Person Tracking in a Video using Particle Filter

Devrat Singh
devrat@kth.se

Abstract

The aim of this project is to implement and compare the performance of particle filter based pedestrian/person tracking mechanism, that considers different methods for target representation. The application of Pedestrian/Person tracking can be found in autonomous cars, sports tracking, and surveillance. There are a plethora of other applications as well, making pedestrian tracking an important research area. This report details the particle filter methodology which utilizes the color histogram and moment invariant observation models along with a Fusion strategy that calculates scene adaptive weights for combining the before stated models. The implemented algorithms are then tested on a challenging video sequence of a person walking, with frequent occlusions, moving camera, and similar looking distractions. Finally, a comparative study is then conducted to analyze the difference in performance between the utilized methods i.e. Color Based, Moment Based and Fusion. This evaluation is performed w.r.t two metrics: euclidean distance and overlap area. The results are ranked, revealing the Fusion based method as the best, closely followed by Color histogram and then Moment invariant model with comparatively least accurate tracking.

1. Introduction

The ability to track people in a videos is a common problem and it's significance keeps increasing, as we race towards a future where autonomous cars and other human-machine interaction become the norm. The applications of pedestrian tracking also stem to other areas, for example in surveillance to track suspicious people, for traffic management, for aerial and wheeled robots that follow a target, in sports to track players etc. Since the inherent application is related to images, one may solely think of visual tracking as a computer vision (CV) problem. This, in some way, is correct, as computer vision methods could be used to detect the person frame by frame. However, as it usually is, things are not that simple for practical implementations, due to factors like occlusions, light variations, other people, non-stable camera etc. For example, when tracking a person, the view of the camera could be obstructed by another person crossing in-between. There could also be similarly dressed people which make it harder for the detector to provide good measurements. During the duration of tracking, the illumination can change due to shadows, lights etc. and might confuse the detector if a set target profile is initialized. Finally, along with movements of the target, the camera might not be stable. These factors could contribute to poor measurements and may make tracking simply through CV methods lack robustness. Moreover, as we know, applied estimation filters are adept at combining the prowess of prediction and measurement to provide a reliable estimate. Accordingly, inspired by [9], we follow a strategy of combining the color histogram and moment invariant observation models to get a robust pedestrian tracker. The robustness is checked with regard to tracking under challenging circumstances, provided in the chosen dataset [4].

Outline This report follows an information flow beginning with section 1, which introduced the problem of pedestrian tracking, as well as it relevance. Next, section 2, will deal with discussing the literature surrounding the topic of object tracking in video and how that motivates this study. This is then followed by the Methodology section (3), which lists a detailed description of the techniques utilized in this implementation and how they tie all together. Finally, section 4 entails the experiments conducted, their results and subsequent analysis. Succeeded by section 5 reflecting on the conclusions drawn from this project.

2. Related Work

There are two main categories that need to be discussed when considering the topic of object tracking in an video sequence. These are Object detection and Object tracking.

The detection part, as the name implies, concerns the identification of the target from the background and other objects present in the image. The significance of the detection method is solidified by the reasoning that any tracking method would rely on the detector. As stated in [15] [11], the object detection methods can further be categorized into, 1) Point detectors, 2) Segmentation, 3) Background Modelling and 4) Machine learning based methods. The point

detectors include well known techniques like Harris interest point detector [6], SIFT detector [13] etc. The aim here is to find interest points and these are invariant to illumination and camera viewpoint changes. As for Segmentation methods, some examples are Mean Shift clustering [2] and Graph cut [16]. Jumping to the remaining relevant category, i.e., machine learning based detectors, these methods have gained significant popularity in the recent years. The idea is to identify a function based on labelled training data, such that the function can then provide an output (detected region of interest), given the input image. An example of a supervised classifier that has seen utilization for this purpose is SVM (Support Vector Machines)[1]. Accordingly, in this implementation, for object detection, we will be employing a linear SVM based on Histogram of gradients (HOG) [3].

For the purpose of this course, the literature search for object tracking was limited to particle filters. The origin of particle filter's application on object tracking is attributed to [8], [5], [10], which have individually proposed the concept. A reason that further catapulted particle filter's usage, is their ability to handle non-gaussian posterior distributions and complex non-linear behaviors, as opposed to Kalman filter variants. This makes them suitable for person tracking tasks where one could vitness complex and unexpected movements as well as occlusions. At the base, the particle filter implementation in most of the papers reviewed, seems to following the same structure. Moreover, the difference usually comes from the scheme used for target representation. For instance, in [17], the target is represented using a vector of seven invariant moments. On the other hand, [14] makes use of features dependent on color profile of the target. This form of tracking through color based features is shown to be robust against out of plane rotations, scale and rotation invariance but more sensitive to changes in illumination. The study in [9], combines the above two approaches for modelling the target, but for the purpose of face tracking. The authors in [9] show that a more reliable tracking could be achieved by combining the two types of observation models and thus, in tern compensate for their individual weaknesses.

Therefore, in this report, we will be following the fusion based approach from [9], but to track a person instead. This problem can be considered comparatively more challenging due to frequent partial and full occlusions, moving camera and detection confusion caused by similar looking objects to the target.

3. Methodology

3.1. Particle Filter

This section explores the particle filter implementation. A particle filter is a non-parametric filter, and often referred to as Sequential Monte Carlo (SMC) method [18]. The belief here is represented by a set of samples/particles with varying weights. Something that gives it an edge over the ubiquitous Extended Kalman filter, is that it can model non-linear transformations, without needing any form of linearization. This makes it especially appealing for video tracking situations, which often handles non-gaussian and non-linear distributions [15].

3.1.1 States

A conventional way to locate and track objects in images is through bounding boxes which encapsulate the property of interest. Accordingly, our target (person) will be defined in terms of the bounding box that covers it. This leads to the following state vector:

$$s_k = [x, y, v_x, v_y, h_x, h_y, \alpha]^T \tag{1}$$

Here, from Equation 1, x and y are the center coordinates, v_x and v_y are the velocities of the center coordinates, h_x and h_y are the width and height of the bounding box and finally, α is the scaling factor with regard to the initial bounding box.

3.1.2 Initialization

Selecting the Target At the start of the algorithm, we need to define a single target that will be subject to the tracking. In our case, this is obviously a person and will be the sole target that the algorithm will consider. For selecting the target, mainly two ways were considered: 1) Select the person manually by choosing the region in which the individual is located. 2) In the used dataset, the very first frame only has one person, and this is also our desired target. Therefore, we can automize the target selection without having much confusion between multiple bodies.

For this purpose, We make use of a method that combines Support Vector Machine (SVM) and Histogram Of Oriented Gradients (HOG), to detect a person. This module will be referred to as the Person detector or simply, detector in further sections. The output from this detector is a bounding box that encapsulates the target. The bounding box from the detector is characterized by $[x, y, h_x, h_y]$, where x, y are the center coordinates of the box and h_x, h_y are the width and height, respectively. The target information then needs to be modified to fit the state vector format (Equation 2) and thus the target state vector is of the following form:

$$s_0^{target} = [x, y, 0, 0, h_x, h_y, 1]^T \tag{2}$$

Note, after the target initialization, the target is only updated according to a confidence level judged in the detector.

Initialize Particles The particles that represent the filter's estimated belief are each described through their corresponding state and weight. Hence, we can denote them as: $p_k^n = [s_k^n \quad w_k^n]^T$, where this is for the n^{th} particle at iteration k and s_k is the state vector for that particle along with w_k being its weight i.e., the value that describes how accurately can that particle represent the state of the system.

During initialization, the particles are spread uniformly across the whole state space (the whole frame) and weighed equally. So, if there are N particles, then each particle is initialized with weight $\frac{1}{N}$ at a random position in the frame.

In this implementation, there will be two particle initialization, one for each type of observation model. So, basically this implementation could be considered a combination of two particle filters whose posterior beliefs are combined. This will be discussed in section 3.1.4 and 3.1.7

3.1.3 Prediction Model and Propagation

In the video, the target individual is moving continuously and with no significant acceleration during the whole motion. As such, we utilize a constant velocity model, inspired by the related literature [15]. Therefore, in each iteration of the algorithm, the particles are propagated as shown in equation 3. In other words, the center coordinates change between frames according to the person's walking velocity, while the width and height of the bounding box are changed according to α from previous iteration.

For each particle:

$$\begin{aligned} x_k &= x_{k-1} + v_{xk-1} + \mathcal{N}(0, \Sigma_R) \\ y_k &= y_{k-1} + v_{yk-1} + \mathcal{N}(0, \Sigma_R) \\ v_{xk} &= v_{xk-1} + \mathcal{N}(0, \Sigma_R) \\ v_{yk} &= v_{yk-1} + \mathcal{N}(0, \Sigma_R) \\ h_{xk} &= \alpha_{k-1} \cdot h_{xk-1} + \mathcal{N}(0, \Sigma_R) \\ h_{yk} &= \alpha_{k-1} \cdot h_{yk-1} + \mathcal{N}(0, \Sigma_R) \\ \alpha_k &= \alpha_{k-1} + \mathcal{N}(0, \Sigma_R) \end{aligned} \tag{3}$$

In the above motion model, $\mathcal{R}$ represents the process noise. So, $\mathcal{N}(0, \Sigma_R)$ is a multivariate Gaussian distribution with zero mean and standard deviation given by $\mathcal{R}$.

3.1.4 Measurement Models

The prediction part did the job of propagating the particles, where each particle represents a probable state. But not all particles would provide good estimates, so in order to differentiate the particles, their weights need to decided. We decide this based on the comparison between the particle's estimated bounding box and the measured target's bounding box. Correspondingly, this comparison can be done in various ways, moreover, in this implementation it is done through color based and moment based models.

Color Based Model According to [12], the HSV color space provides some invariance to illumination changes. Therefore, the enclosed region within each particle's bounding box undergoes some transformations to get to the HSV-histograms of that space. The Bhattacharya distance (shown in equation 4) is then calculated for each particle w.r.t to the target bounding box's HSV-histogram. This distance shows the similarity between the two compared spaces, and its values are between 0 and 1, where 0 denotes an exact match [9].

$$d = \sqrt{1 - \sum_{1}^{512} \sqrt{q \cdot p^{(n)}}} \tag{4}$$

Here, q and $p^{(n)}$ are 1×512 flattened vectors. q is for the target and $p^{(n)}$ is for the n^{th} particle.

Moment Based Model Another type of observation model that we have is the Moment based model. Before we discuss the details, it's worth pointing out the benefits of adopting this technique. As opposed to the color based information, the moment values are capable of describing the spatial information. Furthermore, this is invariant to 2D transformations like, translations, rotation, and scaling and independent of chromatic content [9].

The main goal here again is to parameterize the region within the bounding boxes and then compare the predicted estimates against the target. In this method, we try to find the moments of order (p+q) for the image function $f(x, y)$ using the following equations defined in [7] and used in [9]

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad for \quad (p, q = 0, 1, 2.....) \tag{5}$$

The image centroid is defined as:

$$\mu_{pq} = \sum_x \sum_y (x - \frac{m_{10}}{m_{00}})^p (y - \frac{m_{01}}{m_{00}})^q f(x, y) \tag{6}$$

Then calculate the normalized central moment:

$$\eta_{pq} = \mu_{pq} / \mu_{00}^{(p+q+2)/2+1} \tag{7}$$

All this leads to the central moments being expanded into the Hu moments as follows:

$$
\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})^2(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&\quad - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\quad [3(\eta_{21} + \eta_{03})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})^2[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&\quad - 3(\eta_{21} - \eta_{03})^2] + 3(\eta_{12} - \eta_{30})(\eta_{21} - \eta_{03}) \\
&\quad [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned} \tag{8}
$$

With this, we can make a 1×7 Hu moment vector (Φ) for the particles ($\Phi_p^{(n)}$) and the target (Φ_T). The moments need to be standardized and then we can calculate the similarity between the target and the hypotheses using equation 9. Here, as before, $d = 0$ signifies an exact match.

$$
d = \frac{1}{7}\sum_{i=1}^{7} abs\left(\frac{\Phi_T - \Phi_p^{(n)}}{\Phi_T + \Phi_p^{(n)}}\right) \quad for \quad n^{th} \quad particle \tag{9}
$$

3.1.5 Weight Update

The likelihood of each particle being the target is determined by the similarity distances that were determined through the observation models, as discussed in the preceding section 3.1.4. Therefore, the weights are updated as shown in equation 10. Moreover, the weights should also be normalized and then each particle will hold a weight according to its likelihood.

$$
w_k^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(d_k^{(n)})^2}{2\sigma^2}\right)} \tag{10}
$$

The Fusion strategy will be discussed in a later section, however, something worth notifying is that if both observation models are used then there will be two sets of weights calculated w.r.t to each model for all particles.

3.1.6 Posteriori State Estimation

Once the weights have been updated, we can calculate the final state estimate by taking a weighted average of all particles (equation 11). One drawback of this method is that it can lead to incorrect results when there is more than one likely cluster of particles. Correspondingly, the choice of this method may seem counterintuitive, especially when particle filters excel at handling multimodal distributions. However, in our work case, the above scenario wasn't a recurring issue and with this method being quite computationally inexpensive, proved to be sufficient.

$$
s_k = \sum_{i}^{N} w_k^{(n)} \cdot s_k^{(n)} \tag{11}
$$

3.1.7 Fusion

The particle filter can function with either of the above mentioned observation models, moreover, there is an additional possibility to have the combined advantages of both models. Once the posterior state estimate has been calculated for both color based and moment based method, we can combine them through a weighted sum, where more prefernce is given to the most suitable method in that frame. These fusion weights are calculated as follows:

$$
s_k^F = w_k^{color} \cdot s_k^{color} + w_k^{moment} \cdot s_k^{moment} \tag{12}
$$

where,

$$
\begin{aligned}
w_k^{color} &= \frac{\exp(-\beta d_k^{color})}{\exp(-\beta d_k^{color}) + \exp(-\beta d_k^{moment})} \\
w_k^{moment} &= \frac{\exp(-\beta d_k^{moment})}{\exp(-\beta d_k^{color}) + \exp(-\beta d_k^{moment})}
\end{aligned} \tag{13}
$$

In equation 13, β is called the attenuation constant, d_k^{moment} and d_k^{color} are the euclidean distance between the respective observation method's estimated center (bounding box's center) and the target's center [9].

3.1.8 Re-sampling

This is one of the crucial steps for Particle filters. As the weights get updated, the particles representing less likely hypotheses gets weighted less. Then during resampling, these particles that are far from the likely state are removed and updated to be in the vicinity of the higher weight particles. Basically, we are allowing the higher weight particles to survive while re-placing the lower weight ones. With the outliers being removed and the high particle concentration in high likelihood regions, we end up with a more accurate system state estimate.

Out of many available methods for this purpose, because of familiarity and effectiveness, Systematic re-sampling was used.

3.2. Tracking Dataset

To test the particle filter implementation, we selected the BoBoT benchmark tracking dataset [4]. Although there are different tracking scenarios available in BoBot, we choose a video sequence which included the challenges that one

might encounter when tracking a person in real life. As such, the selected video sequence involves a person walking with a moving camera following him. During the walk, other people frequently cross in front of the camera, acting as occlusions. Some people are also dressed similar to the target individual, so it can create issues for the detector as well. The video sequence plays at 25 fps and there are 1017 frames in total, with each frame being equal to 320 x 240. Its characteristics could be summarized as: moving camera, moving target, non-rigid target, rotation, similar distractors, full occlusion, and outdoor [4] [15]. Finally, this dataset also includes the ground truth in terms of the bounding box around the tracked individual in each frame. This would allow for a more comprehensive evaluation of the implemented methodology.

4. Experiments and Results

To even begin, something that needed to be tested was if the particles were converging around the correct state. Which they did. As it can be seen in figure 1, where the red boxes represent the particles, the blue box is the averaged state estimate and the black box is the ground truth. The particles are initially spread uniformly over the whole workspace (figure 1b) and then in the subsequent iterations, they coverage around the true state as illustrated in figure 1c.

The objective now can be shifted to a comparative evaluation of Particle filter's tracking performance for Color based method, Moment based method and Fusion method. To quantify the difference between the ground truth and estimated state, two types of error were utilized. First, based on euclidean distance between ground truth's center and the state estimate's center. This is a fairly simple way of assessing the difference, however, it dose not take the width and height of the bounding box into account. For example, there could be scenarios where the center's co-align but the corresponding boxes might have drastic differences in terms of height and width. Therefore, to cover for this issue, we also assess the overlap between the given ground truth box area and estimated box area. The overlap of areas is calculated as follows:

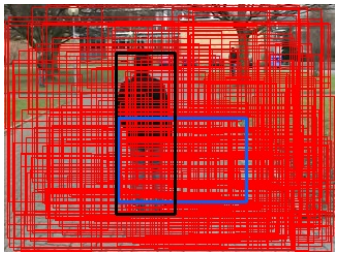
$$E_{overlap} = \frac{area(R^{EST} \cap R^{GT})}{area(R^{EST} \cup R^{GT})} \tag{14}$$

where, R^{GT} and R^{EST} are the bounding box for ground truth and state estimate, respectively.

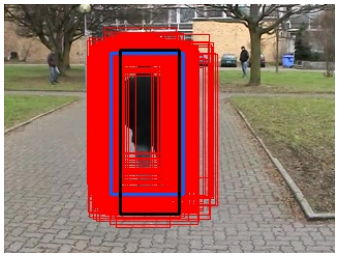
It should be noted that for the euclidean error, a lower value would correspond to a better performance. Contrarily, for area overlap, a higher value would correspond to better performance. Here, a value of 1 is tantamount to complete overlap and 0 to none.

Now, for the comparison between the methods, the implemented algorithm was run 20 times with 100 particles

(a) First Frame



(b) Particle Initialization



(c) Particles Converge

Figure 1

for each type of method. This was done to bypass the effect of randomness that is inherent in the particle filter. This allowed us to assess the average overlap and standard deviation in different parts of the video (see Figure 2). As for the overall tracking accuracy (for entire duration of video) in terms of overlap is listed in Table 1. Here it can be noted that the performance of color based method alone and Fusion method was comparable.

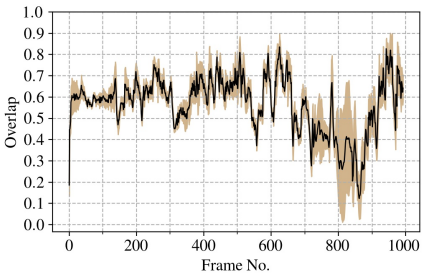
For Area Overlap

	μ	σ
Color Based	0.647	0.039
Moment Based	0.567	0.060
Fusion	0.643	0.041

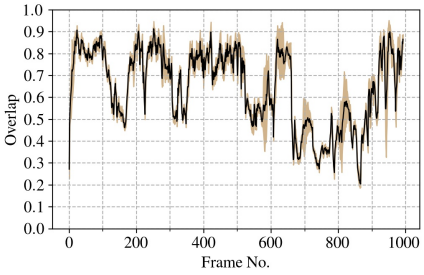
Table 1: Tracking accuracy, measured in terms of area overlap between the ground truth and estimated bounding box

The euclidean error was also tracked and its plots can be seen in Figure 3. For this, Table 2 summarizes the overall tracking accuracy w.r.t to the euclidean error. As expected, the values in Table 2, reconfirms the observation that Color histogram and Fusion method perform better than the Moment invariant method.

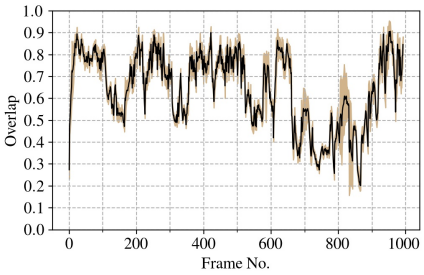
A noticeable aspect of the tracker was its relative robust-



(a) For Moment Based Method

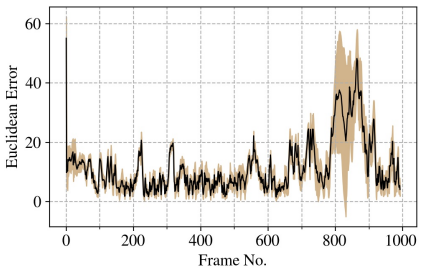


(b) For Color Based Method

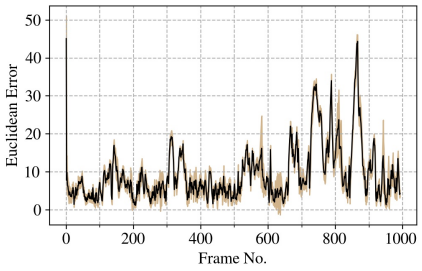


(c) For Fusion Method

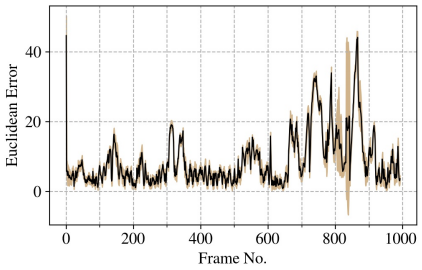
Figure 2: Overlap area evolution throughout the tracking, averaged over 20 runs. The black line is for the mean of area overlap and the shaded part represents its standard deviation over the 20 runs



(a) For Moment Based Method



(b) For Color Based Method



(c) For Fusion Method

Figure 3: Euclidean error evolution throughout the entire tracking, averaged over 20 runs. The black line is for the mean of euclidean error and the shaded part represents its standard deviation over the 20 runs

For Euclidean Error

	μ	σ
Color Based	9.427	1.780
Moment Based	11.552	3.834
Fusion	8.739	1.819

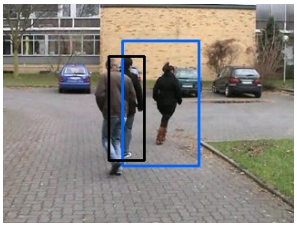
Table 2: Tracking accuracy, measured in terms of euclidean distance between ground truth center and estimated center

ness to similar looking distractors. Shown in figure 4a, although the bounding box increases in size due to the confusion caused by distractors, the overall pose estimate remains consistent with the tracked person. Moreover, from Figures 2 and 3, a clear distinctive trend could be noticed in the tracking between frame 600-900. This part of the video can be considered the most difficult, as in this part, there are a lot of occlusions, similar distractors (see figure 4b), seemingly more unstable camera and the target person turns to the right (see figure 4c). When the person turns, its profile is different from the usual back shot, so this may cause problems in measurement. These bombardment of challenges, made the estimate worse with each frame, however, the estimate does come back to the true system state eventually when things become normal. In figures 2a and 3a, a higher deviation can also be observed in the mentioned video section, as compared to remaining methods. Therefore, through just visual inspection and the over all standard deviation for 20 runs, it is safe to say that moment invariant method was found to be the least robust. While the Fusion method seems to be the best tracker, at the cost of higher computation and complexity.

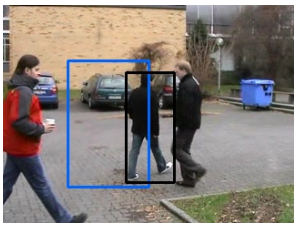
5. Conclusion

The purpose of this project has primarily been to explore the capabilities of particle filters for tracking a person in a video sequence. This entailed a comprehensive literature review, followed by selection of the most feasible, yet suitable strategy for tracking people. The task is further augmented with an objective of analyzing the tracking performance differences between the chosen Color based, Moment based and Fusion based methods. Consequently,

(a) Good estimation despite similar looking pedestrians (Frame No. 718)



(b) Deviation from target due to occlusion and other similar looking person (Frame No. 840)



(c) Target is tuned sideways, along with multiple bodies in the frame (Frame No. 882)

Figure 4

to facilitate a valid comparison and assessment, testing is done via a seemingly difficult video sequence taken from the Bobot benchmark. The test video is considered challenging due to partial and full occlusions, out of plane rotation, moving camera and similar looking distractions. From the experiments and subsequent results, it can be stated with confidence that this respective particle filter was successful in tracking the person, with the best performance being comparable to [15]. As for the respective target representation strategies, i.e. Color, Moment and Fusion method. The Fusion method provided the best tracking performance. However, the difference is almost negligible between Fusion and Color Based method, with respect to overlap area metric. For the euclidean error, the Fusion method is significantly in lead. Moreover, for both evaluation metrics, the Moment based method ranked last. In the end, a crucial takeaway from this study is the insight into particle filters, object tracking, and object detection methodologies.

References

[1] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery.

[2] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[4] Bobot dataset. `https://www.kaggle.com/kmader/videoobjecttracking`, 2022.

[5] Neil J. Gordon, David Salmond, and Craig Ewing. Bayesian state estimation for tracking and guidance using the bootstrap filter. 1993.

[6] Christopher G. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[7] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[8] Blake. A Isard. M. CONDENSATION—Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[9] Gao Junxiang, Zhou Tong, and Liu Yong. Face tracking using color histograms and moment invariants. 10 2009.

[10] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[11] Jaya S. Kulchandani and Kruti J. Dangarwala. Moving object detection: Review of recent research trends. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–5, 2015.

[12] Zhong Liu, Weihai Chen, Yuhua Zou, and Cun Hu. Regions of interest extraction based on hsv color space. In *IEEE 10th International Conference on Industrial Informatics*, pages 481–485, 2012.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004.

[14] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. A color-based particle filter. 01 2004.

[15] Mateusz Owczarek, Przemysław Barański, and Paweł Strumiłło. Pedestrian tracking in video sequences: A particle filtering approach. In *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 875–881, 2015.

[16] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[17] Yang Taekyu and Kang Sukbum. Tracking for moving object using invariant moment and particle filter. In *2008 27th Chinese Control Conference*, pages 351–354, 2008.

[18] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, Cambridge, Mass., 2005.