Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Embedding ensemble tracking in a stochastic framework for robust object tracking*

Yu GU, Ping LI†‡, Bo HAN
(*Institute of Industrial Process Control, Zhejiang University, Hangzhou 310027, China*)
†E-mail: pli@iipc.zju.edu.cn

Received Sept. 7, 2008; Revision accepted Dec. 22, 2008; Crosschecked June 10, 2009

Abstract: We propose an algorithm of embedding ensemble tracking in a stochastic framework to achieve robust tracking performance under partial occlusion, illumination changes, and abrupt motion. It operates on likelihood images generated by the ensemble method, and combines mean shift and particle filtering in a principled way, where a better proposal distribution is designed by first propagating particles via a motion model, and then running mean shift to move towards their local peaks in the likelihood image. An observation model in the particle filter incorporates global and local information within a region, and an adaptive motion model is adopted to depict the evolution of the object state. The algorithm needs fewer particles to manage the tracking task compared with the general particle filter, and recaptures the object quickly after occlusion occurs. Experiments on two image sequences demonstrate the effectiveness and robustness of the proposed algorithm.

Key words: Ensemble tracking, Particle filter, Mean shift, Likelihood mean
doi:10.1631/jzus.A0820647 **Document code:** A **CLC number:** TP317.4

INTRODUCTION

Tracking objects in a video is a critical step in computer vision applications, including vision-based robot control, surveillance, human computer interactions, and sports video analysis (Yilmaz *et al*., 2006). As a task of locating objects in successive frames, it has been intensively studied in these decades along with the prevalence of computers. Due to the difficulties of illumination changes, cluttered background, partial occlusion, and abrupt motion, robust object tracking in unconstrained environment is still an open problem, although considerable efforts have been made (Isard and Blake, 1998a; 1998b; Perez *et al.*, 2002; Comaniciu *et al.*, 2003; Jepson *et al.*, 2003; Zhou *et al.*, 2004; Collins *et al.*, 2005).

Visual tracking can be roughly divided into two categories: deterministic tracking and stochastic tracking (Comaniciu *et al.*, 2003). The former type is reduced to an optimization problem, where an error function with respect to the object state is minimized. Partial occlusion can be successfully handled with the advantage of computational efficiency, but this type is likely to suffer from abrupt motion and cannot recover after full occlusion. The latter one can be viewed as an estimation problem. Particle filters (Arulampalam *et al.*, 2002) are successfully used to cope with nonlinearity and non-Gaussianity during tracking, but when the dimension of the state is very large, more particles will be needed to represent the posterior density and hence it will be too computationally expensive to meet real-time requirements.

Several algorithms have already been presented to combine these two types of tracking algorithms. Zhou *et al.*(2004) devised an algorithm of tracking objects using an adaptive motion model in a stochastic framework, where the motion model was calculated based on the appearance difference between two successive frames. In Odobez *et al.*(2006), an inter-frame motion estimate was used as their motion

‡ Corresponding author
* Project (No. 2006AA10Z204) supported by the National High-Tech Research and Development Program (863) of China

model and observation model to improve the performance of particle filter based tracking. The algorithm of the mean shift embedded particle filter (MSEPF) (Shan *et al.*, 2007) was proposed for real-time hand tracking, which incorporates mean shift optimization (Bradski, 1998) into the particle filter to improve sampling efficiency and reduce computational cost.

Ensemble tracking (Avidan, 2007) is a general framework for tracking objects, which generates likelihood images using the ensemble method, runs mean shift to locate objects, and maintains temporal integration by updating the ensemble. Although achieving robust tracking performance in most cases, it has the same disadvantages as deterministic tracking due to the use of mean shift optimization. We extend the work of Avidan (2007) to embed ensemble tracking in a stochastic framework for robust object tracking. The work most closely related to ours is MSEPF (Shan *et al.*, 2007), which also uses mean shift to move particles towards their local modes to design a better proposal distribution for particle filters, but the observation model in our algorithm incorporates global and local information within a region, and meanwhile an adaptive motion model is adopted to depict the evolution of the object state. Our algorithm features a better generalization capacity owing to the application of the ensemble method for computing likelihood images.

EMBEDDING ENSEMBLE TRACKING IN A STOCHASTIC FRAMEWORK

Ensemble tracking

Ensemble tracking treats target tracking as a local discrimination problem, and combines a set of weak classifiers to separate pixels of the object from those of its surroundings. The general algorithm of ensemble tracking is summarized in Algorithm 1.

Algorithm 1 General ensemble tracking

Input: *n* video frames and rectangle of the object in the first frame.

Output: rectangles in successive frames.

Initialization: Train a set of weak classifiers and add them to the ensemble.

For each new frame do:

1. Create a likelihood image using the ensemble.
2. Run mean shift to find the position of the object in the current frame.
3. Update the ensemble.

The difference between ensemble tracking and other tracking algorithms using mean shift is the way of generating likelihood images. The likelihood image in CAMSHIFT (Bradski, 1998) is computed by color histogram back projection. In Collins *et al.* (2005), the algorithm operates on likelihood images computed through log likelihood ratios of class conditional sample densities from the object and background. The candidate features are the combinations of R, G, B values, which are suitable for color images only. It was improved in Avidan (2007) by modeling pixels using local information and training a set of weak classifiers online to separate them using the ensemble method. The likelihood image is computed through the margins of weak classifiers. Ensemble tracking can be used for color, gray-scale, and IR sequences.

Particle filter

Particle filtering (Arulampalam *et al.*, 2002), also known as CONDENSATION (Isard and Blake, 1998a), is a technique for implementing Bayesian state estimation by Monte Carlo simulations. It approximates the required posterior density by a set of weighted particles.

Two models are required when this technique is applied to object tracking. One is the motion model, depicting the evolution of the object state with time; the other is the observation model, being used to compute the weight of each particle. The general form of these two models is given in the form of the state space model as follows:

$$\begin{cases} \boldsymbol{x}_t = F_t(\boldsymbol{x}_{t-1},\ \boldsymbol{m}_{t-1}), \\ \boldsymbol{z}_t = H_t(\boldsymbol{x}_t,\ \boldsymbol{n}_t), \end{cases} \tag{1}$$

where $\boldsymbol{x}_t$ denotes the object state at time t; $\boldsymbol{z}_t$ is the measurement; $\boldsymbol{m}_t$ and $\boldsymbol{n}_t$ are system noise and measurement noise, respectively.

Given a set of weighted particles at time $t-1$, particles are drawn by sampling from a proposal distribution $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\ \boldsymbol{z}_t)$, and measured according to the new measurement $\boldsymbol{z}_t$. The weight of each particle is updated as follows:

$$\pi_t = \pi_{t-1} \frac{p(\boldsymbol{z}_t \mid \boldsymbol{x}_t)\, p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1},\ \boldsymbol{z}_t)}, \tag{2}$$

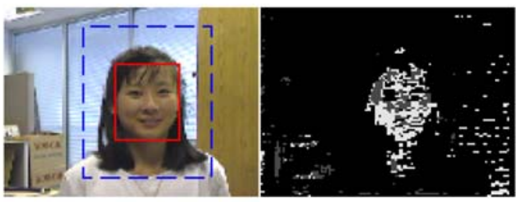
where $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ is the likelihood function, and $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is the state transition probability function. Both can be derived from Eq.(1). The object state is estimated according to these particles after their weights are normalized. A new set of particles with equal weight is obtained after resampling, which is often involved to avoid a degeneracy problem.

In the implementation of particle filters, an important issue is how to choose an appropriate proposal distribution. Usually the choice is to sample from the temporal prior $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, which is equivalent to propagating each particle via the motion model, and updating the weight by $\pi_t=\pi_{t-1}p(\boldsymbol{z}_t|\boldsymbol{x}_t)$. This algorithm is used in most particle filter based visual tracking algorithms.

Ensemble tracking embedded particle filter

Ensemble tracking exploits mean shift to locate the object in image frames, and consequently suffers from local minima and cannot handle full occlusion and abrupt motion. In the original ensemble tracking algorithm (Avidan, 2007), another particle filter based tracking algorithm is used to recapture the object after occlusion occurs. This method is empirical, combining strength of neither ensemble tracking nor particle filter based tracking. As seen from Fig.1, the object is located within the highlighted region in the likelihood image, from which samples can be drawn to improve tracking performance. This idea is similar to ICONDENSATION (Isard and Blake, 1998b), where color segmentation is employed to find high likelihood regions and draw particles; thus, an algorithm is proposed to embed ensemble tracking into a stochastic framework, and to incorporate the advantages of both deterministic tracking and stochastic tracking to achieve robust tracking performance.

The shape of the tracked object is modeled as a 0-centered rectangle R, the width and height of which



(a) (b)

Fig.1 A video image (a) and its likelihood image (b)
The object lies inside the inner rectangle in (a) and the selected background lies outside the inner rectangle and inside the outer rectangle when computing the likelihood image

are fixed and supplied manually in the first frame. Let $\boldsymbol{x}_t=(u_t, v_t, s_t)$ denote the object state at time t, where $\boldsymbol{c}_t=(u_t, v_t)$ is the center of the rectangle, and s_t is the scale with an initial value 1, so the candidate region in the image plane is given by $f(\boldsymbol{x}_t)=\boldsymbol{c}_t+s_t\times R$.

Denote the value at pixel (u, v) in the likelihood image L as $L(u, v)$, and the likelihood mean within a region W is defined as follows:

$$w(W)=\frac{1}{\text{AREA}(W)}\sum_{(u,v)\in W}L(u,v)>T_1, \tag{3}$$

where T_1 is fixed when the way of generating likelihood images is chosen (e.g., T_1=0), if we adopt the same way as in Avidan (2007).

The difference of the likelihood mean between an object and its surroundings, which is defined as Eq.(4), is used to monitor tracking conditions. The likelihood mean of the object, $w_{\text{obj}}(f(\boldsymbol{x}_t))$, is computed within the region $f(\boldsymbol{x}_t)$; the likelihood mean of the background, $w_{\text{bg}}(f(\boldsymbol{x}_t))$, is computed within the region outside the rectangle of the object and inside a bigger one, as illustrated in Fig.1.

$$\text{diff}(\boldsymbol{x}_t)=w_{\text{obj}}(f(\boldsymbol{x}_t))-w_{\text{bg}}(f(\boldsymbol{x}_t)). \tag{4}$$

If occlusion occurs, w_{obj} decreases; if distraction occurs, w_{bg} increases. diff decreases in both conditions. If w_{obj} and w_{bg} decrease or increase at the same time, and diff changes gradually, illumination changes or an abrupt motion may occur.

When embedding ensemble tracking into the particle filter, the whole algorithm proceeds as follows. After the particle set with equal weight is obtained, particle $\boldsymbol{x}_{t-1}$ is first propagated via the motion model to obtain $\hat{\boldsymbol{x}}_t$, and then moves towards its local peak $\tilde{\boldsymbol{x}}_t$ in the likelihood image. The posterior density is biased if we directly use $\tilde{\boldsymbol{x}}_t$ as another new sample, and thus a Gaussian distribution is superimposed on $\tilde{\boldsymbol{x}}_t$, from which a new particle $\bar{\boldsymbol{x}}_t$ is drawn. This makes sequential importance sampling unbiased and valid. The weight of this new particle is then updated according to Eq.(2), because importance sampling is used to draw samples from this new proposal distribution.

After obtaining $2N$ particles, we update the weight of each particle according to the new measurement, which is then normalized. The object state is estimated using these particles, and the resampling process is performed to obtain N particles with equal weight. We compute $w_{\rm obj}$, $w_{\rm bg}$ and diff using the current state, which is useful to find the situation of distraction and occlusion. If $w_{\rm obj}$ is below a threshold T_2, occlusion may occur, and the current mode is switched to FIND mode, which means that resampling is not needed. Otherwise, if $w_{\rm obj}$ is above this predefined threshold, and some similarity score is higher than T_3, the mode is switched to TRACK mode again. If the current mode is TRACK mode, an adaptive motion model is adopted by updating the motion parameters.

Although it is helpful to update the ensemble to adapt to changes of the object and background, ensemble tracking may learn patterns not belonging to the object if occlusion occurs, and the model often drifts since the ensemble method is sensitive to outliers. Avidan (2007) also stated that the tension between adaptation and drifting appears when updating the ensemble, so the update mechanism is not used in this study. The overall algorithm of the ensemble tracking embedded particle filter is presented below:

Given a set of N particles $\{\boldsymbol{x}_{t-1}^i,\ 1/N\}_{i=1}^N$ at time $t-1$, perform the following steps:

1. Create likelihood image L_t using the ensemble.

2. Draw particles as follows:

Propagate each particle via the motion model to obtain $\{\hat{\boldsymbol{x}}_t^i,\ 1/N\}_{i=1}^N$, and set $\lambda_t^i=1$.

Sample another N particles using importance sampling. For each particle $\hat{\boldsymbol{x}}_t^i$:

(1) Run $\tilde{\boldsymbol{x}}_t^i=\mathrm{MeanShift}(\hat{\boldsymbol{x}}_t^i)$ in the likelihood image.

(2) Sample $\bar{\boldsymbol{x}}_t^i \sim q(\bar{\boldsymbol{x}}_t^i \mid \tilde{\boldsymbol{x}}_t^i,\ \boldsymbol{z}_k)=\mathrm{Gaussian}(\tilde{\boldsymbol{x}}_t^i,\ \Sigma)$, Σ is the variance of the state. Set $\lambda_t^i=p(\bar{\boldsymbol{x}}_t^i \mid \boldsymbol{x}_{t-1}^i)\big/q(\bar{\boldsymbol{x}}_t^i \mid \tilde{\boldsymbol{x}}_t^i,\ \boldsymbol{z}_t)$.

A set of $2N$ particles is obtained when combining these two sets of particles.

3. Compute the weight of each particle by $\pi_t^i=\lambda_t^i p(\boldsymbol{z}_t \mid \bar{\boldsymbol{x}}_t^i)$, and make normalizations so that $\sum_{i=1}^{2N}\pi_t^i=1$.

4. Estimate the state by $E(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t})=\sum_{i=1}^{2N}\pi_t^i\bar{\boldsymbol{x}}_t^i$, and compute $w_{\rm obj}$ using the current state. If $w_{\rm obj}<T_2$, switch to FIND mode; else, run in TRACK mode.

5. If the current mode is TRACK mode, update the motion parameter, and obtain a set of new particles through resampling:

$$\left\{\boldsymbol{x}_t^i,\ 1/N\right\}_{i=1}^N=\mathrm{Resample}\left\{\bar{\boldsymbol{x}}_t^i,\ \pi_t^i\right\}_{i=1}^{2N}.$$

EXPERIMENTAL RESULTS

We implemented the proposed algorithm in MATLAB, and tested it on two image sequences. One test sequence was a public image sequence with 500 frames, which can be downloaded from http://vision.stanford.edu/~birch/. This image sequence has been adopted by many researchers to test their algorithms, because of its capacity in simulating various tracking conditions, including illumination changes, pose variations, occlusions, and distraction. The other was a landmark sequence captured by our own USB camera; the challenges of this sequence include illumination changes and abrupt motion. In our experiments, the ensemble tracking embedded particle filter (ETEPF) was compared with the original ensemble tracking (ET) (Avidan, 2007) and color based probabilistic tracking (PF) (Perez *et al.*, 2002). The number of particles in PF was 100, which was also used in Perez *et al.*(2002) to manage the tracking process, and fewer particles were used in ETEPF because a better proposal distribution was adopted—10 particles were used for achieving comparable results.

Implementation details

Since ensemble tracking is a general framework for tracking objects, several choices, including different local features and boosting algorithms, and a various number of weak classifiers, are evaluated to achieve a good tracking performance. The histogram of oriented gradient (HOG) descriptor (Dalai and Triggs, 2005) combined with R, G, B values was finally chosen as the local feature vector, and AdaBoost (Freund and Schapire, 1996) was used to train and combine five weak classifiers. The likelihood value at each pixel was computed in the way derived from the statistical view of boosting (Friedman *et al.*, 2000), which is defined as follows:

$$L(u,v)=\frac{1}{1+\exp(-2G(u,\ v))}, \tag{5}$$

where $G(u,\ v)$ is the weighted sum of the output of weak classifiers. This makes the likelihood value lie in the domain (0, 1). T_1 was 0.5, and T_2 was set to be 0.1 after evaluating the likelihood image.

An adaptive first order motion model was used to model the dynamics:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + C_t \boldsymbol{m}_t, \tag{6}$$

where $\boldsymbol{m}_t$ is jointly Gaussian with zero mean and unit variance, and C_t can be estimated according to the history of the object state. The lower bound of the object velocity was set to be 5 to handle abrupt motion. The details of estimating C_t can be found in Maggio and Cavallaro (2005).

Color histogram as a global feature for a region was also used to represent the appearance of the object, due to its invariance to translation and scale changes, and robustness to partial occlusion. Let $p(\boldsymbol{x}_t)=\{p(\boldsymbol{x}_t, n)\}_{n=1, 2, \ldots, M}$ denote the color histogram in HSV space given the state $\boldsymbol{x}_t$. The similarity between a candidate model $p(\boldsymbol{x}_t)$ and the reference model p^*, $\rho(p(\boldsymbol{x}_t), p^*)$, is measured by the Bhattacharyya distance, d:

$$d = \sqrt{1-\rho(p(\boldsymbol{x}_t), p^*)}, \tag{7}$$

where

$$\rho(p(\boldsymbol{x}_t), p^*) = \sum_{n=1}^{M} \sqrt{p(\boldsymbol{x}_t, n) p^*(n)}.$$

Then, the observation likelihood is computed as follows:

$$p(\boldsymbol{z}_t \mid \boldsymbol{x}_t) = \exp(-\lambda d^2(p^*, p(\boldsymbol{x}_t))), \tag{8}$$

where λ=20, which is the same as in Perez *et al.* (2002). $\rho(p(\boldsymbol{x}_t), p^*)$ was used to measure the similarity between candidates and the template when switching the current mode from FIND to TRACK, and T_3 was set to be 0.5 in our case.

The likelihood mean within a region reflects its discriminative power from background, usually high within the region of the object and low in other regions. The final likelihood function is given below when combined with the likelihood mean:

$$p_{\text{final}}(\boldsymbol{z}_t \mid \boldsymbol{x}_t) = w_{\text{obj}}(\boldsymbol{x}_t) \exp(-\lambda d^2(p^*, p(\boldsymbol{x}_t))). \tag{9}$$

Results on the public image sequence

The object we wanted to track was a woman's frontal face. The initial bounding box was supplied manually with center $\boldsymbol{c}_0$=(71, 46), width and height (W, H)=(33, 39). The tracking results are shown in Fig.2, and the trajectories along u- and v-direction in the image plane for the three algorithms are plotted in Fig.3. At the beginning, the likelihood image was very clean, so all algorithms succeeded in tracking the object during this period. From frame 83, the object turned around and a self-occlusion occurs; these three algorithms failed to track the object, and ETEPF switched to FIND mode. After the object reappeared, ETEPF captured the object, and lost it after several frames, because illumination changed severely and the likelihood mean was very low at this time. ETEPF recaptured the object in frame 247, and ET relocked onto the object until frame 350. A man with a similar appearance entered the scene from frame 411. All algorithms locked onto this distraction, and the position error became large when it occluded the object. PF and ETEPF recaptured the object after the man moved away, but ET failed.

The root mean square errors (RMSEs) along u- and v-direction are presented in Table 1, showing that ensemble tracking performed the worst, and the

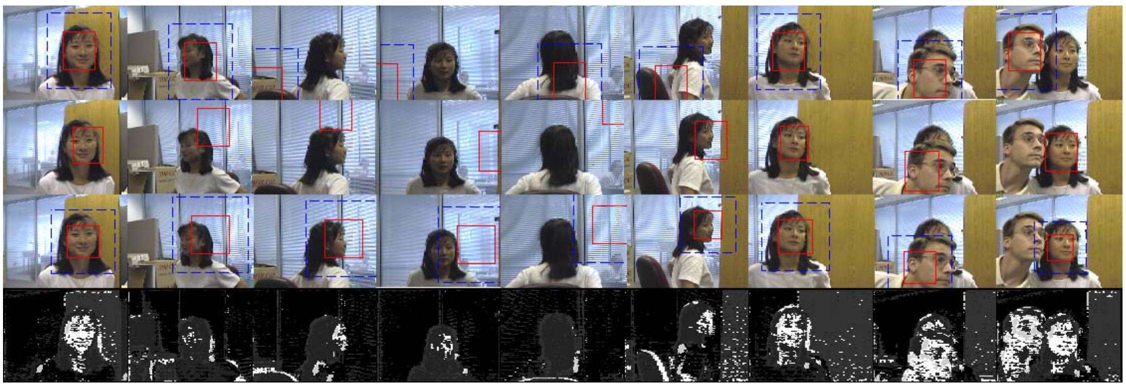


Fig.2 Results on the public image sequence

ET, PF, and ETEPF are used from the first to the third row, respectively, and the corresponding likelihood images are shown in the fourth row. From left to right, the frame indexes are 16, 83, 113, 144, 202, 247, 350, 438, and 469

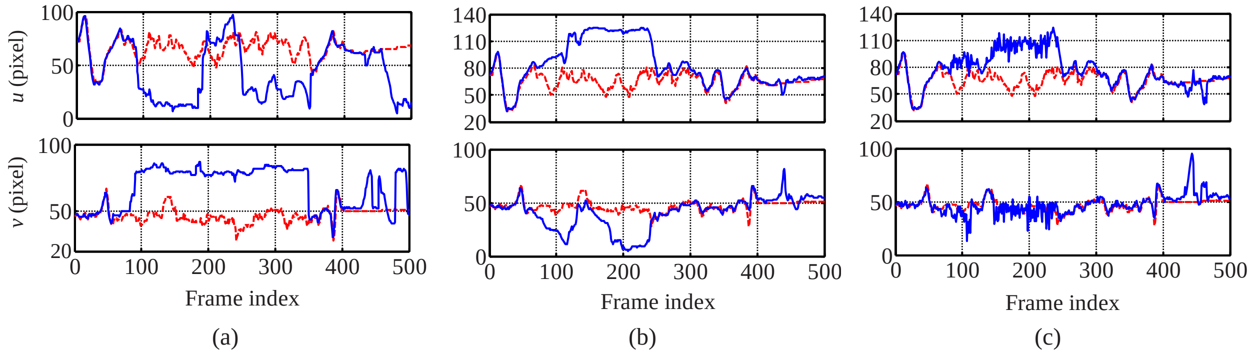


Fig.3 Trajectories in the image plane along *u*- and *v*-direction
(a) ET; (b) PF; (c) ETEPF. The solid line stands for estimation and the dashed line stands for ground truth

Table 1 Root mean squared errors for the public image sequence

Algorithm	RMSE (pixel)	
	u	*v*
ET	24.68	21.44
PF	18.51	19.33
ETEPF	13.28	4.51

position error reduced greatly when embedding ensemble tracking into the particle filter. Our algorithm, as a result of incorporating the advantages of both ET and PF, can recover quickly after occlusion occurs.

The curves of likelihood mean within the region of the object and background are plotted in Fig.4. Their difference and the relative position error between estimation and ground truth are also plotted for evaluations. Fig.4 shows that the position error was low when the difference was high, and vice versa, which validated our assumption that tracking conditions can be monitored by analyzing w_{obj} and diff in the likelihood image. Experiments on this sequence demonstrated that, although ensemble tracking can manage tracking in most cases, it must combine other methods to achieve a robust performance, such as embedding it into the particle filter framework in this study.

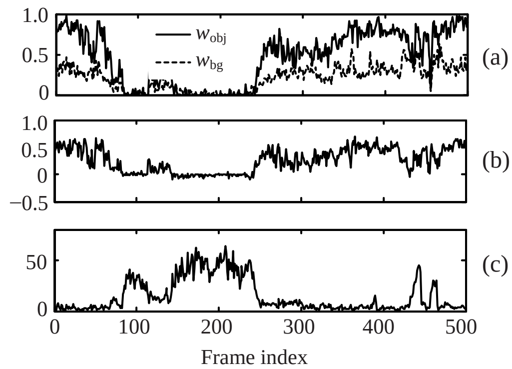


Fig.4 The curves of likelihood mean within the region of the object and background (a), their difference (b), and the relative position error (in pixels) between centroid of estimation and ground truth (c)

Results on the landmark sequence

Fig.5 presents the results on the landmark sequence, and Table 2 lists the RMSE of the three algorithms. The position error of PF was the highest for this sequence, but our algorithm still showed the best performance. The observation model in our algorithm incorporates global and local information within a region, and combines the advantage of ensemble tracking and particle filter based tracking, so it can cope with abrupt motion and illumination changes.

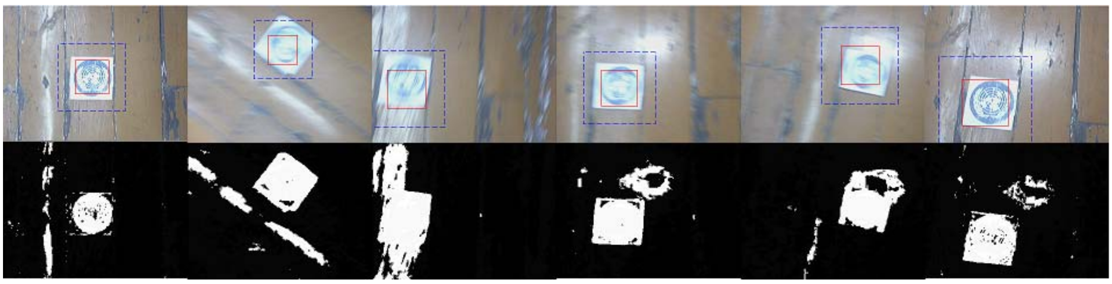


Fig.5 Results on the landmark sequence obtained using ETEPF
The challenges of this sequence include illumination changes and abrupt motion. The respective likelihood images are shown in the bottom row

Table 2 Root mean squared errors for the landmark sequence

Algorithm	RMSE (pixel)	
	u	*v*
ET	8.92	6.40
PF	9.40	8.14
ETEPF	7.87	5.73

CONCLUSION

In this paper, an algorithm of embedding ensemble tracking in a stochastic framework is proposed to achieve robust object tracking under difficult conditions. By combining their advantages, ensemble tracking and particle filter based tracking are integrated into a unified framework. The proposed algorithm can be used to track articulated objects, and has better generalization capacity, because the ensemble method can be used to combine different types of local features. Fewer particles are needed to recapture the object quickly after occlusions. Experiments on two image sequences demonstrate that, the algorithm reduces tracking position errors greatly, and performs better when embedding ensemble tracking in the stochastic framework. Although the update mechanism is not used in this study, our algorithm achieves better tracking performance than ensemble tracking. The future work is to update the ensemble in a principled way, and to test our algorithm on more videos.

References

Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, **50**(2):174-188. [doi:10.1109/78.978374]

Avidan, S., 2007. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**(2):261-271. [doi:10.1109/TPAMI.2007.35]

Bradski, G.R., 1998. Real Time Face and Object Tracking as a Component of a Perceptual User Interface. Proc. 4th IEEE Workshop on Applications of Computer Vision, p.214-219.

Collins, R.T., Liu, Y., Leordeanu, M., 2005. Online selection of discriminative tracking features. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(10):1631-1643. [doi:10.1109/TPAMI.2005.205]

Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(5):564-577. [doi:10.1109/TPAMI.2003.1195991]

Dalai, N., Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, **1**:886-893. [doi:10.1109/CVPR.2005.177]

Freund, Y., Schapire, R.E., 1996. Experiments with a New Boosting Algorithm. Proc. 13th Int. Conf. on Machine Learning, p.148-156.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting. *Ann. Statist.*, **28**(2):337-374. [doi:10.1214/aos/1016218223]

Isard, M., Blake, A., 1998a. CONDENSATION: conditional density propagation for visual tracking. *Int. J. Comput. Vis.*, **29**(1):5-28. [doi:10.1023/A:1008078328650]

Isard, M., Blake, A., 1998b. ICONDENSATION: Unifying Low-level and High-level Tracking in a Stochastic Framework. Proc. 5th European Conf. on Computer Vision, **1406**:893-908.

Jepson, A.D., Fleet, D.J., El-Maraghi, T.F., 2003. Robust online appearance models for visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(10):1296-1311. [doi:10.1109/TPAMI.2003.1233903]

Maggio, E., Cavallaro, A., 2005. Hybrid Particle Filter and Mean Shift Tracker with Adaptive Transition Model. Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing, **2**:221-224.

Odobez, J.M., Gatica-Perez, D., Ba, S.O., 2006. Embedding motion in model-based stochastic tracking. *IEEE Trans. Image Process.*, **15**(11):3514-3530. [doi:10.1109/TIP.2006.877497]

Perez, P., Hue, C., Vermaak, J., Gangnet, M., 2002. Color-based Probabilistic Tracking. Proc. 7th European Conf. on Computer Vision, **2350**:661-675.

Shan, C.F., Tan, T.N., Wei, Y.C., 2007. Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recogn.*, **40**(7):1958-1970. [doi:10.1016/j.patcog.2006.12.012]

Yilmaz, A., Javed, O., Shah, M., 2006. Object tracking: a survey. *ACM Comput. Surv.*, **38**(4):Article No. 13, p.1-45. [doi:10.1145/1177352.1177355]

Zhou, S.K., Chellappa, R., Moghaddam, B., 2004. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Process.*, **13**(11):1491-1506. [doi:10.1109/TIP.2004.836152]