Multi-Pedestrian Tracking based on Kalman Filter and Particle Filter

Author: Wenjie Yin	Partner: Jiangpeng Tao
TSCRM	TSCRM
951026-8213	950227-0177
Email: yinw@kth.se	Email: jtao@kth.se

Abstract

In this project, we achieve multi-pedestrian tracking mostly using Kalman Filter and particle filter which combines the position estimation and size estimation. We first utilize ACF detector and morphological Operation as the observation model, and explore the combination of these models. We use the Hungarian algorithm for data association, calculating the cost based on the distance of positions and changes of size, then also adopting the strategy of replicating observation to cope with occlusion, taking the main moving direction into consideration to improve the accuracy of association. Finally, we compare the results of different approaches by MSE, and the experiment results shows the Kalman Filter and the particle filter with complex association strategies and enhancement can work well in the multi-pedestrian tracking problem in static environments.

1 Introduction

This project focus on how to locate and tracking multi-pedestrian in a video surveillance system. Pedestrian detection and tracking is a process of automatic detection of people in a video sequence and positioning the detection target in the video afterwards, which is an important part of computer vision. [1] In recent years, detecting moving human in a video sequence is drawing researchers' attention and relevant research work is booming because of a large range of application, such as pedestrian counting, people identification, intelligent transportation and other fields, which has extensive application prospects. [2] Even though the moving objects in a video sequence is easy to be extracted, tracking people in a video is still a challenging problem for the difference in illumination, body poses and viewing angle can cause various changes, which even lead to occlusion. The occlusion will lead to incomplete or wrong observation of pedestrians, even missing some pedestrian in the scene, which make data association in the process of tracking really difficult. [3]

In this regard, this report tends to study and analyze different multi-pedestrian tracking approaches, then try different methods to avoid disturbances and improve the results. We focused on a parametric approach, namely the Kalman Filter. There are many related tracking tasks based on this method, for example in [4], [5], [6]. We also explore a non-parametric approach for multi-pedestrian tracking, namely Particle Filter, which also has been widely used, for example in [7], [8].

In this report, we achieve the Kalman Filter and particle filter. And further improve the motion model. we adopt Hungarian algorithm for data association and eliminate the noise detection, object occlusions through several algorithms.

1.1 Contribution

During the implementation in this report, we made the following contributions:

- we adopt the morphological operation mentioned in [9] and a pre-trained upright people detector using aggregate channel feature, which is trained using the Caltech pedestrian dataset to segment the target pedestrian.
- We improve the motion model by taking the size into consideration and propose several methods to enhance the accuracy during the step of data association. [10] For instance, we replicate the observation during data association to solve the problem of occlusions, also take the orientation and size of pedestrians to solve the problem of wrong

association. [11] And we consider two previous state to eliminate the noise, improve the particle filter by consider the change of the process noise covariance matrix

- We compare the performances of the Kalman Filters, the particle filter and the enhanced filter in terms of the Mean Square Error (MSE) between the ground truth and the estimated trajectory.

1.2 Outline

Section 2 briefly introduces the related work for multi-pedestrian tracking using Kalman Filter and particle filter. Section 3 explains the theory of the method and different implementations. Section 4 shows some experimental results using different approaches, then compare and analyses these results to draw some preliminary conclusions. Section 5 draws some final conclusions and some promising new ideas.

2 Related Work

We can separate the task of multi-pedestrian tracking in a video sequence into three steps. First of all, we need to detect the possible locations of pedestrians in the video scene, then use these observations to implement tracking and produce the trajectory of pedestrian. In general, the following four kinds of widely known classical approaches can be adopted for pedestrian tracking: mean-shift methods, filtering framework methods, correlation-based template matching and motion detection-based tracking algorithms. [12] Mean shift for tracking is a non-parametric iterative method to find mode of distribution based on points to cluster object. And filtering framework methods using recursive algorithms and filters based on motion model for tracking, such as Kalman Filter, particle filter. In correlation-based template matching methods, the initial target is selected and then the representation of the target is used to locate target in a video sequence considering the correlation. The motion detection methods using background subtraction, temporal difference, background modeling, and optical flow to detect the relative motion then tracking the objects in video based on the motion.

The filtering framework methods are effective to solve pedestrian tracking and estimation. Monica et al. [9] proposed a Kalman Filter method using a linear prediction model and an observation model based on background subtraction and motion detection. Kokul et al. [13] presented a tracking approach for pedestrians in dynamic backgrounds by combining pre-trained generic person detector, online trained person-specific detector and a motion tracker with particle filter. Sahbani et al. [14] explained the implementation of modified Hungarian algorithm and Kalman Filter as the main core of multiple objects tracking.

In our project, we use filtering framework methods including the Kalman Filter and particle filter to estimate the position of the multi-pedestrians, and we use the motion detection methods like background subtraction and a pre-trained person detector in the observation model. In data association step, we use Hungarian algorithm for assignment problem to assign the most proper trajectory for each observed measurement.

3 Method

3.1 The Prediction Model

After we acquire the previous position and size of the pedestrians, we can define a motion model which model the motion of pedestrian and combine the model with previous states to make a prediction of the current state. In our model, the state includes the position, velocity, and size of the pedestrians. We define the state as a 7×1 vector shown below:

$$\mu = [x,\ y,\ u,\ v,\ w,\ h] \tag{1}$$

Where $[x, y]$ is the center coordinate of the pedestrian, and $[v_x, v_y]$ is the velocity of the pedestrian, $[w, h]$ represents the width and height of the bounding box. Then we define the motion model as following:

$$\mu_{t+1} = A_k \mu_t + B_k u_k + \varepsilon_k \tag{2}$$

Because the velocity between two frames is similar, we use a constant velocity model. And there is no external influence in the pedestrian tracking, thus both the control vector u_k and the control matrix B_k can be ignored. In this way, we can derive the motion model as

$$\mu_{t+1} = A_k \mu_t + \varepsilon_k \tag{3}$$

$$\begin{cases} x_{(t+1)} = x_t + v_{x(t)} + \varepsilon_k \\ y_{(t+1)} = y_t + v_{y(t)} + \varepsilon_k \\ u_{(t+1)} = u_{(t)} + \varepsilon_k \\ v_{(t+1)} = v_{(t)} + \varepsilon_k \\ w_{(t+1)} = w_{(t)} + \varepsilon_k \\ h_{(t+1)} = h_{(t)} + \varepsilon_k \end{cases} \tag{4}$$

A_k is the state transition matrix, μ_k is the control vector. ε_k is the process noise covariance matrix.

3.2 The Observation Model

The observation model in this project is used to detect the pedestrians in a frame of video sequence. Then similarity between the detection and prediction can be calculated in the step of data association. Here, two observation models implemented by morphological operation methods and Aggregate Channel Feature detector.

3.2.1 Morphological Operation

Morphological operation is implemented in 3 steps: [9]

First, foreground detector is used to compute foreground mask, which can segment moving objects from the background. The foreground detection also is called as background subtraction. A video sequence consists of a series of frames with respect to time which are continuously captured by camera. It's not easy to process the video directly. Here we use gaussian filter to pre-process the frame to remove the noise, which can improve the accuracy of the result. As long as the background frame is initialized, the background frame can be referred as reference frame P[B]. In our experiment we assume the first 40 frames of the video sequence as the background.

After the reference frame is defined, the moving object detection can be implemented by subtracting the current frames and the reference frame, which is done pixel by pixel of between both frames. and the foreground is the result of the subtraction.

$$\begin{gathered} P[F(t)] = P[I(t)] - P[B] \\ |P[F(t)]| - |P[F(t+1)]| > Threshold\ value \end{gathered} \tag{5}$$

Here *P[F(t)]* is the foreground video frame, *P[I(t)]* is the input frame of time t, *P[I(t+1)]* is the next input frame of time t+1 and *P[B]* is the reference background initialized by the first 40 frames. When the subtracted pixel value is greater than the threshold value, it will be represented by 1, otherwise it will be represented by 0. In this way the segmented image shows the moving object in white and the background in black.

Then using gaussian mixture model (GMM) to compute the foreground mask. The GMM will give a series of grayscale video frames, and then connecting groups of foreground pixels which are likely to correspond to moving objects. It then performs morphological opening and morphological closing on the resulting binary mask to remove noisy pixels and fill the holes in the remaining blobs. In this way such groups can be found, and their characteristics of centroid and the bounding box can be computed, see Fig. 1. The multiple moving pedestrians are detected, and the positions are shown by red boxes.

(a) The labeled video frame

(b) The foreground mask

Fig. 1: Moving objects detection using morphological operation. The figure (a) shows the characteristics of centroid and the bounding box (red box). The figure (b) shows the result after foreground detection and morphological operation.

3.2.2 Aggregate Channel Feature Detector

Aggregate Channel Feature (ACF) Detector [15] is a well-known feature used in person detection, which computes and aggregate several feature channels, including the normalized gradient magnitude, histogram of oriented gradients and LUV channels. Features in the aggregated channels are single pixel. This detector can succeed in detecting a person region and then returns a state including the upper-left coordinates of the bounding box, and the size of the bounding box in a vector. In this project, we use the detector trained using the INRIA Person dataset, namely *peopleDetectorACF*. Fig. 2 shows the result after pedestrian detection using this detector.

Fig. 2: Moving objects detection using Aggregate Channel Feature Detector, the bounding box is blue.

3.2.3 Combined Detector

The morphological operation is a motion detection method implemented by background subtraction, which is done pixel by pixel. Thus, this method is very sensitive to movement of pixels. Even the pedestrian is blurry or partially blocked, this method can still detect the pedestrian. Although the sensitivity of the morphological method is high, its accuracy is low that can be easily disturbed by the environment and noise. And may outliers will be generated easily, leading to large detection error.

Aggregate Channel Feature Detector is a kind of detector trained through a dataset using ACF, thus this detector is stable and more accurate than the morphological operation method. This detector is not easily affected by the environment changes and the pedestrians' pose. The results of coordinate and size returned by ACF detector is in high belief. However, the ACF detector's sensitivity is low, pedestrians can not be successfully detected when the pedestrian is moving in the distance that the size is small or when their bodies are partially blocked.

We can see this conclusion according to the comparison between Fig. 1 and Fig. 2. In Fig. 1, the morphological operation method can detect both pedestrian in the frame but there exists wrong detection. Fig. 2 shows ACF detector accurately detect the pedestrian but not sensitive enough.

In our implementation of observation, we combine both method to trade off between sensitivity and accuracy. Using the ACF detector as the main observation model, and regard the morphological operation method as a secondary detector. When the observed bounding box of two detectors overlap and the area of overlap more than a threshold, we taken the observation of ACF detector as the final observation. Otherwise, we merge both observations and solve the assignment problem in data association step. See in Fig. 3, both pedestrians can be detected, and the noise has been reduced.

Fig. 3: Moving objects detection using the combined detector of the ACF Detector and the morphological operation detector.

3.3 Association Method

Data association is one of the most challenging problems in estimation, we need to decide which detection belongs to which tracker, then assign each detection in the current frame to a target. In our project, we solve this optimal single-frame assignment by Hungarian algorithm. [14] The Hungarian algorithm is the method to do linking process between the identified tracker *tr* in frame *k* to the unassigned detection *dt* in frame *k+1* by calculating the extreme solution of the Euclidean distance in the assignment matrices.

We achieve this algorithm in three steps. First of all, we need to compute the cost, which is defined as matching assignment matrix D for each pair (*tr*, *dt*) of tracker *tr* and detection *dt*. The assignment matrix is illustrated below

$$D_{tr,dt} = \begin{bmatrix} d_{tr1,dt1} & \cdots & d_{tr1,dtn} \\ \vdots & \ddots & \vdots \\ d_{trn,dt1} & \cdots & d_{trn,dtn} \end{bmatrix} \tag{6}$$

with $d_{tri,dtj} = euclidean\ dis\tan ce(tri,\ dtj)$

The cost takes the change of position and the change of the size into consideration. For the cost of position, we calculate the Euclidean distance between the centroid of the predicted track and the centroid of the detection.

For the cost of size, we calculate size of the predicted size and the size of detection, mapping these size to Cartesian coordinates then calculate the Euclidean distance of the size. The final cost is the combination of both position and size with a coefficient to maintain the importance weight between these two costs.

The next step is to find the solution of the assignment problem. The solution is found by calculate the minimum point solution the assignment matrices:

$$f = Min(\{D_{tr,dt}\}) \\ \{f: \{tr_i \to dt_j\} | tr, dt \in R^2\} \tag{7}$$

In the project, we adapt Munkres' version of Hungarian algorithm to execute this step to minimize the total cost, the algorithm returns an assignment matrix containing the indices of the assigned tracks and the corresponding detections, as well as indices of the unassigned tracks and detections.

During the detection process of the video sequence, when multiple pedestrians overlap in the video, the observation model will detect fewer objects, which may lead to mistakes in data association

We adopt the strategy of replicating the observation to avoid this situation. When the number of detected objects decreased, we judge the degree of overlap in previous frame, if the degree of overlap exceeds a threshold, we believe the decrease of the observed pedestrian is caused by overlapping. Thus, we replicate the overlapped observation to make sure the number of observation in data association can be correct.

However, because the replicated detection is the same as the origin detection, then the association may be wrong in certain probability. In order to solve this problem, we take the main moving direction into consideration. If we find the direction of the overlapped pedestrians become opposite suddenly, we can exchange the indices of these pedestrians to avoid this problem.

3.4 Outlier Detection

Sometimes, because of the complex environment and unreasonable noise, the detections may be not reliable, and we refer these detections as outliers. we need to set a threshold to detect outliers and get rid of wrong matching by setting these detections unassigned and invisible. And only assign the detections below the threshold to a track. For these outliers that remain unassigned, we assume these are the start of new tracks. When they remain unassigned more than a threshold times, they will be used to create new tracks.

3.5 Kalman Filter

In this section, we use the Kalman Filter to estimate the state of multi-pedestrian. The algorithm of Kalman Filter for multi-pedestrian tracking is shown below:

Tab. 1: The KF algorithm for multi-pedestrian tracking

Kalman Filter Algorithm

1. Using detector in section 3.2 to get the initial state of the pedestrians;
2. Predict: Using the prediction model in section 3.1 to predict the state of each pedestrian in next frame:
$$\bar{\mu}_k = A_k \mu_{k-1} \\ \bar{\Sigma}_k = A_k \Sigma_{k-1} A_k^T + R_k \tag{8}$$
A_k is the transition matrix, R_k is the process noise covariance matrix;
3. Observe: using the observed states of pedestrians in the current frame as measurement z_k, which is used to correct the predicted state and covariance during update. And in the observation model, the current position is measured through the current frame, the velocity is measured through the previous three frames, so as to reduce the influence of accidental errors and noise, which can improve the accuracy and stability of speed measurement, then improve the performance of tracking.
4. Association, using method in section 3.3 to do data association. And the KF should be able to run even no measurement and only has prediction for the observation model may fail to detect in some frames. We implemented it by skip the update step for these tracks.
5. The update equation is:
$$K_k = \bar{\Sigma}_k C_k^T \left(C_k \bar{\Sigma}_k C_k^T + Q_k\right)^{-1} \\ \mu_k = \bar{\mu}_k + K_k\left(z_k - C_k \bar{\mu}_k\right) \\ \bar{\Sigma}_k = A_k \Sigma_{k-1} A_k^T + R_k \\ \Sigma_k = (I - K_k C_k)\bar{\Sigma}_k \tag{9}$$
K_k is the Kalman gain, Q_k is the measurement noise covariance matrix. Update the state of each pedestrian.
6. Iteratively do the step 2 to 4 until the track lost.

3.6 Particle Filter

In this section, we use particle filter for multi-pedestrian tracking. Particle filter is a non-parametric method, which can track multiple hypotheses at the same time and even can recover from some occlusions. The algorithm of the particle filter for multi-pedestrian tracking is shown below:

Tab. 2: The PF algorithm for multi-pedestrian tracking

Particle Filter Algorithm

1. Using the detector to get the initial states of multiple pedestrians, and using N particles randomly distributed around the initial states in a range with equal weights. the initial distribution is depending on the belief of observation model, here the observation model is accurate, thus we set the range not too large.
2. Predict: Using the prediction model in to predict the state of each particle in next frame, add a number drawn from a normal distribution to each particle for diffusion;

$$\begin{aligned}(x,\ y)_t &= (x,\ y)_{(t-1)} + (u,\ v)_{(t-1)} + \varepsilon_{(x,\ y)} \\ (u,\ v)_t &= (u,\ v)_{(t-1)} + \varepsilon_{(u,\ v)} \\ (w,\ h)_t &= (w,\ h)_{(t-1)} + \varepsilon_{(w,\ h)}\end{aligned} \tag{10}$$

   Here we improve the particle filter by change the process noise covariance matrix. The variance of position $\sigma^2_{(x,y)}$ is proportional to the size of the observed pedestrian, the variance of velocity $\sigma^2_{(u,v)}$ and the variance of size $\sigma^2_{(w,h)}$ is inversely proportional to the number of successfully tracked frames. [16] In this way, we can modify the uncertainty of prediction to a suitable value. If we track the pedestrian longer, the particles will be less spread.
3. Observe: the same as the observation step in KF, using the observed states currently as measurement.
4. Association: We apply association to each particle. And using the Hungarian algorithm in section 3.3 to implement data association. the PF also should be able to run even no measurement and only has prediction. If there only has the prediction for each particle, we skip the step of update weights and re-sampling.
5. Update weights: in order to re-sampling, we need to compute the importance weight for each particle by calculating the likelihood of each particle from the target distribution. To be specific, we calculate the distance between the predicted coordinates and the observed position. The equation is shown below:

$$w_i = N(\mu_i - z_i,\ \sigma^2) \tag{11}$$

   We assigned the new weight to each particle after normalization. If the distance between prediction and observation is small, this particle is similar with the observation, and we need to assign a high weight to this particle. And the value of σ influence the distribution of weight. If σ is higher, these particles will converge quickly, on the other hand, these particles in large distance will be assigned a relative higher weight and can survive with higher probability.
6. Re-sampling: we resample these particles using the weights calculated in previous step. Here we use systematic re-sampling to resample, because this resampling method is more stable than vanilla resampling method. Particles with high weight are more easily to survive from time t to next time.
7. Density extraction: we estimate the state of pedestrian in one frame by calculating the weighted mean value of all particles of one pedestrian. The equation is:

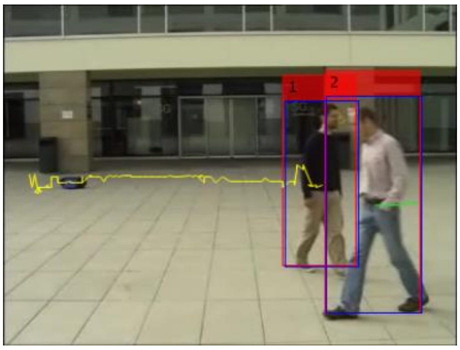
$$\mu_k = \sum_{i=1}^{N} w_i \cdot \mu_i \tag{11}$$

8. Iteratively do the step 2 to 7 until the track lost.

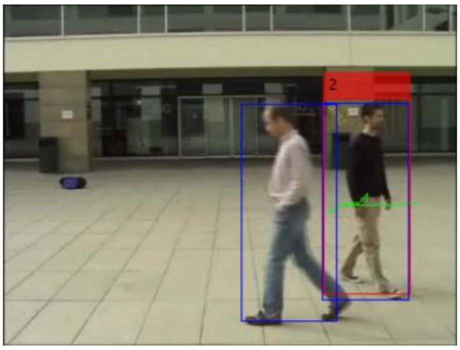
4 Experiment Result

In our implementation, we test both Kalman Filter and particle filter and the modified versions on "EPFL" data set: Multi-camera Pedestrian Videos, which is a set of multi0camera sequences that can be used to test people detection and tracking tasks.

During the step of data association, we proposal a strategy to replicate the observation to avoid the wrong data association caused by multiple pedestrians overlap in the video. The result is shown in Fig. 4. when two pedestrians overlap, the observation model can only detect one pedestrian, the numbers of observation are fewer than the existing tracks, then the data association will failed, thus the tracking of one pedestrian will lost. The right image of Fig. 4 shows the data association failed and we cannot track the second people successfully after overlapping. The link of this overlapping situation is attached: [overlap].



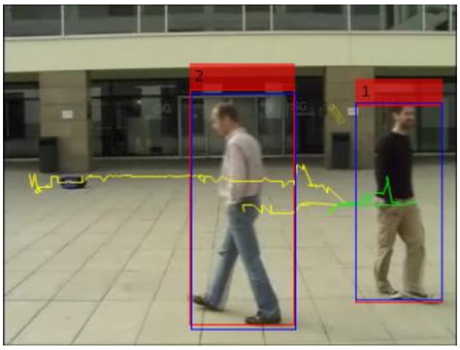
(a)



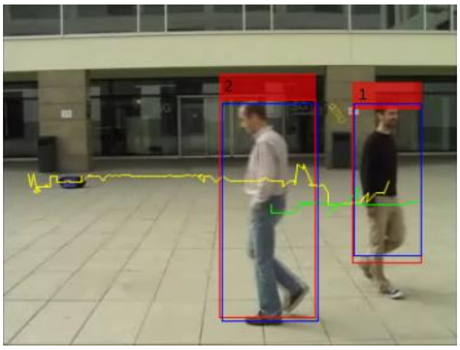
(b)

Fig. 4: the left figure is the frame before overlapping, the right figure is the frame after overlapping.

After we replicate the observation, we can track both pedestrians even they overlapped in previous frames. However, the data association also failed because the replicated detection is the same as the origin detection. Just as shown in the left image of Fig. 5, the track suddenly turns. And after we take the main orientation into consideration, this problem has been solved, just as shown in the right image of Fig. 5. The link of the results consider replicating observation methods and orientation is attached: [replicate], [orientation].



(a)



(b)

Fig. 5: the upper figure is the result of using replication, the right figure is the result considering the main direction.

But this method has some limitations. In the real situation, pedestrians may suddenly turn to the opposite direction after overlapping. In this case, this method cannot solve the data association problem. We need to consider more factors, such as the color histogram.

We use both the Kalman Filter and the particle filter to implement multi-pedestrian tracking, see Fig. 6, 7. the links of both methods' implementation are attached: [KF], [PF].

Fig. 6: multiple-pedestrian tracking using KF

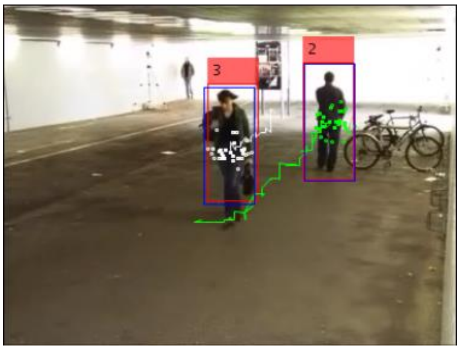


Fig. 7: multiple-pedestrian tracking using PF

The performance is evaluated in terms of the Mean Square Error (MSE) between the ground truth and the estimated trajectory, the true positions are labeled manually. We regard the distance between the true position and the estimated position as the error. We pick the center 250 frames, and calculated the MSE every 20 frames, the results are shown in Fig. 8. The tendency of the curve MSE is decreased first, then increased, there is a small peak in the center of the curve. The reason account for it the filter didn't converge at first, thus the MSE is higher, and after a few times, for EKF, the velocity and position converge to the true value, for PF, the particles converge to high probability regions, thus the MSE decreased. And when in the center of the video, two pedestrians walk towards, the are some occlusions that will influence the observation, thus the MSE increased when there exists some overlap between the two pedestrians. Then one pedestrian walks further, the detector cannot follow him, thus the observation model cannot measure it well, thus the error increased.

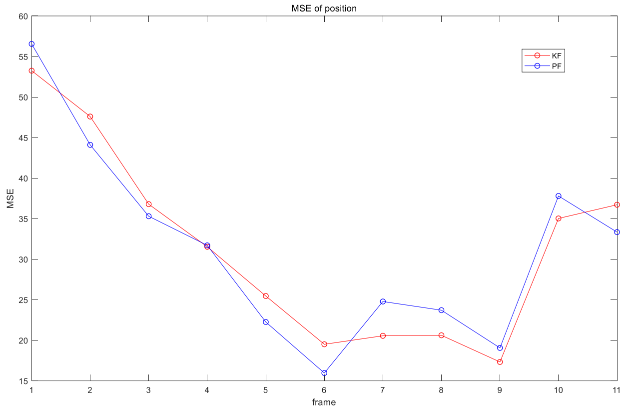


Fig. 8: the MSE of every 20 frames of the position using KF and PF.

In more complex environment, the results are shown below, see Fig. 9, 10. the particle filter and Kalman Filter can still work, but will get lost easily for these are too many noise. The link is [KF][PF].

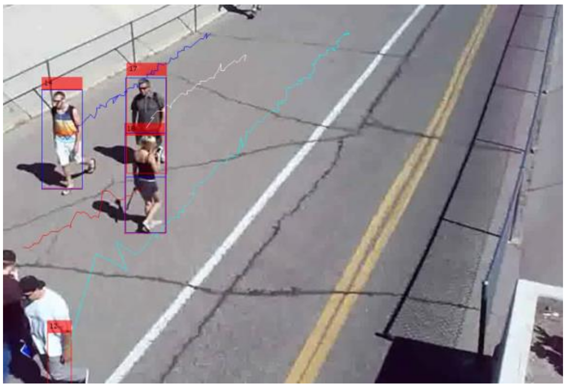


Fig. 9: multiple-pedestrian tracking using KF in complex environment.

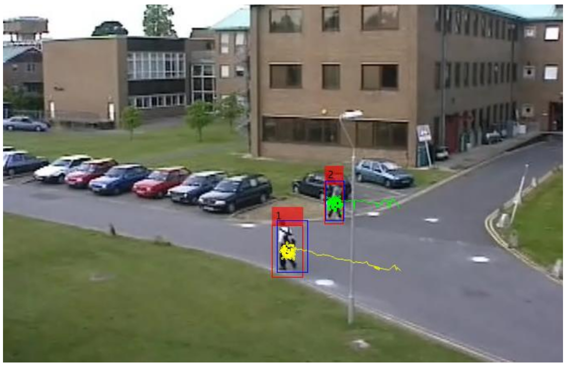


Fig. 10: multiple-pedestrian tracking using PF in complex environment.

5 Conclusion

This project using Kalman Filter and particle filter both can work well for multi-pedestrian tracking. And the strategy we proposed can solve some occlusion, such as some pedestrian overlap each other in the video. But this strategy cannot solve all situations, if the color information and pedestrian features are considered, the result will be better. The results of the experiment show that the algorithm work well when the environment is statistic, but when the environments are changing, or the camera is moving, the algorithm cannot track the pedestrian well for there are too much noise in dynamic environments. In the future, we can improve the accuracy of the detector to apply this method to dynamic environments. In this project, because the motion between frames is simple, we only use linear model, and KF is enough to cope with linear system, if we consider more movement into consideration, the result may can be improved, and we need to use other filters like EKF, UKF.

References

[1] H.Wang, Y.Fan, B. Fang, S.Dai. Generalized Linear Discriminant Analysis Based on Euclidean Norm for Gait Recognition?International Journal of Machine Learning and cyber. 2016,7(4):1-8.

[2] Z. Jiang and D. Q. Huynh, "Multiple Pedestrian Tracking From Monocular Videos in an Interacting Multiple Model Framework," in IEEE Transactions on Image Processing, vol. 27, no. 3, pp. 1361-1375, March 2018.

[3] F. Baofu, W. Zaijun, C. Jinjin, W. Hao and F. Lu, "3-Layer specific pedestrian continuous tracking framework," 2017 Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 7334-7339.

[4] G. Jati, A. A. S. Gunawan, S. W. Lestari, W. Jatmiko and M. H. Hilman, "Multi-sperm tracking using Hungarian Kalman Filter on low frame rate video," 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Malang, 2016, pp. 530-535.

[5] P. Chong and Y. H. Tay, "A novel pedestrian detection and tracking with boosted HOG classifiers and Kalman filter," 2016 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, 2016, pp. 1-5.

[6] P. Negri and D. Garayalde, "Concatenating multiple trajectories using Kalman filter for pedestrian tracking," 2014 IEEE Biennial Congress of Argentina (ARGENCON), Bariloche, 2014, pp. 364-369.

[7] R. Xu and Q. Liu, "Multi-pedestrian tracking for far-infrared pedestrian detection on-board using particle filter," 2015 IEEE International Conference on

Imaging Systems and Techniques (IST), Macau, 2015, pp. 1-5.
[8] X. Fen and G. Ming, "Pedestrian Tracking Using Particle Filter Algorithm," 2010 International Conference on Electrical and Control Engineering, Wuhan, 2010, pp. 1478-1481.
[9] V. M. Monica and K. G. J. Nigel, "Object tracking based on Kalman filter and gait feature extraction," 2017 International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2017, pp. 1-5.
[10] S. Gujjunoori, S. S. S. Reddy and K. V. Gouthaman, "Tracking and Size Estimation of Objects in Motion," 2017 International Conference on Machine Vision and Information Technology (CMVIT), Singapore, 2017, pp. 87-92.
[11] X. Li, K. Wang, W. Wang and Y. Li, "A multiple object tracking method using Kalman filter," The 2010 IEEE International Conference on Information and Automation, Harbin, 2010, pp. 1862-1866.
[12] Ali, Ahmad, et al. "Visual object tracking—classical and contemporary approaches." Frontiers of Computer Science Selected Publications from Chinese Universities 10.1(2016):167-188.
[13] T. Kokul, A. Ramanan and U. A. J. Pinidiyaarachchi, "Online multi-person tracking-by-detection method using ACF and particle filter," 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, 2015, pp. 529-536.
[14] B. Sahbani and W. Adiprawita, "Kalman filter and Iterative-Hungarian Algorithm implementation for low complexity point tracking as part of fast multiple object tracking system," 2016 6th International Conference on System Engineering and Technology (ICSET), Bandung, 2016, pp. 109-115.
[15] P.Dollar, R.Appel, S.Belongie and P.Perona, "Fast Feature Pyramids for Object Detection", Pattern Analysis and Machine Intelligence (PAMI), vol. 36, pp. 1532 - 1 545, 201 4.
[16] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier and L. V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," 2009 IEEE 12th International Conference on Computer Vision, Kyoto, 2009, pp. 1515-1522.