Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186–5140
Volume 3, Number 1, pages 38–41, January 2014

Pedestrian Tracking based on Hybrid Structure Filters

ChinLun Lai*, LiYin Lee
Communication Engineering Department
Oriental Institute of Technology
Taipei, Taiwan
fo001@mail.oit.edu.tw

***Abstract*—This paper proposed a object tracking method based on a hybrid filtering structure which includes Adaboost classifiers and particle filters to track a target automatically. The proposed algorithm first applied Adaboost classifier for object detection, filtering, and positioning of candidate targets, and then applied particle filter for confirming and tracking of the targets. With the help of Adaboost calculation for correction of the tracking results from particle filter, target missing events can be prevented efficiently. According to the experiment results, it is observed that compared with existing tracking methods, the performances of the proposed method in cases with disappeared objects, masked objects, and reappeared objects, in tracking target objects were much better.**

Keywords—object tracking; Adaboost detection;, particle filter;

I. INTRODUCTION

Object tracking techniques had been widely used in many fields such as video surveillance especially for the purpose of safety consideration [1]-[5]. In a time domain system, the target must be able to show time differences. In other words, the target has to move so that judgments can be made. In a space domain system, judgments are made based on image characteristics of the target. And usually judging methods based on characteristic information are more complex and diversified.

However, if only a time domain method is applied, the only thing that can be detected is that whether the target is moving. It is impossible to find out if this target is the target of interest. Methods like K-Means and Mean-Shift require manual settings. And object size changes are not allowed. Kalman Filter is used for linear systems. Real-world systems are usually no-linear systems, thus this method is not proper. And this method is only applicable when the scene is known and possible locations of the target objects are pre-set [6]. However, adding the Adaboost algorithm helps to solve this issue [7]. Therefore, this study proposed to combine the Adaboost structure and the Particle Filtering method to resolve the problems mentioned above for pedestrian tracking.

Adaboost classification is a method used to find targets by training different weak classifiers for the same training set, combining weak classifiers to make strong classifiers, and connecting strong classifiers in series. The Particle Filter is based on the Monte Carlo method. It shows probabilities with particle sets and can be applied to state spaces of any shapes. It is to look for a random sample in a state space and perform likelihood calculations with the Probability Density Function (*pdf*). It is a Sequential Importance Sampling (SIS), combining the two structures above to track objects. In order to prevent particle distribution from being too scattered due to large changes of Particle Filter *pdf* of the target, when applying Particle Filter, adjustments needed to be made based on the Adaboost detection results. Thus, the drawback of the particle filter tracking can be overcome.

This paper is organized as follows. In Section II, the system framework is proposed and described. The experiment results and discussions are presented in Section III, while the conclusions and suggestions for future works are included in the last section.

II. THE PROPOSED SYSTEM FRAMEWORK

To describe the proposed system framework, Figure 1 is used to illustrate the system structure and the detailed steps are described as follows:

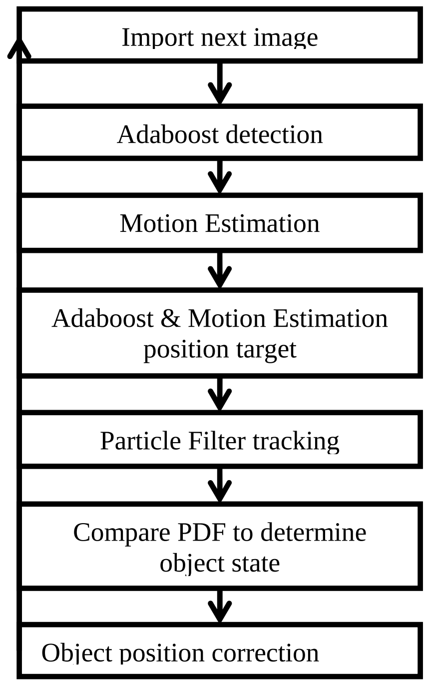


Fig.1. system framework

A. Adaboost detection

In order to improve the prediction rate and reduce the error rate, four Adaboost cascades were combined to form the initial conditions for pedestrian detection. The types of the four cascades were upper body, lower body, front side of whole body, back side of whole body. All four cascades were objects or parts of objects, resolving the issue of imprecise detection with only one cascade.

The sample included 2094 positive subjects and 1333 negative ones. The width and height were 10 and 24, using all upright rectangle feature and 45 rotated rectangle feature [8].

2 of the 4 cascades were used to detect upper body and lower body. Therefore, whole body normalization was required. An upper body includes head and shoulders. Thus, a bounding box may contain head and shoulders. And in the cascade training, the size was 22x20, with the shape being close to a square. However, human's upper bodies are rectangles. That's why normalization was required. The upper bounding boxes are normalized as follows.

$$BBox_{up}(x) = upbody(x) + \frac{upbody(w)}{4} \quad (1a)$$

$$BBox_{up}(w) = \frac{3}{4} \times upbody(w) \quad (1b)$$

$$BBox_{up}(w) = \sqrt{2} \times upbody(h) \quad (1c)$$

where $\boldsymbol{BBox_{up}}$ is the normalized bounding box and ***upbody*** is the bounding box detected using Adaboost. The 4 parameters (***x***, ***y***, ***w***, **h**), are the *x*-coordinate, *y*-coordinate, width, and height, respectively.

As for the lower body training, the size was 19x23. And whole body normalization was performed. Similarly, The lower bounding boxes can be normalized by

$$BBox_{low}(y) = lowbody(y) - \frac{lowbody(h)}{2} \quad (2a)$$

$$BBox_{down}(h) = \sqrt{2} \times downbody(h) \quad (2b)$$

where $\boldsymbol{BBox_{low}}$ is the normalized bounding box and ***lowbody*** is the bounding box detected by Adaboost.

B. Motion estimation filtering

After adopting the Adaboost detection, many possible targets were found and further filtering process, called motion estimation filtering (*MEF*), is required to remove the misjudged ones. In the *MEF* a binary threshold for 3 continuous images was calculated. Then Morphology erosion and closing were applied, with 3x3 and 5x5 matrices as masks, respectively. Then these 3 images were combined into a binary image using *OR* logical gate, in order to determine the moving range of an object. The results after applying *MEF* to the normalized bounding boxes were then used to calculate the areas of the bounding boxes. The bounding boxes of small sizes were considered as noises. The centers of those which were not noises were then obtained. After that, their average and standard deviation can be calculated by

$$Mx = \frac{1}{m}\sum_{i=1}^{m} x_i \quad (3a)$$

$$My = \frac{1}{m}\sum_{i=1}^{m} y_i \quad (3b)$$

$$MS_i = \sum_{i=1}^{m} \sqrt{(x_i - Mx)^2 + (y_i - My)^2} \quad (3c)$$

where ***m*** is the number of targets detected using Adaboost, ***Mx*** and ***My*** are average values of *x*-axis and *y*-axis respectively, while $\boldsymbol{MS_i}$ is the distance from the average of point ***i***.

Then with the average being the threshold, the bounding boxes with areas over the average were removed. Among the bounding boxes left, the one with the smallest area was considered as the most possible target object. Then, target positioning was performed with the most possible bounding box. And the positioning information, including the center and the range of the target, was sent to the Particle Filter.

C. Particle filter tracking

Once the detected object was confirmed, the particles of the Particle Filter were then initialized using the positioning information of the object to perform Particle Filter [6] tracking. And the re-sampling method adopted was the systematic method [9]. The new model was based on the average of the two previous models. To do that, the *pdf* of the target at *t*-1 was compared with that at *t* through the corresponding correlation coefficient. If the difference was too large, the target was considered disappeared or lost. In that case, Adaboost was applied again for re-positioning. In this experiment, the threshold value was set to be 0.1. Because the *pdf* of the target changed with its posture, the threshold value could not be too high.

In the whole tracking process, there were several judgments needed to be made.

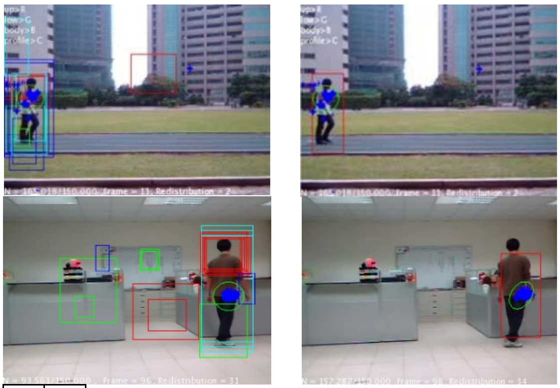
1. Target appearing: Judged by Adaboost classifier
2. Target disappearing: Judged by low model *pdf* correlation of the target
3. Whether a disappeared target is the target appeared later: Judged by model *pdf* correlation of the target higher than 0.1.

D. Adaboost & Particle Filter correction

In order to prevent particle distribution from being too scattered due to large changes of Particle Filter PDF of the target, when applying Particle Filter, adjustments needed to be made based on the Adaboost detection results. In continuous images of the target, the functions of the size and the distance were supposed to be continuous. Therefore, by referencing continuous sizes and distances from Adaboost and Particle Filter, targets of abnormal sizes were removed and targets of large distances were re-positioned according to corresponding Adaboost information while Particle Filter was re-initialized.

III. Experimental results and discussions

Two detection results with different scenes are shown in Fig. 2. It is noted that, in fig. 2(a)(c), squares with different colors represent the Adaboost detection results corresponding to different training database, where red, green, blue, and cyan squares resulted from upper bodies, lower bodies, frontal bodies, and whole pedestrians respectively. In fig. 2(b)(d), the red squares are generated from (a) and (c) after filtering, while the blue points and green ellipses represent the particles and average statuses of particles respectively.



a	b
c	d

Fig. 2. (a)(c): original image with adaboost detection results, and (b)(d): corresponding final detection result and target location of particles.

On the other hand, Fig. 3 shows the number of detection and that of error judgments for each method in scene 1 and scene 2 respectively.

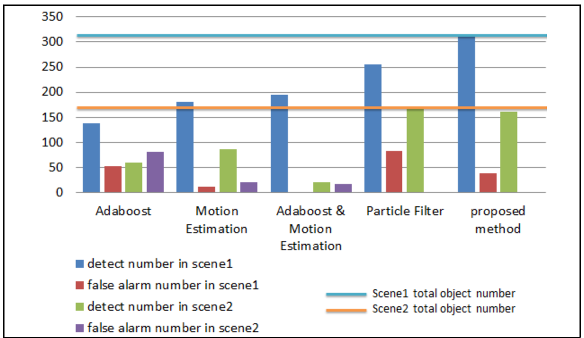


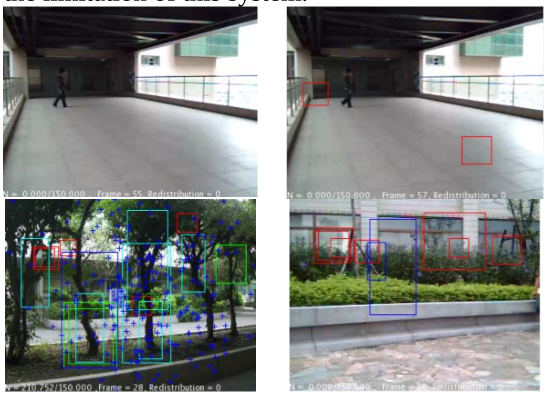
Fig. 3. Detect and false alarm number of each algorithm in test scenarios

Table 1 summarizes the statistical results of the number of detections and that of error judgments from Fig3. The efficiency differences between the method proposed by this study and the current tracking methods are revealed in Table 1. The detection rate for each algorithm was calculated by definition as: the number of successful tracking of the object in the 507 images (there were a total of 701 images from the videos for the two testing scenes, and after the ones with the target disappearing were removed, a total of 507 images were left.).

Table1. The detection rate and false alarm rate of each algorithm in test scenarios

Method	Detectio n rate	False alarm rate
Adaboost	0.4065	0.1572
Motion Estimation	0.5341	0
Adaboost & Motion Estimation	0.5756	0.0356
Particle Filter	0.7566	0.2433
proposed method	0.9139	0.1483

According to Table 1, compared with other algorithms, the detection rate of the proposed method was higher and the error judgment rate was lower. However, its performances in other experiment scenes weren't all as outstanding. For example, in a scene with complex environment and weak lights, as shown in Fig4, tracking was not possible. The reason was the high error rate and low accuracy of Adaboost, making successful positioning impossible. In the future, before Adaboost is applied, pre-processing using histogram equalization can be done to reduce issues caused by weak lights and low contrast. In addition, by increasing the number of negative subjects in the sample for cascade training, more features which do not belong to the object can be learned to break the limitation of this system.



a	b
c	d

Fig. 4. Several positioning example of the failure (a)(b) Adaboost have a low detected and a false positive rate in low light scene (c)(d) Adaboost will be difficult positioning in an environment more complex scenarios.

IV. Conclusions and Future Works

This study proposed an efficient method to track objects automatically. After detecting and positioning possible targets with Adaboost and Motion Estimation, Particle Filter started tracking with the positioning information. And adjustments were then made repeatedly based on what was detected and tracking positions. This method broke the limitation of having to specifying target positions in advance manually and also resolved issues such as object disappearing, object being masked, form changes, and object re-appearing. Therefore, this method can be applied in various scenes. This method can be widely applied in different areas with high efficiency.

Like other existing methods, the issue of bad performances in detection and tracking in a complex environment still exists with the object tracking method proposed by this study. In the future, we will add a pre-processing step of image enhancement before target detection and increase the number of negative subjects in the sample for cascade training, to reduce errors and resolve the issue above, so that this system can be applied more widely with higher efficiency.

REFERENCES

[1] Stinson, Leo F. “An Experimental Video Tracking System for Application in Air Traffic Control”, *Aerospace and Navigational Electronics, IRE*, vol. ANE-9, pp.200-210, 1962.

[2] Saxena, V., Grover, S., and Joshi, S. “A real time face tracking system using rank deficient face detection and motion estimation”, *Cybernetic Intelligent Systems*, pp.1-6, 2008.

[3] Luping Ji, Xiaorong Pu, Ming Ye, and Peng Qihua. “Face extraction from video sequences by K-means clustering and fusion”, *Computer Science and Automation Engineering (CSAE)*, vol. 1, pp.358-361, 2011.

[4] Zhidong Li, Jing Chen, and Schraudolph, N.N. “An improved mean-shift tracker with kernel prediction and scale optimisation targeting for low-frame-rate video tracking”, *Pattern Recognition, ICPR 2008*, pp.1-4, 2008.

[5] Xangdong Xie, Sudhakar R., and Hanqi Zhuang, “Real-time eye feature tracking from a video image sequence using Kalman filter”, *Systems, Man and Cybernetics*, vol. 25 , pp.1568-1577, 1995.

[6] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. “An Adaptive Color-Based Particle Filter”, *Image and Vision Computing*, pp.1-22, 2002.

[7] Viola, P., and Jones, M. “Robust Real-Time Face Detection”, *Computer Vision, ICCV 2001*. vol. 2. pp.137-154, 2001.

[8] R. Lienhart , and J. Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection”, *Image Processing*, vol. 1., pp.I-900-I-903, 2002.

[9] Gordon N, Salmond D. J, and Smith A. F. M., “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”. *IEE Proceedings F Radar and Signal Processing,* pp.107-113, 1993.