie, R. Zhang, and W. Jiang, "Research of mean shift target tracking with spatiogram corrected background-weighted histogram," in *2015 IEEE International Conference on Information and Automation*, pp. 1942–1946, 2015.

[5] K. Chen, D. Li, Q. Huang, and L. E. Banta, "Video motion tracking using enhanced particle filtering with mean-shift," in *2010 3rd International Congress on Image and Signal Processing*, vol. 1, pp. 387–391, 2010.

[6] S. Yin, J. Na, J. Choi, and S. Oh, "Hierarchical kalman-particle filter with adaptation to motion changes for object tracking," *Comput. Vis. Image Underst.*, vol. 115, pp. 885–900, 2011.

[7] https://docs.opencv.org/3.4/df/def/tutorial_js_meanshift.html.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.