Self-tuning hierarchical Kalman-particle filter for efficient target tracking

Xu Chao[1], Gao Min[1], Yang Yao[2]

(1. Missile Engineering Department, Ordnance Engineering College, Shijiazhuang 050003, China;
2. Chongqing Optoelectronics Research Institute, Chongqing 400060, China)

Abstract: Hierarchical Kalman-particle filter (HKPF) is successfully applied to target tracking with adaption to motion changes. However, it only focuses on the optimization of the target position rather than other affine parameters, resulting in many particles needed to find the optimal state. To achieve fast tracking in complex environment, self-tuning strategy-based hierarchical Kalman-particle filter was proposed to solve the problem. The proposed algorithm marginalized out the linear states in the dynamics to reduce the state dimension, and then found the optimal nonlinear states in a chainlike way with a very small number of particles. The detail process of our algorithm was as follows: first, a local region was estimated by KF; second, self-tuning strategy was used to incrementally generate particles in this region, and an online-learned pose estimator (PE) was introduced to iteratively tune them along the optimal directions according to observations. The comparison among the proposed algorithm and the existing tracking algorithms with real video sequences was implemented, in which the target undergo rapid and erratic motion, or/and dramatic pose change. The results demonstrate that the proposed tracking algorithm can achieve great robustness and very high accuracy with only a very small number of particles.

Key words: hierarchical Kalman-particle filter; self-tuning strategy; pose estimation; target tracking
CLC number: TP391 **Document code:** A **Article ID:** 1007-2276(2015)06-1942-09

自调整分层卡尔曼粒子滤波的快速目标跟踪

徐 超[1],高 敏[1],杨 耀[2]

(1. 军械工程学院 导弹工程系,河北 石家庄 050003; 2. 重庆光电技术研究所,重庆 400060)

摘 要: 分层卡尔曼粒子滤波成功应用于目标跟踪,但其只对目标位置进行了优化,忽略了其他仿射参数,导致跟踪中的粒子数目仍然很大。为了实现复杂环境下的快速目标跟踪,提出一种带有自调整策略的分层卡尔曼粒子滤波方法。该方法将目标划分为线性和非线性状态空间,并通过少量粒子的迭代过程在非线性状态空间逐步搜索最优状态。其详细过程如下:首先,利用卡尔曼滤波预测目标位置,结合目标运动信息计算潜在目标区域;然后在该区域内生成一组随机粒子,通过在线姿态估计对

收稿日期: 2014-10-05; **修订日期:** 2014-12-15
基金项目: 军内科研项目;军械工程学院科研基金(YJJM11018)
作者简介: 徐超(1987-),男,博士生,主要从事计算机视觉及图像末制导技术方面的研究。Email:475084845@qq.com
导师简介: 高敏(1963-),男,教授,博士生导师,博士,主要从事计算机视觉及图像末制导技术方面的研究。
Email:gaomin1103@gmail.com

粒子状态进行调整,并将观测结果与目标模板进行比较,修正粒子摄动的方向以逼近目标。把该方法应用于大机动目标的视频序列中,并与现有的跟踪方法进行了对比。结果表明,所提方法能够以少量粒子实现准确、稳定的目标跟踪,大大降低了跟踪算法的运算量,提高了跟踪效果。

关键词: 分层卡尔曼粒子滤波; 自调整策略; 姿态估计; 目标跟踪

0 Introduction

Particle filter (PF) is used to tackle the nonlinear non-Gaussian tracking problem by a set of random samples with associated weights[1-3]. However, PF is quite computing intensive, and the computational complexity increases quickly with the state dimension and the number of particles[4-7]. One remedy for this problem is to generate a small number of samples that can better fit the true distribution by optimizing importance function. Kwon et al[8] derived an optimal importance function from an analytical appearance measurement function. Robust tracking was achieved but the number of sampling particles was still large. Kalman-particle filter[9-10] and unscented particle filter[11-12] provide a deterministic way to sample particles. The reduction in the number of particles is often offset by added computational cost of generating samples. Another remedy for fast tracking is to reduce the dimension of the state space by marginalizing out the states appearing linearly in the dynamics. Khan and Hu et al[13-14]. applied Rao-Blackwellized particle filter (RBPF) to Eigen-tracking, in which the appearance part was handled efficiently, while the location (motion) part still remained in efficient. Unlike those approaches, Yin et al[15] proposed HKPF to partition the state space into an analytically tractable part and an intractable part, and the analytically tractable state variable was shared by all particles. HKPF noticeably improved the efficiency of practical tracking, but the optimization is centered on the position of the target rather than all states such as scale, orientation, aspect ratio and so on. An ideal solution is to tune the particles perturbation around true position. Fortunately, such an issue had been introduced in[16], called incremental self-tuning particle filter (ISPF), in which particles were incrementally drawn and then an online-learned PE was applied to guide random particles to move toward their neighboring. The major problem of ISPF is: autoregressive process can not deal with abrupt motion and the initial samples in each frame have to be distributed in a large region, in which some particles drawn randomly often are far away from the true position. In this case, PE cannot predict the pose of target successfully from background, resulting in some useless calculation.

Considering the performance of HKPF and ISPF synthetically, we propose a novel framework called self-tuning strategy-based HKPF for efficient tracking. Figure 1 gives an illustration of the detailed optimal process of traditional PF, HKPF, ISPF and the proposed framework. The process of our PF is detailedly divided into two steps: first, object motion is analyzed in a coarse-to-fine way, in which kalman filter predicts a local region around the estimation of global linear motion, and the local elliptical region is adaptive to motion change; second, the self-tuning strategy incrementally generates small number of particles in the local region, and then utilizes an online-learned PE to iteratively tune them close to the true pose according to observations. In each loop of iteration, resampling is implemented and would be terminated if the maximum similarity of all tuned particles exceeds a desired target-patch similarity trained online or if the number of particles reaches maximum. The result is that a set of particles forms a short chain in the estimation region to find the optimal state fast.

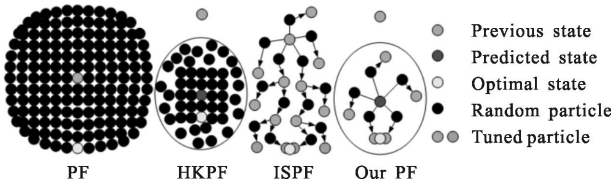

Fig.1 Detailed optimization processes of standard PF, HKPF, ISPF and our PF

1 Overview of the tracking algorithm

We describe the framework of our tracking algorithm in Fig.2, consisting of hierarchical estimation and PE framework. The hierarchical estimation includes position estimation by KF and region estimation by maneuver detection, providing a local region for random distribution of particles. Pose estimation is introduced to make the "smart" particles close to the optimal state, and in the process resampling is implemented iteratively until a desired target-patch similarity is obtained. The final state would be feed back to KF in the hierarchical estimation as an observation.

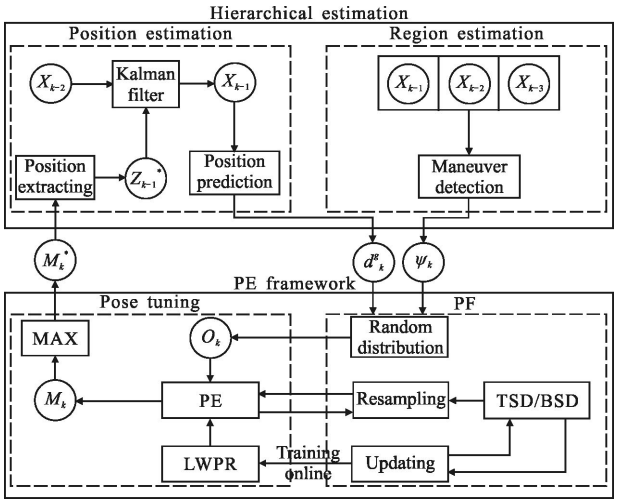

Fig.2 Framework of our tracking algorithm

The global linear m otion state of object is described by $X_k=(x_k, y_k, \dot{x}_k, \dot{y}_k, \ddot{x}_k, \ddot{y}_k)^{\mathrm{T}}$, where x_k, y_k, $\dot{x}_k$, $\dot{y}_k$, $\ddot{x}_k$, $\ddot{y}_k$ denote horizontal and vertical position, velocity and acceleration at time k, respectively. The local non-linear terms of motion in PE framework is represented by 2-D affine translation parameters $Y_k=(\Delta x_k, \Delta y_k, \theta_k, S_k, \alpha_k, \phi_k)^{\mathrm{T}}$ where Δx_k, Δy_k, θ_k, S_k, α_k, ϕ_k denote local dis-placements, orientation, scale, aspect ratio, and skew direction. Then, the state Y_k is rewritten as a 2-D affine transformation matrix

$$M_k=\begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix} \tag{1}$$

Where, A is a nonsingular 2×2 matrix and $b \in A^2$, denotes local displacements. Given the elliptical region by KF, PF randomly draws a little number of informative samples on the affine group, and then an online-learned PE is used to tune particles to their neighboring. In this pose-tuning process, appearance model containing target- patch similarity distribution (TSD) and background-patch similarity distribution (BSD) is needed to judge whether the random particles are moving toward the correct directions. TSD is the similarity distribution of target patches and denoted by $N_T(\mu_T, \sigma_T)$. BSD is similarity distribution of background patches and described as $N_B(\mu_B, \sigma_B)$ respectively. If the maximum similarity between all tuned particles and appearance model is smaller than $\mu_T-\sigma_T$, we will resample sparse particles for next pose-tuning process by PE. The iteration would not stop until the threshold is satisfied, i.e., the maximum similarity score $S_k>\mu_T+\sigma_T$ or the predefined number of loop is achieved. The optimal state M_k^* is defined as the maximum confidence particle and calculated as

$$M_k^*=M_k^{(\mathrm{argmax}(S_k^i))} \tag{2}$$

Where, i is the subscript of the particles, and M_k^* is the output of fine estimation and should be feed back into coarse estimation as a measurement of KF. Note that the needed observations Z_k^* are two positional elements extracted from M_k^*. If the final maximum similarity score S_k is larger than $\mu_B+\sigma_B$, TSD, BSD and appearance model (target template) should be updated based on the estimated state M_k^*. Moreover, LWPR for pose-tuning is also trained online by new samples generated from M_k^*.

2 Self-tuning hierarchical Kalman-particle filter

The hierarchical estimation is referring to[9], and we do not describe the detail pleonastically. Our focus is on the fine estimation. Thus, the affine transform is used to model local tracking by the state transition from M_{k-1} to M_k between two consecutive frames. As men- tioned above, the fine state is modeled by 2D transfor- mation matrix, often described by six basis elements

$$A_1=\begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix}\quad A_2=\begin{bmatrix}1&0&0\\0&-1&0\\0&0&0\end{bmatrix}\quad A_3=\begin{bmatrix}0&-1&0\\1&0&0\\0&0&0\end{bmatrix}$$

$$A_4=\begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix}\quad A_5=\begin{bmatrix}0&0&1\\0&0&0\\0&0&0\end{bmatrix}\quad A_6=\begin{bmatrix}0&0&0\\0&0&1\\0&0&0\end{bmatrix}\tag{3}$$

Brownian motion model is introduced for the particle filter in the local region, and the local dynamic model is expressed by a nonlinear state transition equation evolving on the affine group

$$M_k=\underbrace{\exp(\sum_{i=5}^{6} d_k^g(i-4)\cdot A_i)}_{\text{Llinear}}\cdot\underbrace{h(M_{k-1},\ \Omega_k)}_{\text{Nonlinear}}\tag{4}$$

$$h(M_{k-1},\ \Omega_k)=M_{k-1}\cdot\exp(\Omega_k)\tag{5}$$

The first term of the right in Eq.(4) is a pure linear term that represents the translation of the previous state at time $k-1$, where $d_k^g(\cdot)$ is the global displacement predicted by KF, A_i is the basis element in Eq.(3). $h(M_{k-1},\ \Omega_k)$ implies the dynamics of natural motion, in which the non-linear motion is assumed to be restricted in a local region and considered as Brownian motion. Ω_k denotes the process noise of nonlinear filter determined by region estimation, calculated by algebraic expression $\sum_{i=5}^{6} w_{k,i}A_i$ with $w_k=\{w_{k,1},\ w_{k,2},\ w_{k,3},\ w_{k,4},\ w_{k,5},\ w_{k,6}\}$ sampled from the Gaussian distribution $N(0,\ \psi_k)$, ψ_k is covariance matrix of local region.

In HKPF, a small number of particles are generated randomly after hierarchical estimation to find the best estimate of target state. Restricted in a local region, those particles are not sparse concerned with translation, but the parameters of pose are not expressed exactly. By using a PE framework, we adjust each particle for optimal pose iteratively and make accurate and time-efficient tracking possible. Algorithm Self-tuning process of particles shows the whole process in detail. PE framework addresses the tracking problem using piecewise strategy. Firstly, The initial particles are drawn by Eq.(4), and the number of samples ΔN_0 for each frame is usually 10% of the maximum number $N_{\max}$. These particles are tuned to their neighboring best states M_k^t iteratively, see Step(2), where, t is the index number of loop for pose tuning and $t=0,1,2,\ \cdots,L_{\max}$; $L_{\max}$ is the maximal loop determined by step 4) b). Secondly, ΔN_{t+1} particles M_k^{t+1} will be added by resampling from previous optimized particles $(M_k^t,\ W_k^t)$, see Step(4)-1), where $W_k^t=\{w_{k,i}^t\}_{i=1}^{\Delta N_t}$ is the normalized weights of the particles in M_k^t. The resampling process makes the incremental particles concentrate on the tuned particles with large weights. These newly particles are propagated by a Gaussian motion model concern with the value of similarity, see Step (4)-4), which means the uncertainty of Gaussian model will be enlarged if the similarity is small and vice versa. Such iteration for gradual optimization of state would be stopped until the maximum similarity score of tuned particles exceeds the predefined threshold, i.e., $S_{\max}^t<\mu_T-\sigma_T$, see Step(5). Since we set a terminating condition to make the iteration stop in advance while achieving the satisfied similarity score, see Step(4)-5), not all particles need to be calculated. If the final similarity score is larger than $\mu_B+\sigma_B$, all online models are updated, see Step (7).

Algorithm Self-tuning process of particles

Input: maximum number of particles $N_{\max}$, initial number of particles ΔN_0, increment factor α_{inc}, previous particle states M_{k-1}, $S_{\max}^0=0$.

Process:

(1) Draw ΔN_0 particles $M_{k-1}=\{M_{k-1},\ i\}_{i=1}^{\Delta N_0}$ and then

propagate them to new states $M_k^0=\{M_{k,i}^0\}_{i=1}^{\Delta N_0}$ according to state transition equation.

(2) While ($i<=\Delta N_0$ and $S_{\max}^0<\mu_T-\sigma_T$) do

Tune $M_{k,i}^0$ to the best neighbor $M_{k,i}^0$, and assign M_k^0 accordingly. Computer S_i^0=Sim (O ($M_{k,i}^0$)) and $S_{\max}^0$ = max$\{S_{\max}^0, S_i^0\}$;

End

(3) $n=\Delta N_0$, $t=0$.

(4) While $S_{\max}^t<\mu_T-\sigma_T$ and $n<N_{\max}$ do

1) $t=t+1$;

2) $S_{\max}^t=0$; $\Delta N_t=\min(\alpha_{\text{inc}}\Delta N_{t-1}, \Delta N_{\max}-n)$;

3) Draw ΔN_t particles $\Delta M_k^t=\{M_{ki}^t\}_{i=1}^{\Delta N_t}$ by re-sampling from(M_k^{t-1}, W_k^t);

4) Propagate new particles; for each particle, $M_{k,i}^t=M_{k,i}^t\exp(v_{k,i}^t)$ with $v_{k,i}^t\sim N(0, \Sigma_{k,i}^t)$, where

$$\Sigma_{k,i}^t=(1-S_{\max}^{t-1})\psi_C$$

5) While ($i<=\Delta N_t$ and $\Delta S>T_{\text{SI}}$) do

If ($S_i>\mu_T-\sigma_T$)

Calculation terminates;

End

Tune $M_{k,i}^t$ to the best neighbor $M_{k,i}^t$, and assign ΔM_k^t accordingly. Compute S_i^t=Sim($O(\widetilde{M}_{k,i}^t)$) and $S_{\max}^t$ = max$\{S_{\max}^t, S_i^t\}$;

End

6) $n=n+\Delta N_t$;

7) Concatenate all tuned particles: $M_k^t=\{M_k^{t-1}, \Delta M_k^t\}=\{M_{k,i}^t\}_{i=1}^n$;

End

(5) $S_{\max}^t=\max\{S_{\max}^j\}_{j=0}^t$;

(6) $M_k^*=M_{k,i}^t|_{\text{Sim}(O(M_{k,i}^t))=S_{\max}^t}$, $\widetilde{S}_k=S_{\max}^t$.

(7) If $\widetilde{S}_k>\mu_B-\sigma_B$

Update: target template, LWPR, and TSD/BSD.

End

Output: M_k^*

3 Experiment results

3.1 Experiment setup

Experiments are implemented to verify the effective-ness and efficiency of the proposed algorithm. The videos are "Cock", "Sylvester", and "Dudek". We employ HKPF, ISPF and our PF for those real tracking and analyze the experiment results to find well-behaved tracking algorithm under these different circumstances. Therein, "Cock" describes a toy with abrupt and erratic motion while the "Sylvester" shows great illumination changes and severe pose changes. To further evaluate our framework quantitatively, "Dudek" face sequence is introduced with fast motion and pose change.

Some important parameters of our algorithm are set as follows: the maximum number of particles $N_{\max}$= 50; the number of initial particles in each frame is ΔN_0 =5; the particle incremental factor α_{inc} =2. The image patch is usually resized to 36 ×36. Various parameters of ISPF are referring to[10]. It also gives the detail of initialization and updating of LWPR. Our algorithm does not discuss this procedure and mainly verifies the performance on dynamic affine transformation.

3.2 Performance of HKPF, ISPF and our PF

The tracking results for sequence "Cock" are shown in Fig.3. ISPF is trapped in the cluttered background at about frame #148 due to a fast and sudden motion and the blear image resulting from severe camera vibration. Under this circumstance, most particles are far from the ground truth, seldom samples at the neighborhood of target are often moving toward wrong direction, because the input feature vectors of LWPR are beyond the regression scope of the learned model. HKPF also could track the target robustly and accurately with as much as 150 particles per frame. On the contrary, our PF achieves best performance with least particles. Note that in each loop of resampling, not every particle

need to be computed if the similarity of any one satisfies the threshold as described in Algorithm 1, which makes the number of particles needed is further reduced. The mean number per frame is only 28.2 for this sequence.

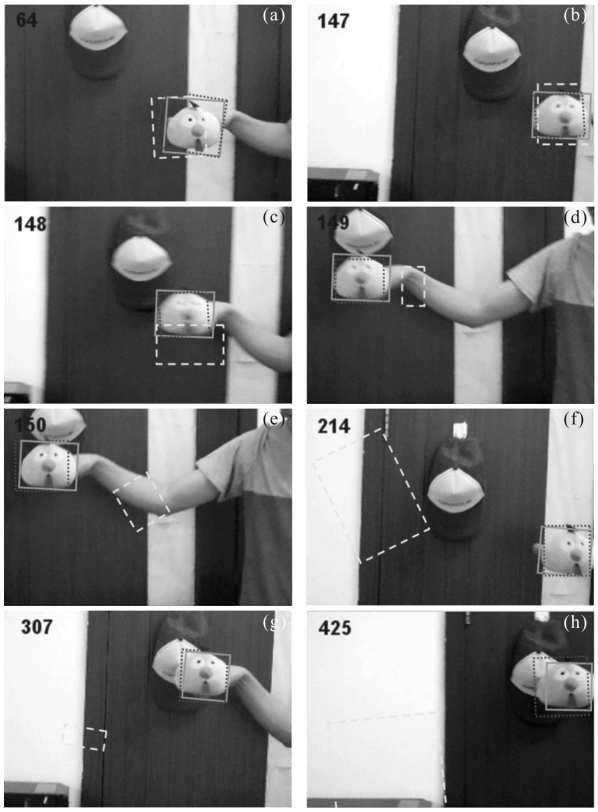

Fig.3 Tracking results by our PF , ISPF and HKPF for "Cock" sequence

The tracking results for "Sylvester" sequence are shown in Fig.4. At the frame #76, HKPF with 100 particles tracks the target with a big error on orientation and finally fall into failure at the frame #269 due to severe pose change. ISPF solve the problem easily because the ISPF tracker incrementally generates many position hypotheses; as long as one of the hypotheses is a good bridge state to the optimal state, the global optimal state can be found in a chainlike way effectively by state propagation and pose tuning. Only the frame #107 and the follows with target motion trouble ISPF needs slightly more particles. Our PF tracker achieves similar tracking results and could follow the target very tightly. In such a sequence without dramatic motion and blear target, the mean number of particles is reduced from 13.2 by ISPF to 9.7 by our PF. For one reason, the local region is smaller to some extent by hierarchical estimation; for another reason, the resampling will be stopped in time if any particle is close to the target rather than computing all particles. At the first sight, the improvement of our PF is not significant, but the calculation of the algorithm does not just depend on the number of particles, in which the pose-tuning process of each particle also costs much time because the feature vector should be extracted from the template. Fortunately, PE of a random particle in our PF would execute as described in Algorithm 1, which means less calculation of our PF even with the same number of particles as ISPF.

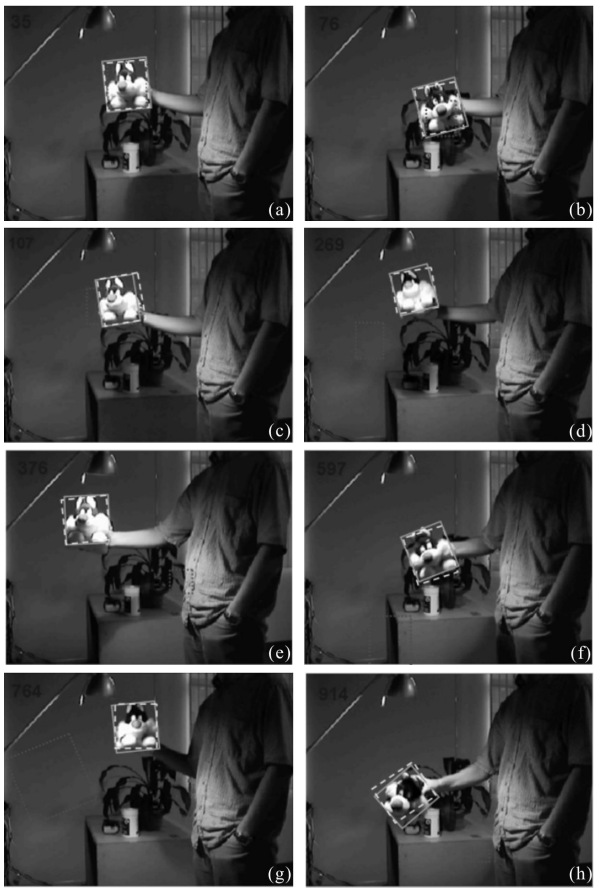

Fig.4 Tracking results by our PF, ISPF and HKPF on "Sylvester" sequence

3.3 Quantitative analysis

The above sequences have verified the performance of the three algorithms under different

environments qualitatively. For further quantitative evaluation, "Dudek" sequence by proper affine transformation is adopted. To analyze the detail of the tracking results, only the center point of the object is not enough, and seven feature points on the eyes and mouse are usually used to describe the position and pose of target. Considering the non-rigid motion of the face, the seven feature points could not represent the tracking results perfectly, which means that the estimated seven feature points maybe not match the ground truth well although the algorithm seize the target tightly. Hence, we retain the center point and use the weighted results as the standard for quantitative evaluation. Fig.5 describes the detail, and as shown in this figure we manually labeled seven ground-truth feature points and the center point in every frame, i.e., denoted as $\{P_k^i\}_{i=1}^7$ and C_k. The estimated feature points at k frame $\{\hat{P}_k^i\}_{i=1}^7$ are computed by $\{\hat{P}_0^i\}_{i=1}^7$ and affine matrix from M_0 to M_k, where $\{\hat{P}_0^i\}_{i=1}^7$ are set equal to the ground- truth in the first frame $\{\hat{P}_0^i\}_{i=1}^7$. $\hat{C}_k$ is the center of the tracking box. We use RMS error to represent the tracking accuracy:

$$\mathrm{RMS}(k)=\sqrt{1/14\sum_{i=1}^{7}\|\hat{P}_k^i-P_k^i\|^2+1/2\|\hat{C}_k-C_k\|^2} \quad (6)$$

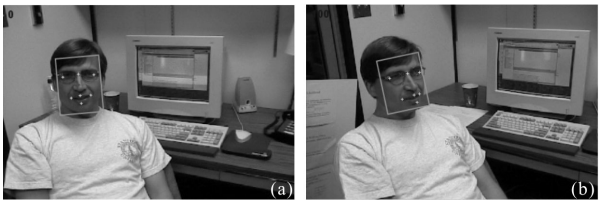

Fig.5 Representive examples with true feature points marked "+", calculated feature points marked "×", and the box surrounding the center point

The representative "Dudek" face sequence tracking results by three algorithms and the errors between the estimation and ground truth are shown in Fig.6 and Fig.7, respectively. As shown in Fig.6, HKPF with 50 particles tracks the face in the whole process, but the scale and orientation of the tracking box do not match the target accurately, especially at frame #689, #880 and #1010 with large pose changes. It can also be seen that the errors are inferior to that of another two trackers from Fig.7. ISPF develops such a precise tracker that it is hardly distinguished from our PF in Fig.6 and Fig.7. However, the disadvantages of ISPF are emerged in Fig.8, which describes the number of particles used by ISPF and our PF in each frame. Obviously, ISPF draws more particles than the proposed approach, and the trend is along the change of target motion, shown at frame #570, #689, #1010 and so on. Our PF tracker still maintains a small set of very effective particles, consuming only 8.6 particles per frame while ISPF drawing 15.5 particles in each frame.

Fig.6 Tracking results by our PF, ISPF and HKPF on "Dudek" sequence

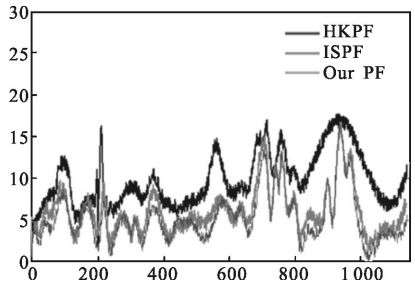

Fig.7 RMS error curves of HKPF, ISPF and our PF in "Dudek" face sequence

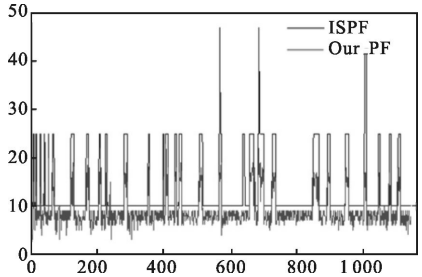

Fig.8 Number of particles consumed by ISPF and our PF for "Dudek" face sequence

4 Conclusions

Combining the advantages of HKPF and ISPF, we propose an algorithm called self-tuning hierarchical kalman-particle filter for fast target tracking. The tracking process is divided into two steps: hierarchical estimation and PE framework. Hierarchical estimation is to reduce the searching region and eliminate the influence of dramatic motion.. PE framework in the local region is to adapt the pose change by tuning the particles in each loop towards the optimal state. Experimental results show that: HKPF is prefer to dramatic motion rather than pose change, whereas ISPF is effective for pose variation, and could also solve the non-significant maneuver to some extent with sacrificid efficiency. However, this approach fails absolutely when the target moves too quickly, especially when the feature of target changes significantly. Our PF over performs another two algorithms and could provide accurate and efficient tracking results under various affine transformations, i.e., motion and pose changed dramati-cally, even blear images caused by sudden maneuver. Moreover, our algorithm can be extended to video surveillance for city live, which contains pedestrian and car tracking. Pedestrian tracking is verified effectiveness in this paper and car tracking would be studied in future.

Pose tuning process is sensitive to the change of target template, and we need improve the updating manner of template to insure the input feature vectors of LWPR included in the regression scope of the learned model.

References:

[1] Wang Fasheng, Lu Mingyu, Zha o Qingjie, et al. Particle filtering algorithm [J]. ***Chinese Journal of Computers***, 2014, 37(8): 1679–1694. (in Chinese)

[2] Thomas V, Ray A K. Fuzzy particle filter for video surveillance[J]. ***IEEE Transactions on Fuzzy Systems***, 2011, 19(5): 937–945.

[3] Shana C, Tan T, Wei Y. Real-time hand tracking using a mean shift embedded particle filter [J]. ***Pattern Recognition***, 2007, 40: 1958–1970.

[4] Tian Li, Zhou Fugen, Meng Cai. Parallel particle filter object tracking based on embedded multi-core DSP systems [J]. ***Infrared and Laser Engineering***, 2014, 43(7): 2354–2361.

[5] Cao Yang, Zhao Mingfu, Luo Binbin, et al. Airborne platform′s tracking algorithm for free space optical [J]. ***Infrared and Laser Engineering***, 2012, 41(11): 3065–3068.

[6] Cheng H Y, Hwang J N. Adaptive particle sampling and adaptive appearance for multiple video object tracking [J]. ***Signal Processing***, 2009, 89: 1844–1849.

[7] Bimbo A D, Dini F. Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation[J]. ***Computer Vision and Image Understanding***, 2011, 115: 771–786.

[8] Kwon J, Lee K M, Park F C. Visual tracking via geometric particle filtering on the affine group with optimal importance functions [C]//IEEE CVPR, 2009: 991–998.

[9] Li P, Zhang T, Pece A. Visual contour tracking based on particle filters[J]. ***Image Vis Comput***, 2003, 21(1): 111–123.

[10] Xia Nan, Qiu Tianshuang, Li Jingchun, et al. A nonlinear filtering algorithm combining the kalman filter and the particle filter [J]. ***Acta Electronic Sinca***, 2013, 41(1): 148–152. (in Chinese)

[11] Rui Y, Chen Y. Better proposal distributions: object tracking using the unscented particle filter [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2001, 2: 786–793.

[12] Li Quan, Zhao Xunjie, Peng Qingyan, et al. Windows adaptive particle filter algorithm based on principal component analysis [J]. ***Infrared and Laser Engineering***, 2014, 43(10): 3474–3479. (in Chinese)

[13] Khan Z, Balch T, Dellaert F. A Rao–Blackwellized particle filter for Eigen tracking [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2004: 980–986.

[14] Hu Zhentao, Liu Xianxing, Hou Yandong. Real-time marginalized particle filter based on weights consistency optimization [J]. ***Acta Electronica Sinca***, 2014, 42 (10): 1970–1976. (in Chinese)

[15] Yin S, Na J H, Choi Y J, et al, Hierarchical kalman-particle filter with adaptation to motion changes for object tracking [J]. ***Computer Vision and Image Understanding***, 2011, 115: 885–900.

[16] Li M, Tan T, Chen W, et al. Efficient object tracking by incremental self-tuning particle filtering on the affine group [J]. ***IEEE Transactions on Image Processing***, 2012, 21(3): 1298–1313.