DENSITY PROPAGATION BASED PARTICLE FILTER ALGORITHM FOR VIDEO OBJECT TRACKING

Zexin Lin
Netease Information Technology Co., LTD
No.16, TangShi Road, TianHe Distract, GuangZhou City, GuangDong Pro., China
Email: jaylin.ca@hotmail.com

ABSTRACT

These Video object tracking is an important topic in multimedia technologies. Particle filtering has proven very successful for non-linear and non-Gaussian estimation problems. In this paper, we proposed a novel approach for video object tracking, named by Density Propagation based Particle Filter (DP-PF). Our approach exploits color histogram to capture the features from object in the video, integrates density propagation algorithm for particle initialization together with resample technique into particle filtering and uses Bhattacharyya distance as a similarity measure for the model. Experimental results of applying the density propagation technique show improvement in tracking and robustness in recovering from partial or complete occlusions. The superior real-time performance and higher discriminative power is also demonstrated by comparison with Data-Driven Adaption based Particle Filter.

KEYWORDS

Computer version, video object tracking, particle filter, color histogram and ensity propagation.

1 INTRODUCTION

Object tracking establishes the correspondences of the target of interest between successive frames, which is a fundamental research problem in video analysis and is required by many real-world applications such as human-computer interaction, video surveillance and traffic intelligence. In recent years, video object tracking experienced a steady advance in both theory and practice. Although many tracking tasks can be successfully handled by the current tracking methods, real situations in practice, such as long duration tracking in unconstrained environment, still pose enormous challenges to these techniques. One common challenge arisen from these real situations is the nonstationary changes of the visual appearances of the target due to partially or completely occlusion, view changes, and shape deformation. Such appearance changes can form a serious impact on the stability and accuracy of object tracking. Moreover, if we relax the constraint about the cooperativeness of the individual and the controllability of the acquisition environment, the challenge becomes even more demanding.

2 RELATED WORKS

Existing video object tracking methods can be broadly classified into four categories: trajectory prediction based [1, 2], density shift based [3~7], multi-algorithm combination based [8, 9], and Bayesian filter based [11, 12]. Trajectory prediction based approaches employ motion information about the target, e.g., position, acceleration, and velocity to predict its position in the next frame. Kalman filter [1], which provides the optimal solution for linear dynamic systems with Gaussian noise, is a classical estimation method. However, the linear and Gaussian assumption does not hold in many real-world problems, especially if there exists camera movement. Density shift based approaches, including mean shift tracking algorithm [3] and all its extensions [4~7], use the color distribution as tracking cue. Tracking is accomplished by iteratively finding the local minima of the distance measure functions using the the mean shift algorithm. These approaches become popular due to their simplicity and robustness. However, the mean shift algorithms using histograms have several serious defects, such as its quadratic calculation in the number of samples and low discriminative power in higher dimension. Multi-algorithm combination based approaches [8~9], such as combination of mean shift algorithm, SIFT descriptor [10] and Kalman filter, which has been proved more effective and accurate, overcomes the limitations of a single method. However, they also suffer from the unacceptable computational cost for real-time applications.

To remedy these difficulties, Bayesian filter based approaches, which essentially treat the video object tracking problem as a sequence estimation problem [11], have been proposed. In the Bayesian approach, a recursive filter is achieved using Bayes theorem, which is the mechanism for updating knowledge about the target state in the light of extra information from new data.

Particle filtering [12] is a popular approach to apply a recursive Bayesian filter based on sequential Monte Carlo (MC) technique [13]. Over the past decades, this sequential MC (SMC) based approach is also known as sequential importance sampling (SIS) algorithm [11]. It is a technique for implementing a recursive Bayesian filter by MC simulations. The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, this MC characterization becomes an equivalent representation to the usual functional description of the posterior pdf, and the SIS filter approaches the optimal Bayesian estimate.

However, it frequently suffers from the degeneracy problem, where after a few iterations, all but one particle will have negligible weight. To address this problem, Gordon et al. [11] propose to use resampling whenever a significant degeneracy is observed. The condensation algorithm, proposed by Isard et al. [14], and bootstrap filtering, proposed by Gordon et al. [15], further develops this idea.

Although the resampling step reduces the effects of the degeneracy problem, it leads to a loss of diversity among the particles as the resultant sample will contain many repeated points. This problem, which is known as sample impoverishment, is severe in the case of small process noise [12]. One effective approach is to use MCMC sampling [16].
There are various versions of other related particle filters proposed in the literature [12], such as sampling importance resampling (SIR) filter, auxiliary sampling importance resampling (ASIR) filter, regularized particle filter (RPF). All of them can be regarded as special cases of the general SIS algorithm. These special cases can be derived from the SIS algorithm by an appropriate choice of importance sampling density of the resampling step. All these extensions, however, cannot yet give a perfect and complete solution to the robust real-time tracking of non-rigid objects in the video.

In order to achieve better robustness of particle filtering against partial occlusion, Nummiaro et al. [17] uses a particle filter with color-based image features. A target is tracked with a particle filter by comparing its histogram with the histograms of the sample positions using the Bhattacharyya distance. But when objects with similar color are approaching, it will be quite difficult to discriminate between the true target and other objects, and thus lead to target missing, and false detection. Yang et al. [18] introduce the concept of Data-Driven Adaption to resort to the data-driven constraints to select positive and negative data from current image observations for the appearance model adaptation. It provides a robust tracking framework, but its computational cost is often unacceptable. Moreover, the target cannot be recognized again once complete occlusions occur. In this article, we name the algorithm which integrates the DA concept into particle framework as DA-PF.

3 OUR APPROACH

3.1 Processing Overview

According to the analysis above, there is great need for an object tracking algorithm that can handle event variations with better robustness and real-time performance.

In this paper, we present a new approach for video object tracking: Density Propagation based Particle Filter (DP-PF), an integration of density propagation technique into particle filtering. The proposed approach provides a robust and efficient solution to the traditional problems such as illumination changes, occlusions and real-time tracking. Firstly, color histograms are applied as target models as they achieve robustness against non-rigidity, rotation and partial occlusion. Secondly, density propagation algorithm together with resample technique is employed in particle filtering. Thirdly, Bhattacharyya coefficient is calculated to measure the similarity between the target model and the model of the hypotheses.

Fig. 1 shows schematically our approach to track the object from video sequences. Similar to other object tracking approaches, our tracking framework contains several major steps: (1) target modeling using color histograms; (2) initializing particles based on density propagation; (3) calculating Bhattacharyya coefficient to measure the similarity; (4) and particle resampling. We describe in detail each step in the following.

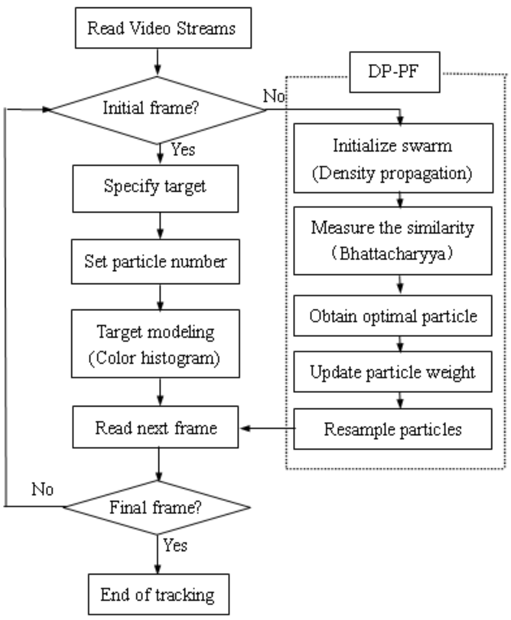

Fig.1 Processing overview

3.2 Target Modeling

Features extraction and target model construction play a crucial role in video object tracking. In this paper, we want to exploits the color-based image features, which are commonly used to represent the features of a moving target.

Color histograms are flexible constructs that can be built from images in various color spaces, whether RGB, HSV or any other color space of any dimension. In our experiments, the histograms are typically calculated in the RGB space using 8× 8× 8 bins. In the 24-bit RGB representation of the true color, color value for each pixel is expressed as an RGB triplet (r, g, b), each component value of which is stored as an integer number in the range of 0 to 255. So we adopt following formulations to reduce the original 2563 levels:

$$r = r/32 \tag{1}$$

$$g = g/32 \tag{2}$$

$$b = b/32 \tag{3}$$

As we want to calculate the color distributions and then measure the similarity among multiple models of the hypotheses, a one-dimensional feature vector *u* corresponding to the three component values *r*, *g*, and *b* are calculated as

$$u = r \times 8 \times 8 + g \times 8 + b \tag{4}$$

Color histograms merely provide a compact summarization of the distribution of data in an image, and don't take into account the position information of pixels. To increase the reliability of the color distribution when boundary pixels belong to the background or get occluded, smaller weights are assigned to the pixels that are further away from the region center by employing a weighting function

$$k(r) = \begin{cases} 1 - r^2, \; r < 1 \\ 0, \; r > 1 \end{cases} \tag{5}$$

where r is the distance from the region center. Thus we take into account the position information of pixels and increase the reliability of the color distribution when these boundary pixels belong to the background or get occluded.

In this case, the distribution for feature value *u* is calculated as

$$H(u) = f \sum_{i=1}^{I} k\left(\frac{\| y - x_i \|}{a}\right) \delta\left(h(x_i) - u\right) \tag{6}$$

where I is the number of pixels in the region, $\| y - x_i \|$ is the distance between pixel x_i and the center of region, δ is the Kronecker delta function, $h(x_i)$ assigns the color of pixel x_i to the corresponding bin, the parameter $a=\sqrt{H_x^2+H_y^2}$ is used to adapt the size of the region, H_x and H_y are respectively the length and width of the region, and the normalization factor

$$f=\frac{1}{\sum_{i=1}^{I}k(\frac{\|y-x_i\|}{a})} \quad (7)$$

ensures that $\sum_{u=0}^{m-1}H(u)=1$, where m is the number of feature levels(m =512 in our case).

According to the above analysis, the steps of the target model construction algorithm using RGB color histograms can be summarized as follows:

1) Extract region of the tracking target from an image, and feed in the RGB triplets (r, g,b), each component of which can vary from 0 to 255.
2) Reduce the original 256× 256× 256 color levels to 8× 8× 8 levels, and combine them to a one-dimensional feature vector u according to (1), (2), (3), (4).
3) Assign smaller weights k to the pixels accordingto their distance r away from the center of region using (5).
4) Calculate the distribution for feature value u employing (6), (7), and thus obtain a target model $H(u)$ based on a weighted color histogram.

3.3 Density Propagation based Particle Filter

In this section, we present the overall structure of DP-PF, where the density propagation algorithm is integrated with particle filtering.

We introduce a density propagation based algorithm for particle initialization. The key idea is to localize the object in the next frame by utilizing useful particle state information which is stored after object tracking in the previous frames. The choice is motivated by its efficiency and robustness. Here, we consider the video clip used for tracking as a consecutive image sequence, generally 30 frames per second, with slight changes frame by frame. On one hand, due to the duration of the target state, the state vector which characterizes the target (region or shape parameters), can have a significant effect to that of the next frame. It therefore decreases the search scope, reduces the computational cost, and most importantly, avoids falling into local optima. On the other hand, when a serious occlusion takes place between the target and other objects, the state information derived from the frame in which the target last appears is of great help to recognize it again when it reappears. Thus, we achieve robustness against partial or complete occlusion.
Specifically, the density propagation technique is mainly applied in the phase of particle initialization, and consists of two aspects, as shown in Fig. 2.

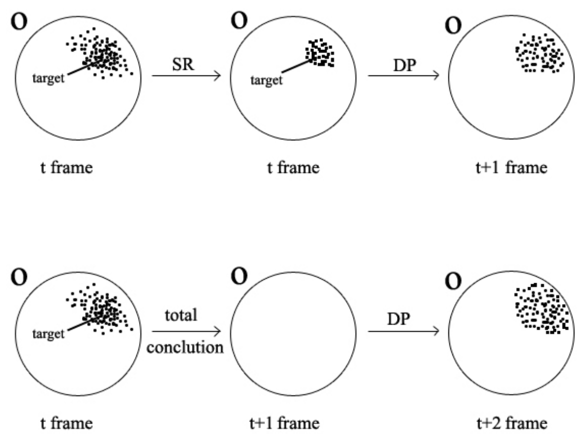

Fig. 2. Two aspect of density propagation

1) According to particle filtering object tracking algorithm, the state information of particles with higher fitness is reserved at time t. When updating the initial state of particles at time $t+1$, the useful information is reused to distribute particles around areas where the particles have obtained higher fitness in the last frame. The particles are re-initialized through application of the first-order ARP equation of particle filtering.

$$q_{t+1}=Aq_t+Bw \quad (8)$$

where A is a transition matrix, B is a weighting component, q_t is the state vector of particle at time t, and w is process noise.

In our work, every particle stands for a possible state of the target, and is given as { cx, cy, hx, hy }, where (cx, cy) denotes the center of the image region, simulated by particle in our case, within which the computation of object's color histogram is carried out; hx and hy denote the image region's width and length. According to (8), the state vector of particle is propagated by

$$cx_{t+1} = Acx_t + Bw \tag{9}$$

$$cy_{t+1} = Acy_t + Bw \tag{10}$$

$$hx_{t+1} = Ahx_t + Bw \tag{11}$$

$$hy_{t+1} = Ahy_t + Bw \tag{12}$$

2) The density propagation algorithm plays a important part in the case of temporal occlusion. Assume that the fitness of the optimal particle is f_{best}^{t+1} in time *t+1*, and we obtain a threshold value f_{thr}, then $f_{best}^{t+1} < f_{thr}$ indicates that the target is missing at time $t+1$. In this case, we calculate the initial state vector q_{t+2} of the particles utilizing the information q_t derived from time *t*, where

$$q_{t+2} = Aq_t + Bw \tag{13}$$

A common problem with the conventional particle filter is the degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight. To mitigate this problem, we adopt the particle resampling technique. The basic idea of resampling is to eliminate particles that have small weights and to concentrate on particles with larger weights. The resampling step involves generating a new sample set $\{x_k^i\}_{i=1,\dots,N}$ at time *k* by resampling (with replacement) *N* times from the posterior probability density, where *N* is the total number of particles. Finally, the original weights $\{x_k^i, w_k^i\}_{i=1,\dots,N}$ are reset to $\{x_k^i, 1/N\}_{i=1,\dots,N}$. Fig. 3 briefly displays the principle of resampling.

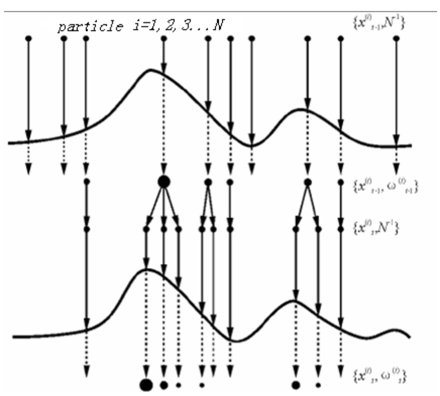

Fig. 3. Particle resampling

More specifically, the major steps of resampling algorithm are given as follows:

Step1. According to the traditional particle filter [16], update the particle weight w_k^i, $i = 1, 2,\dots, N$, at time *k* using the following equation:

$$w_k^i = w_{k-1}^i \times p(z_k \mid x_k^i) \tag{14}$$

where $p(z_k \mid x_k^i)$ is the probability density function of particle's new state.

Step2. Calculate the resampling threshold W_{thr} as

$$W_{thr} = \sum_{i=1}^{N} w_k^i \,/\, N \tag{15}$$

Step3. For each w_k^i, $i = 1, 2,\dots, N$, if $w_k^i < W_{thr}$, eliminate particle *i*, store the particles with larger weights, which will be involved into a new sample set.

Step4. Reset the particle weight to $w_k^i = 1/N$, and finally generate a new sample set with equal weights.

3.4 Similarity Measure

In a tracking approach, the estimated state is updated at each time step by incorporating the new observations. Therefore, it becomes crucial to formulate an effective method to measure the similarity between the target model and the model of the hypotheses. Particularly, our method should be based on color distributions. Here, we adopt a metric derived from the Bhattacharyya coefficient [3], which is commonly employed to measure the similarity between the template region and the current target region.

Considering the target model $H_q(u)$ and the models of candidate $H_t^i(u)$ at time *t*, the coefficient r is defined as

$$r(H_q, H_t^i) = \sum_{u=0}^{m-1} \sqrt{H_q(u) \times H_t^i(u)}, i = 1,\cdots,N \tag{16}$$

ISBN: 978-0-9853483-3-5 ©2013 SDIWC

The formulation of $H_q(u)$ and $H_t^i(u)$ is defined as (6), r measures the similarity between color distributions of the target model and the model of candidate. The larger r is, the more similar the two color distributions are.

4 EXPERIMENTAL RESULTS

We performed an extensive set of experiments to evaluate the performance of the proposed method. The video data for testing includes video data from [23], CAVIAR video set [24], and video clips compiled from indoors and outdoors on our own, in which the environments are unconstrained. The datasets used in our work also differ in the image size, e.g., 390×320, 385×290, 128×96, and 250×225. The proposed method is implemented using C++, and runs on Intel Pentium Dual 1.86G CPU and 2G memory.

4.1 Time performance of DP-PF

In this experiment, we analyze the time performance of DP-PF, which is crucial to real-time tracking. We calculate the average processing time of tracking multiple people in a surveillance video with 500 frames (385×290 pixels) using the CAVIAR dataset. Table 1 shows the results.

Table 1. Time performance of DP-PF

Particle number	100	200	300	400	500
Processing time(ms)	8.25	13.4	18.8	24.4	29.2

As is shown in Table 1, the proposed algorithm runs in real time with about 75 frames per second given that the total number of particles is only 200. Generally speaking, in real-time applications, such a tracking efficiency has sufficiently satisfied their command for real-time performance.

4.2 Comparison between DP-PF and DA-PF

In order to judge the performance of our algorithm, we compare to results obtained using Data-Driven Adaption based Particle Filter (DA-PF) algorithm [18]. We illustrate the differences of time performance and tracking effect between DA-PF algorithm and our proposal by experiments of tracking objects under various complicated conditions.

4.2.1 Time performance comparison

Given a video with 500 frames (385×290 pixels), of which the search region contains 25×55 pixels, we calculate the average processing time of the two algorithms respectively. Table 2 shows the comparison results.

Table 2. Time performance of DP-PF and D A-PF

Particle number	100	200	300	400	500
DP-PF (ms)	8.25	13.4	18.8	24.4	29.2
DA-PF (ms)	16.13	25.9	34.9	44.9	53.8

As shown in Table 2, the processing time of our approach is nearly half of that of DA-PF. Therefore, our approach achieves better real-time performance and lower computational cost than DA-PF.

4.2.2 Tracking a biker with complete occlusion

This experiment shows how the tracking performance of our approach is affected when complete occlusion occurs, also comparing with the DA-PF algorithm. We demonstrate results for a *riding* sequence, which is compiled by ourselves and where the tracker follows a single biker over consecutive frames of 250×225 pixels in size. We firstly track the object using DA-PF. Here we have $N=200$, where N is the total number of particles. We specify the target in the first frame and track it in the following frames. The tracking results for DA-PF are displayed in Fig. 4(a). In this sequence, the rider is completely occluded by the tree in frame 26 such that the DA-PF algorithm drifts rapidly to background and thus loses the tracking object in the following frames.In contrast, although the proposed method also loses the target at frame 26, it is able to recover quickly and recognize the target again at frame 34 thanks to the help of the density propagation mechanism. Specifically, from frame 27 to frame 34, the particles are re-initialized using density propagation and still distributed around the area, in which the target lastly appeared, to search the target. Therefore, our approach is superior for tracking an object in situations where the target may be completely occluded in a short time. The tracking results for DP-PF are displayed in Fig. 4(b). Fig. 4(c) demonstrates the procedure of target searching.

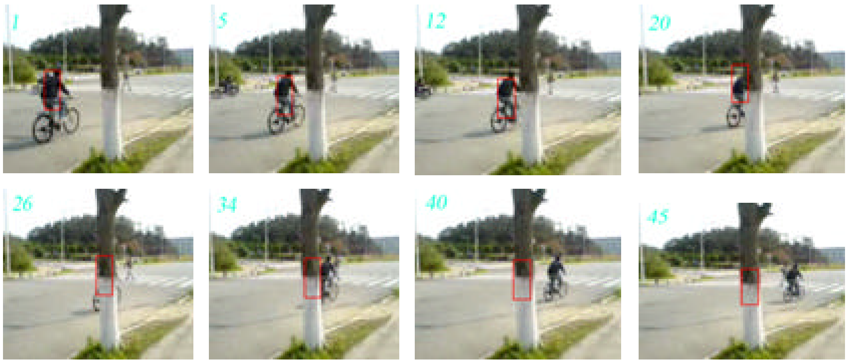

Fig. 4(a). Tracking a biker with complete occlusion using DA-PF

Fig. 4(b). Tracking a biker with complete occlusion using DP-PF

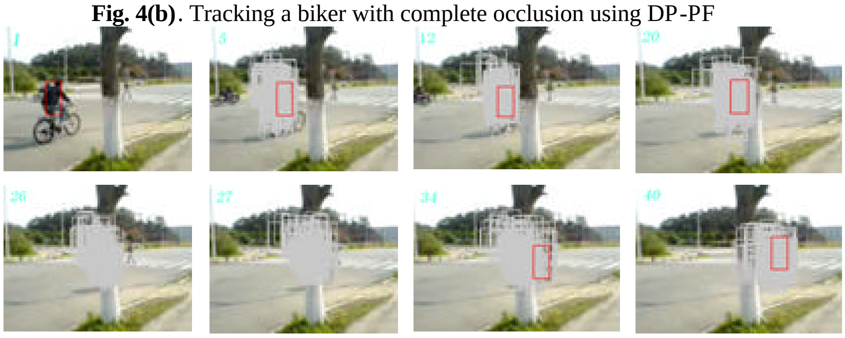

Fig. 4(c). Particle distribution of DP-PF

4.3.3 Tracking a face with similar object disturbance

In this experiment, we demonstrate the discriminative power of our approach, comparing with the DA-PF. In the video sequence as following, the goal is to follow the women's face while another face in the close vicinity is approaching to it. The frame size is pixels and the total number of particles is 200. The tracking result using DA-PF is shown in Fig. 5(a), from which we can see that the tracker is distracted at frame 25 since another similar face gets closer, mistakes it for the true target, and finally leads to false detection in the following frames.

On the other hand, the proposed approach can work comfortably and stably in the case. As shown in Fig. 5(b), when the target is partially occluded by another object with similar appearance, it is able to discriminate between them and tracks the target successfully in the following frames. Therefore, our approach achieves higher discriminative power.

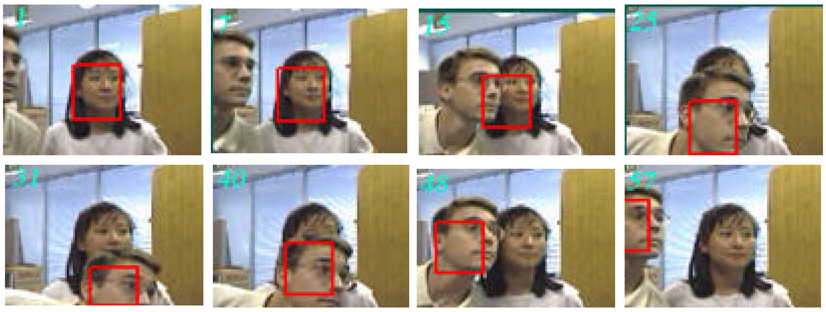

Fig. 5(a). Tracking a face with similar object disturbance using DA-PF

Fig. 5(b). Tracking a face with similar object disturbance using DP-PF

4.3.4 More results

To demonstrate the superior performance of our density propagation based particle filter (DP-PF), we consider a video sequence compiled by ourselves, where the tracker follows a face over consecutive frames and the target is partially occluded gradually. The tracking result is displayed in Fig. 6.

CAVIAR video set, where the targets are subject to scale changes but without appearance changes. As shown in Fig. 7, the proposed method can well handle scale changes induced by the people walking towards the camera.

5 Conclusions

This paper presents a novel approach to video object tracking. In the work, we employ color histograms as target models and integrate density propagation algorithm into particle filtering for particle initialization, which makes the system especially robust to view partial and complete occlusions, and with a low computational cost. This approach can also achieve a high discriminative power against similar object disturbance and tolerate scale changes. Proposed method has been tested on real videos. The results show the effectiveness of the proposed algorithm.

Fig. 6. Tracking a face with partial occlusion using DP-PF

Fig. 7. Tracking a person with scale changes using DP-PF

6 REFERENCES

1. Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems. Transaction of the ASME-Journal of Basic Engineering 82, 35--45 (1960).
2. Ribeiro, M. I.: Kalman and Extended Kalman Filters: Concept, Derivation and Properties. Institute for Systems and Robotics, 1--40 (2004).
3. Comaniciu, D., Ramesh, V., Meer, P.: Real-time Tracking of Non-rigid Objects Using Mean Shift. IEEE Conference on Computer Vision and Pattern Recognition 2, pp. 14--149, Hilton Head, SC (2000).
4. Gao, J. P., Wang, Y. J., Yang H.: Face Tracking Algorithm Based on Mean Shift and Ellipse Fitting. In: Proc. Neural Informational 4233, pp. 270--277 (2006).
5. Birchfield, S., Rangarajan, S.: Spatiograms Versus Histograms for Region-based Tracking. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition 2, pp. 115--1163, San Diego, CA (2005).
6. Park, M. W., Liu, Y. X.: Collins R T. Efficient Mean Shift Belief Propagation for Vision Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1--8 (2008).
7. Yang, C., Duraiswami, R., Davis, L.: Efficient Mean-shift Tracking via A New Similarity Measure. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition 1, pp. 176--183 (2005).
8. Jiang, Z. L., Li, S. F., Jia, X. P.: An Improved Mean Shift Tracking Method Based on Nonparametric Clustering and Adaptive Bandwidth. In: International Conference on Machine Learning and Cybernetics 5, pp. 277--2784, Kunming, China (2008).
9. Corvee, E., Velastin, S. A., Jones, G. A.: Occlusion Tolerent Tracking using Hybrid Prediction Schemes. Acta Automatica Sinica 29, 356--369 (2003).
10. Lowe, D G.: Distinctive Image Features from Scale Invariant Key Points. International Journal of Computer Vision 60, 91--110 (2004).
11. Gordon, N. J., Salmond, D. J., Smith, A. F. M.: A Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. In: Proc. IEE Radar and Signal Proceedings 140, pp. 107--113 (1993).
12. Arulampalam, M. S., Maskell, S., Gordon, N,: A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Trans. on Signal Processing 50, 174--188 (2002).
13. Hammersley, J. M., Morton, K. W.: Poor man's Monte Carlo. Journal of the Royal Statistical Society, Series B (Methodological) 16, 23--38 (1954).
14. Isard, M., Blake, A.: Condensation-Conditional Density Propagation for Visual Tracking. International Journal on Computer Vision 29, 5--28 (1998).
15. Gordon, N., Salmond, D.: Bayesian State Estimation for Tracking and Guidance Using The Bootstrap Filter. Journal of Guidance, Control and Dynamics 18, 1434--1443 (1995).
16. Zia, K., Balch, T., Dellaert, F.: MCMC-Based Particle Filtering for Tracking A Variable Number of Interacting Targets. IEEE Trans. on Pattern Analysis and Machine Intelligence 27, 1805--1819 (2008).
17. Nummiaro, K., Koller-Meier, E., Gool, L. V.: An Adaptive Color-Based Particle Filter. Image and Vision Computing 21, 99--110 (2003).
18. Yang, M., Fan, Z. M., Fan, J. L.: Tracking Nonstationary Visual Appearances by Data-Driven Adaptation. IEEE Trans. on Image Processing 18, 1633--1644 (2009).
19. Han, B., Zhu, Y., Comaniciu, D.: Visual Tracking by Continuous Density Propagation in Sequential Bayesian Filtering Framework. IEEE Trans. on Pattern Analysis and Machine Intelligence 31, 919--930 (2009).
20. Venkatesh, B. R., Makur, A.: Kernel-Based Spatial-Color Modeling for Fast Moving Object Tracking. In: IEEE Conference on Acoustics, Speech and Signal Processing 1, pp. I-901--I-904 (2007).
21. Wu, P. L., Kong, L. F., Zhao, F. D.: Particle Filter Tracking Based on Color and SIFT Features. In: IEEE Conference on Audio, Language and Image Processing, 932--937 (2008).
22 Jeyakar, J., Babu, R. V., Ramakrishnan, K. R.: Robust Object Tracking with Background-Weighted Local Kernels. IEEE Trans. on Computer Vision and Image Understanding 112, 296--309 (2008).
23. Zhang, X. Q., Hu W. M., Maybank, S.: Sequential Particle Swarm Optimization for Visual Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, 1--8 (2008).
24. EC Funded CAVIAR Project/IST 2001 37540, http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIAR DATA1/.