An Improved Particle Filter Approach for Real-time Pedestrian Tracking in Surveillance Video

Yaowen Guan, Xiaoou Chen, Yuqian Wu, Deshun Yang
Institute of Computer Science and Technology
Peking University
Beijing, P.R.China
e-mail: {guanyw, chenxiaoou, wuyuqian, yangdeshun}@pku.edu.cn

***Abstract*—This paper presents a method for pedestrian tracking in surveillance video, and the method is based on an improved particle filter. In our algorithm, the dynamics is modeled as a second-order autoregressive process. And for the observation model, color histogram features are used for likelihood measure. The proposed color histogram method is operated on a sub-region of the target region and we explore how the background subtraction process affects the color histogram model. We further adopt rectangle filters and pixel-difference cues in the observation model to overcome the limitation of individual cue. Experiments show that the method yields better tracking performance with the improved observation model.**

Keywords-Pedestrian tracking, particle filter, surveillance video

I. INTRODUCTION

Recently, the amount of accessible surveillance video recording is increasing rapidly, especially in public areas such as roads, airports and banks. Intelligent visual surveillance (IVS) technology in computer vision has become a hotspot and has aroused the interest of worldwide researchers. Object tracking plays an important role in IVS. It is based on object detection and provides data for further analysis and interpretation of object behaviors to understand the visual events of the scene.

Most of the tracking approaches broadly fall into one of two categories. The first category consists of deterministic methods that localize the tracked object by iteratively searching for a region which maximizes the similarity measure to the target region, e.g. Mean Shift [1]. These methods are computationally efficient. However, the iterative search may converge to a local maximum. The second category is stochastic methods based on recursive Bayesian filter theory. These methods localize the object to be tracked by propagating and updating the posterior probability density function (pdf) as new information is received. They are able to achieve more robustness to the local maximum. Commonly used stochastic tracking methods are Extended Kalman Filter which can be applied to the nonlinear-Gaussian estimation situations and particle filter which has been proven to be a robust algorithm to deal with the nonlinear, non-Gaussian problemscitegordon1993novel. Due to most of the tracking problems in surveillance videos belong to the nonlinear, non-Gaussian situations, particle filter has been widely used in video surveillance domain.

Particle filter methods simultaneously track multiple hypotheses and recursively approximate the posterior pdf in the state space with a set of random sampled particles. Proper dynamic model and observation model have a strong influence on the tracking performance, especially the definition of the observation model because it determines the particle weights and the following resampling step. The histogram-based observation models [3] are often preferred because of their simplicity and robustness to scaling and rotation. However, the similarity measure based on color histograms (e.g., Bhattacharyya coefficient) is often not discriminative to the change of position and illumination. Usually fusion of multiple cues like spatial information, edge information [4], and texture information [5] is used for likelihood measure to resist the drawback of single cue.

This paper adopts particle filter for pedestrian tracking in surveillance video. For model construction, a second-order AR process is used as motion model. And color histogram features, rectangle filter features, and pixel difference features are fused to the observation model calculation. The pixel cues and rectangle filter features are adopted to our method for robust tracking as single color cue does not work in all cases. The impact of background subtraction is considered in observation model construction. In algorithm implementation, we use weight buffer and integral images techniques to speed up calculations. The cues added to our algorithm are computational efficiency, which can meet real-time requirements.

In the remainder of this paper, Section II and III introduce the motion model and observation model of our method before Section IV provides algorithm implementation. Finally it comes to the experimental results and conclusion.

II. STATE SPACE AND MOTION MODEL

State Space For there are many occlusions in surveillance video, we apply head-shoulder tracking instead of human body tracking. And the head-shoulder region is simplified as a rectangle box. The state $\mathbf{x} = \{x, y\}$ consists of the 2D image position (x, y). The scale change of target is not considered here, so the size of the box is not included in the state space representation of particles.

Published by Atlantis Press, Paris, France.

Motion Model To propagate the particles, we just use a simple second-order AR process model. The current state is estimated as follows:

$$(x, y)_t = (x, y)_{t-1} + a \cdot ((x, y)_t - (x, y)_{t-1}) + e(p, q). \quad (1)$$

Where the coefficient a decreases the motion estimation's weight, for the motion estimation is just a rough approximation and over-trust it will lead to particles jitter, which decrease the tracking precision. The process noise $e(p, q)$ for each state variable is independently drawn from zero-mean normal distributions, where *p* and *q* are constants. Experiments show that it yields good tracking performance when a is set to 0.5.

III. OBSERVATION MODEL

To compute the weight $w(\mathbf{x}_t)$ for a particle with the candidate state $\mathbf{x}_t$ of a tracker in frame *t*, our algorithm estimates the likelihood of a particle to the reference state. We use the initial target region as reference. The region is human-annotated. We fuse different cues, namely color histogram features $F_C(\mathbf{x}_t)$, rectangle filter features $F_R(\mathbf{x}_t)$ and pixel change features $F_P(\mathbf{x}_t)$ to construct the observation model:

$$w(\mathbf{x}_t) \propto F_C(\mathbf{x}_t) \cdot F_P(\mathbf{x}_t) \cdot F_R(\mathbf{x}_t). \quad (2)$$

Each model is described below in detail.

A. Color Histogram Model

The color histogram-based models are popular because of their simplicity and robustness to scaling and rotation. We adopt the color histogram algorithm proposed by P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. [3]. Our method is different from Pérez's in that: 1) only a sub-region is used to calculate histogram in our method (the inner box in Fig. 1 is an example). 2) In our method, we make the best use of background subtraction process to improve the tracking performance: when the video background is simple and the background subtraction process works well, histogram is calculated on the foreground frame with background subtraction; otherwise histogram is calculated on the original frame. We compare these two strategies through experiments in Section V. The reason for using sub-region and taking background subtraction into consideration is that the target region to be tracked may involve big background changes which will largely influence histogram calculation. The color histogram likelihood model is described below in detail.

A histogram in our system is calculated in the Hue-Saturation-Value (HSV) color space with 2D H-S histogram combined with 1D V histogram. This method decouples the intensity (i.e. value) from color (i.e. hue and saturation), and suppress the sensitivity to illumination effects. An HSV histogram is composed of $N = N_h \cdot N_s + N_v$ bins. We denote $H(\mathbf{x}_t) = \{h(n; \mathbf{x}_t)\}_{n=1,\ldots,N}$ as the normalized color distribution of state $\mathbf{x}$ at frame *t*, where *n* is the bin index. We apply the Bhattacharyya similarity coefficient to define a distance *E* on the reference color model and a candidate color model. The formulation is given by [3].

$$E_c[H(\mathbf{x}_0), H(\mathbf{x}_t)] = \left[1 - \sum_{i=1}^{N} \sqrt{h(i; \mathbf{x}_0) \cdot h(i; \mathbf{x}_t)}\right]^{1/2}. \quad (3)$$

And then the color likelihood function $F_C(\mathbf{x}_t)$ is given by:

$$F_C(\mathbf{x}_t) = \exp\{-kE_c^2[H(\mathbf{x}_0), H(\mathbf{x}_t)]\}. \quad (4)$$

Where *k* is set to 20, which is suggested in [3]. Also we set the size of bins $N_h, N_s,$ and N_v to 10.

B. Rectangle Filter Model

P. Viola, M.J. Jones, and D. Snow proposed rectangle filters and applied them to pedestrian detection [6]. Two rectangle filters are selected according to the characteristics of the targets to be tracked in our observation model. Our method is different from Viola's in that rectangle filters are applied to frames with background subtraction in our algorithm. The selected rectangle filters are shown in Fig. 2. The feature value $f_i(t)$ of the *i*-th ($i \in \{(a), (b)\}$) rectangle filter operators on the target region $\mathbf{x}_t$ is defined as:

$$f_i(\mathbf{x}_t) = \frac{|S_{i,b}(\mathbf{x}_t) - S_{i,w}(\mathbf{x}_t)|}{S_{i,b}(\mathbf{x}_t)}. \quad (5)$$

Where $S_{i,b}(\mathbf{x}_t)$ and $S_{i,w}(\mathbf{x}_t)$ denotes the sum of foreground pixels in the black region and white region of the *i*-th rectangle filter operators on the target region $\mathbf{x}_t$, respectively. Then the distance measure of the *i*-th rectangle filter $E_{r,i}$ between the reference region $\mathbf{x}_0$ and the candidate region $\mathbf{x}_t$ is given by:

$$E_{r,i}(\mathbf{x}_0, \mathbf{x}_t) = \frac{min\{f_i(\mathbf{x}_t), f_i(\mathbf{x}_0)\}}{max\{f_i(\mathbf{x}_t), f_i(\mathbf{x}_0)\}}. \quad (6)$$

The likelihood function of each rectangle filter $F_{R,i}$ is defined as:

$$F_{R,i}(\mathbf{x}_t) = \begin{cases} 1 & 0 < E_{r,i}(\mathbf{x}_0, \mathbf{x}_t) < D_1 \\ \exp(-k_r D_1^2) & D_1 \le E_{r,i}(\mathbf{x}_0, \mathbf{x}_t) \le D_2 \\ \exp(-k_r D_2^2) & otherwise \end{cases}. \quad (7)$$

Where k_r, D_1 and D_2 are constants that set experimentally. Unlike the color likelihood function in (4) which is a continuous function, this measure is a piecewise one.

Surprisingly, this measure works better. The reason may be that the piecewise measure can be less sensitive to the noise generated by background subtraction, the change of shape, etc. The likelihood function here is also applied to the pixel difference model describes below.

The reason these two rectangle filters are selected is that they keep the particles from drifting. On the one hand, the up-down filter (Fig. 2(a)) is used to keep the particles from drifting in the vertical direction, because the white rectangle always covers a person's head with many background regions and the black rectangle always covers a person's upper body with few background regions when this filter is applied to the target. On the other hand, the left-middle-right filter (Fig. 2(b)) is used to keep the particles from drifting in the horizontal direction. Because the white rectangle always covers few foreground regions but many background regions and the black rectangle always covers few background regions when this filter is applied to the target. Furthermore, the shape of the target changes little in the scene which keeps the feature value *f* from big changes, so that these filters are useful.

C. Pixel Difference Model

Two pixel based models are considered to deal with the loss of spatial information of color histogram model. The first model is defined by the relative difference L1 norm of the reference target region $\mathbf{x}_0$ minus the candidate region $\mathbf{x}_t$ on the quantized foreground frame. The distance measure E is given by:

$$E_d(\mathbf{x}_0, \mathbf{x}_t) = \frac{\sum_{x,y,c} \left| f\left[I_{\mathbf{x}_0}(x,y,c)\right] - f\left[I_{\mathbf{x}_t}(x,y,c)\right] \right|}{\sum_{x,y,c} f\left[I_{\mathbf{x}_0}(x,y,c)\right]} . \quad (8)$$

Where $I_{\mathbf{x}_t}(x,y,c)$ denotes the pixel value of channel c of foreground frame at position (x,y) in frame t ($c \in \{R,G,B\}$); $f(I)$ is the quantization function. $f(I)$ is applied to each RGB channel of a foreground frame to get the quantized foreground frame. The quantization function is shown in Fig. 3(a). Unlike the usual quantization method, the pixel value smaller than 64 and bigger than 192 are set to 255 and others are set to 128 in our method. This quantization method may make the difference L1 norm larger than the usual quantization method, because the background in the foreground image is black and many pedestrian's clothes and hair are almost black, there will be no difference between the background and the pedestrian if the pixel value close to 0 is set to 0. Fig. 3(b) is an example of the quantization result. Fig. (I) is the original foreground image. Fig. (II) is background subtraction result. Fig. (III) is quantized foreground image. We can see that the quantized image contains some color information: different parts have different colors. The result of (III) is better than (II). Because in Fig.(II), the whole foreground has the same color. And the result of (III) is also better than (I). Because the difference L1 norm applied to original foreground images is sensitive to the change of position, shape and illumination. In addition, experiment confirms this quantization method gets better tracking performance than the others.

The second pixel difference model is the foreground pixels number change ratio between the reference target region and the candidate target region.

Figure 1. The region for tracking (Outer box) and the sub-region for histogram calculation (Inner box).

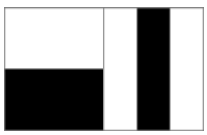


Figure 2. The selected rectangle filters in our method. The up-down rectangle filter (left); and the left-middle-right rectangle filter (right).

IV. ALGORITHM IMPLEMENTATION

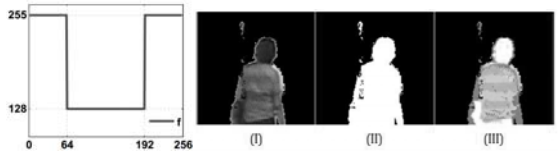
In our algorithm, the state transition density is used as importance distribution to approximate the probability density function. We use the improved particle filter for pedestrian tracking in surveillance video with the motion model and observation model described in the previous sections. The target state is estimated using the mean weighted particles. And our algorithm also re-samples the particles to avoid degeneration of particles as other particle filters do.

We use the improved adaptive Gaussian mixture background subtraction algorithm proposed by Zivkovic [7] to get the foreground images. We select a relative high update rate of the background model, thus the system will be ableto quickly respond to the actual background dynamic, so that more background noise will be detected and removed. Integral image and particles weight cache are adopted to the calculation of the observation model. These two techniques largely decrease the computation time. The entire system is implemented in C++, without taking advantage of GPU processing. On a PC with an Intel Core(TM) 3.09GHz and 3.23G of memory, we achieve processing times of 23.7 frames per second on our date sets.

V. EXPERIMENTS AND DISCUSSION

A. Data Sets

Our experiments are carried out on three common used data sets: PETS'09 S2.L1[8,9] (16 sequences from View001 and View 005),CAVIAR(http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/,15 sequences from 4 videos of Shopping Centre CORRIDOR VIEW),i-Lids AB[10](13 sequences from video Medium). The annotated sequences are taken



(a) (b)

Figure 3. The quantization function (a) and an example of the quantization result(b).

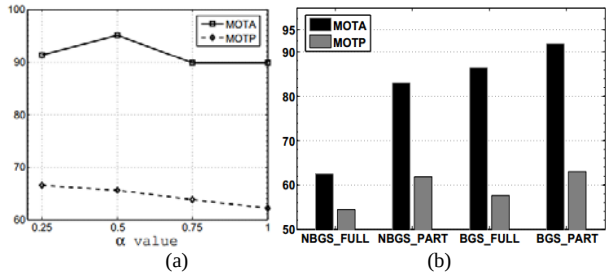
from static cameras, and they vary with respect to viewpoint, type of movement, and amount of occlusion. We use the CLEAR MOT metrics [11] to evaluate the tracking performance. This returns a precision score MOTP (intersection over union of bounding boxes) and an accuracy score MOTA.

B. Setting the a value

We performed experiments with different a values to show how the a value influences the motion model and the final tracking performance. Fig. 4(a) shows different results with different a values. We can see that the tracking precision decreases when a value increases. Because bigger a value leads to larger motion estimation, which disperses the sampled particles, and thus cause tracking precision decreases. We can also see that the tracking accuracy achieves a peak when a value is 0.5. The reason maybe that it is not easy to track high speed object when a is small, and it is easy to track to other similar object near the target when a is big. We will use a equals to 0.5 in the following experiments, for this is a good selection in terms of MOTP and MOTA.

C. The Influence of Color Histogram Methods

We performed experiments on PETS'09 data set with different color histogram methods described in Section III. In Figure 4(b) we plot the tracking performance of these methods. We can see that the method with a sub-region (PART) outperforms the original one with full region (FULL); the approach with background subtraction (BGS) is better than the original one without background subtraction (NBGS). The reason why the PART and BGS methods are useful is that they all decrease background noise in some degree. On the one hand, the PART-method decreases background noise by removing fixed regions with many background pixels. This method does not dependon other intermediate results, so that the tracking result is stable and rather better than the FULL-method. In the end, we use the PART-method to calculate color histograms in our observation model. On the other hand, the BGS-method decreases backgroundnoise by running background subtraction procedure. This approach depends largely on the data sets and the background subtraction algorithm. In our experiments, the BGS-method is better than the NBGS-method on PETS'09 data set. However, the BGS-method is worse than the NBGS-method on the other two data sets. In the end, we use different strategies in different situations.



(a) (b)

Figure 4. (a) The influence of a value over the tracking performance; (b) Tracking performance of different color histogram methods. BGS denotes color histogram calculated on foreground image with background subtraction. NBGS denotes color histogram calculated without background subtraction. PART and FULL represent color histogram features are applied to a sub-region and the full region of the target

D. Tracking Performance of Different Observation Models

We demonstrate the influence of different observation models by evaluating the tracking performance on all three data sets. The result shows in Table 1. where B is the baseline method. It is a standard particle filter with NBGS_FULL color histogram model. C, P, R represent Color histogram model, Pixel difference model, Rectangle filter model respectively. C' represents calculate color histogram on original frame without background subtraction, while C denotes calculate color histogram on foreground frame with background subtraction.

TABLE 1. THE TRACKING PERFORMANCE OF DIFFERENT OBSERVATION MODELS(UNIT: %)

Model	PETS'09		CAVIAR		i-Lids AB	
	MOTA	***MOTP***	***MOTA***	***MOTP***	***MOTA***	***MOTP***
B	62.43	54.48	41.48	35.53	45.42	39.81
C'	83.00	61.79	70.93	44.82	79.90	49.97
C'PR	90.06	**67.55**	**87.86**	**53.66**	**87.28**	**51.74**
C	91.85	63.05	71.31	53.66	67.68	49.05
CPR	**95.03**	65.57	86.89	52.48	69.85	50.32

PETS'09 Table 1 shows that in PETS'09 data set the color model with background subtraction outperforms the model without background subtraction especially in terms of accuracy. Because the background is simple and the background subtraction process works well in this data set. In addition, the observation models with pixel different features and rectangle filter features outperform the ones without. Fig. 5 demonstrates how our observation model improves the tracking performance. We show the tracking result of one person for frames 305, 325 and 335 of the PETS'09 sequences. From the results, we see that the C model suffers tracking failure in the last frame while the CPR model works well. Color cues works poorly because the clothes color of the person to be tracked is almost black which causes the particles to drift downwardly, thus the filter failed to track in the last frame.

CAVIAR and i-Lids AB Table 1 shows that in CAVIAR and i-Lids AB, the C'PR model outperforms the

Figure 5. Tracking result of C (top) and CPR (bottom)

baseline method, the C model and the CPR model. The background subtraction process fails to improve the color histogram model due to the elevated camera viewpoint; the persons occlude each other frequently which cause the background subtraction algorithm works poorly.

Fig. 6 is an example explains how the pixel cues and rectangle filters improve the tracking performance when background subtraction is not incorporated to color histogram model. We show the tracking result of one person for frames 480, 600 and 720 of the CAVIAR's TwoEnterShop1cor sequences. Without background subtraction, the C model tracks to the pillar in the last two frames because the color of the person's clothes is very similar to the pillar's. The interference of the pillar is removed when taking pixel cues and rectangle filters into consideration, because the likelihoods between the reference region and the candidate regions in these two cues are small since the pillar belongs to background regions. Thus our algorithm works better.

In addition, we can use from Table. 1, the tracking precision is smaller on these two data sets than the PETS'09. The reason is as follows. Due to the elevated camera viewpoint, the size of the pedestrians changes a lot. The annotated data considers these changes but the tracking region in our system is fixed, thus the MOTP score is small.

VI. CONCLUSION

In this paper, we propose an improved particle filter for pedestrian tracking in surveillance video. A second order AR process is used as motion model. Color histogram features, rectangle filter features and pixel difference features are adopted in observation model. We explore how the background subtraction affects the color histogram model. The experiments have shown that background subtraction largely reduced the background interference when there is clear distinct between foreground and background. However, background subtraction will become a big disturbance if the background subtraction procedure works poorly when the scene is complicated. In this end, we use different strategies to calculate the color histogram for different videos. The rectangle filters and pixel difference cues resist the drawbacks of single cues. At last, it has shown that the proposed method yields better tracking performance with the improved observation model.

Figure 6. Tracking result of C' (top) and C'PR (bottom)

REFERENCES

[1] D. Comaniciu, V. Ramesh, and P. Meer, ''Real-time tracking of non-rigid objects using mean shift," in Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. IEEE, 2000, vol. 2, pp. 142–149.

[2] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, ''Novel approach to nonlinear/non-gaussian bayesian state estimation," in Radar and Signal Processing, IEE Proceedings F. IET, 1993, vol. 140, pp. 107–113.

[3] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, ''Color-based probabilistic tracking," Computer Vision—ECCV 2002, pp. 661–675, 2002.

[4] H. Wang, D. Suter, K. Schindler, and C. Shen, ''Adaptive object tracking based on an effective appearance filter," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 9, pp. 1661–1667, 2007.

[5] Z. Zhao, S. Yu, X. Wu, C. Wang, and Y. Xu, ''A multi-target tracking algorithm using texture for real-time surveillance," in Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on. IEEE, 2009, pp. 2150–2155.

[6] P. Viola, M.J. Jones, and D. Snow, ''Detecting pedestrians using patterns of motion and appearance," International Journal of Computer Vision, vol. 63, no. 2, pp. 153–161, 2005.

[7] Z. Zivkovic, ''Improved adaptive gaussian mixture model for background subtraction," in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. IEEE, 2004, vol. 2, pp. 28–31.

[8] J. Ferryman, ''Eleventh ieee international workshop on performance evaluation of tracking and surveillance," in Proc. IEEE Int, 2009.

[9] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, ''Online multiperson tracking-by-detection from a single, uncalibrated camera," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 33, no. 9, pp. 1820–1833, 2011.

[10] C. Huang, B. Wu, and R. Nevatia, ''Robust object tracking by hierarchical association of detection responses," Computer Vision–ECCV 2008, pp. 788–801, 2008.

[11] B. Keni and S. Rainer, ''Evaluating multiple object tracking performance: the clear mot metrics," EURASIP Journal on Image and Video Processing, vol. 2008, 2008.