On-Line Ensemble SVM for Robust Object Tracking

Min Tian[1], Weiwei Zhang[2], and Fuqiang Liu[1]

[1] Broadband Wireless Communication and Multimedia Laboratory,
Tongji University, Shanghai, China
[2] Microsoft Research Asia, Beijing, China
tminshanghai@yahoo.com.cn, weiweiz@microsoft.com,
liufuqiang@mail.tongji.edu.cn

Abstract. In this paper, we present a novel visual object tracking algorithm based on ensemble of linear SVM classifiers. There are two main contributions in this paper. First of all, we propose a simple yet effective way for on-line updating linear SVM classifier, where useful "Key Frames" of target are automatically selected as support vectors. Secondly, we propose an on-line ensemble SVM tracker, which can effectively handle target appearance variation. The proposed algorithm makes better usage of history information, which leads to better discrimination of target and the surrounding background. The proposed algorithm is tested on many video clips including some public available ones. Experimental results show the robustness of our proposed algorithm, especially under large appearance change during tracking.

1 Introduction

Visual tracking is an important subject in computer vision with a variety of applications. One of the main challenges that limit the performance of the tracker is appearance change caused by the variances of pose, illumination or view-point. In order to develop a robust tracker, lots of former work has been done to address those problems, however, robust object tracking still remains a big challenge.

Object tracking can be considered as an optimization problem. Tracking algorithm is used to find a region with the local maximum similarity score. In [1], the similarity is defined as the SSD between the observation and a fixed template. In [6], Mean-shift is proposed as a nonparametric density gradient estimator to search the most similar region by computing the similarity between color histogram of the target and the search window. Object tracking can also be considered as a state estimation problem. In early works, Kalman filter or its variants are frequently used. However, Kalman can't solve the non-Gaussian and non-linear cases well. In order to solve the non-Gaussian and non-linear cases well, sequential Monte Carlo methods are applied for tracking, among which Particle Filter (PF) [3,8,12,4] is the most popular one. Object tracking can also be regarded as a template updating problem. The classical subspace tracking method is proposed by Black et al. [2]. Ross and Lim [13] extended Eigen-tracking by on-line incremental subspace updating. Along the other direction, in [9] Jepson models the target as a mixture of stable component, outliers, and two frame transient component. And an on-line EM algorithm is use for the parameters of each

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 355–364, 2007.

component. Considering the tracker as a binary classify problem is very popular recently. [18] proposed a transductive learning method for tracking, in which D-EM algorithm is used for transducing color classifiers and selecting a good color space to determine each pixel whether belongs to the foreground or background. The constraints of these color trackers are clear because they can only work on color image sequences. In [10] an off-line SVM is used for distinguish the target vehicle from the background. Similar to [10], in [12] an Adaboost classifier is trained off-line to detect the hockey players for a proposal distribution to improve the robustness of the tracker. Since they need large mount training data, it's not easy to extend those approaches to general objects tracking. In order to get a general tracker, [16] proposed an online learning of ensemble pixel based weak classifier, and tracking is done by classifying pixels into foreground and background. However, in his method the features are limited to an 11D low dimension vector including pixel colors and local orientation histogram. Helmun Grabner proposed an on-line version of the Adaboost algorithm in [17], which can on-line select features to get a strong classifier. Their work is roughly similar to the idea of [16], however it can on-line choose the features with the most discriminative ability.

Among all of the topics that have been discussed in existing classifier based trackers, we found that the history information has not been paid much attention, which is very important for tracking. In order to better use those important information, we proposed an ensemble SVM classifier based tracking algorithm. In our algorithm, the linear SVM can automatically select the "Key Frames" of the target as support vectors. Moreover, through ensemble combining several linear SVM classifiers, history information can be used more reasonable, and the risk of drift can be decreased more effectively. Finally, because the ensemble method can automatically adjust each SVM classifiers weight on-line, so some off-line classifiers can also be trained and add into the framework to form a more robust tracker.

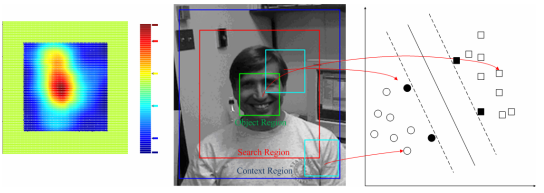
The paper is organized as following, section 2 give out the proposed on-line updating of SVM based tracker. Section 3 give out the ensemble of SVM based tracker. Experiment and conclusion are given in section 4 and section 5 respectively.

2 On-Line SVM Tracker

In this paper, we take the tracking as a binary classification problem, where we choose SVM as our basic classifier. In the following sections, we will show that SVM can automatically select "Key frame" of the target from the historic frames, and that is one of the most important factors for the proposed algorithm in this paper. First of all, let's introduce our on-line SVM classifier based tracker.

2.1 SVM Classifier Based Tracking

Within the context of object tracking, we define the target object region and its surroundings as positive data source and negative data source respectively, as shown in fig. 1(b). Our target is to learn a SVM classifier which can classify the positive data and negative data in the new frame. Starting from first frame, the positive and the negative samples are used to training the SVM classier. Then the search region (fig. 1(b)) can be estimated in the next frame. Finally, the target region in next frame is located with local maximum score within the search region.



(a) (b) (c)

Fig. 1. (a) The confidence map of the search region. (b) The object region, search region and the context region. (c) Demonstration of a linear SVM. The filled circles and rectangles are the "support vectors".

2.2 On-Line Updating Linear SVM

One of the most difficult tasks for a tracker is how to on-line update the tracker to make it adapt to the appearance change of the target. Some former methods use a rapid update model, saying that $\mathbf{x}_t = \hat{\mathbf{x}}_{t-1}$, but it's dangerous and may cause drift problem [11]. Some off-line tracker, such as [7], takes benefit of the off-line selected "Key Frames", which will let the tracker be more robust to against drift problem. Our idea is inspired by the former ones, and the difference is that our tracker can not only record the "Key Frames" of the target as the history information, but can also update on-line to decrease the risk of drift.

We consider the updating of the classifier as an on-line learning process. Here we propose a simple yet effective way for on-line updating the linear SVM classifier. The details are described as Algorithm 1.

Algorithm 1. On-line Linear SVM Tracking & Updating

Input: I_n Video frames for processing (n=1,…, L)
R Rectangle region of the target region
Output: Rectangles of target object's region R_n (n=1,…, L)

Initialization for the first frame I_n (n=1):

- Extract positive and negative samples $S_1 = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $y_i \in \{-1,+1\}$, corresponding to the target region R.
- Train a linear SVM to get $f_1(\mathbf{x}) = \mathbf{w}_1\mathbf{x} + b_1$ and its support vectors $V_1 = \{\mathbf{x}_i, y_i\}_{i=1}^{M}$

For each new frame I_n (n>1):

- Find region R_n with the local maximum score given by $f_{n-1}(\mathbf{x})$. Here $\mathbf{x}$ denotes the search window's feature vector.

$$\mathbf{x}_{R_n} = \arg\max_{\mathbf{x}} f_{n-1}(\mathbf{x}) = \arg\max_{\mathbf{x}} (\mathbf{w}_{n-1}\mathbf{x} + b_{n-1})$$

- If $f_{n-1}(\mathbf{x}_{R_n}) > 0$ go to the next step to get a new SVM.
  Else stop updating and guess R_n is the target region, and go to the next frame.
- Refresh positive samples $P_n = V_{n-1}^{+} \cup S_n^{+}$ and negative samples $N_n = V_{n-1}^{-} \cup S_n^{-}$. Here, V_{n-1}^{+} and V_{n-1}^{-} are the positive and negative support vectors of $f_{n-1}(\mathbf{x})$, S_n^{+} and S_n^{-} are the positive and negative samples of current frame.
- Retrain the SVM using new samples for updating to get $f_n(\mathbf{x}) = \mathbf{w}_n\mathbf{x} + b_n$

By on-line updating, the SVM tracker can adjust its hyper-plane for the maximum margin between the new positive and negative samples. The support vectors transferred frame by frame contain important "Key Frames" of the target object in the previous tracking process (see fig.2 (b)). Figure 2 show the tracking result and part of selected "Key frames" by SVM in the final frame, the video is provided by Jepson [9]. The proposed tracker can adapt to the face with variation in appearance and distinguish it from the cluttered background (fig. 2(a)).

Fig. 2. (a) Tracking results of frame 1, 206, 366, 588, 709, 761, 973 and 1131. (b) In the end of the tracking task, 182 positive support vectors contain enough history information. Images displayed on the bottom are some of these support vectors.

3 Ensemble SVM Classifier Based Tracking

Although the proposed SVM based tracker is powerful for tracking by on-line updating, we found there still existing several issues need to be addressed in the real world video clips. (a) The variance of the target is very large. (b) The tracker is disturbed by scale variation, partial occlusion or movement blur on a certain frame.

Those two issues may lead the tracking algorithm drifting and finally fail. In order to further address those two issues, we proposed an ensemble SVM algorithm in this section.

3.1 On-Line Building the Ensemble of SVMs

Our algorithm starts with a SVM trained with labeled data in the first frame. After then, in each frame new SVM may be added, the current tracking result with previous SVM classifiers. The match ratio r_m is defined as equation (1), where $U(x)$ is a step function that equals to 1 when x is above zero, and otherwise it equals to 0. Here $\{\mathbf{x}_k\}_{k=1}^{N^+}$ are the positive samples, and N^+ is their number. The larger r_m is, the

better current component matches the positive samples. If r_m below a ratio threshold, a new SVM should be added.

$$r_m = \frac{\sum_{k=0}^{N_+} U(f_m(\mathbf{x}_k)-1)}{N_+ - \sum_{k=0}^{N_+} U(f_m(\mathbf{x}_k)-1)} \tag{1}$$

So after several frames, many SVM classifiers are generated and updated during different periods, which is shown as fig. 3.

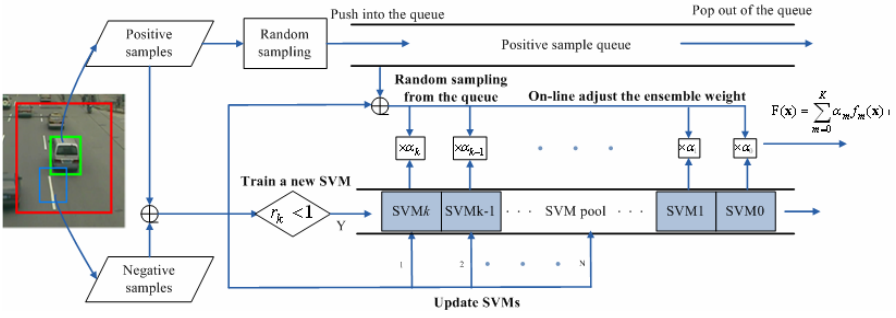


Fig. 3. This flowchart is the demonstration of our framework of on-line ensemble SVM Tracker

After the number of SVM classifier is larger than one, we combine the linear SVM classifiers in the pool to get better classify result. Each SVM classifier is assigned with a coefficient α_m , which is defined as following:

$$\alpha_m = \frac{1}{2}\log\frac{\sum_{i=1}^{N} U(\mathrm{P}_m(\mathbf{x}_i, y_i)\cdot\omega_i)}{\sum_{i=1}^{N} U(-\mathrm{P}_m(\mathbf{x}_i, y_i)\cdot\omega_i)} \tag{2}$$

Here ω_i is the samples' weight. And $\mathrm{P}_m(\mathbf{x}_i, y_i)$ is the output of each SVM classifier to evaluate its discriminative ability for every sample. Here we define $\mathrm{P}_m(\mathbf{x}_i, y_i)$ as following:

$$\mathrm{P}_m(\mathbf{x}_i, y_i) = \begin{cases} 1 & if\ \ y_i \cdot f_m(\mathbf{x}) \geq 1 \\ \dfrac{(|f_m(\mathbf{x})|-T)\cdot|f_m(\mathbf{x})|-T|}{T^2} & if\ 1 > y_i \cdot f_m(\mathbf{x}) > 0 \\ -1 & if\ \ y_i \cdot f_m(\mathbf{x}) \leq 0 \end{cases} \tag{3}$$

When $\mathrm{P}_m(\mathbf{x}_i, y_i)$ is positive, it means the right classifying probability. Meanwhile, when it is negative, it means the wrong classifying probability. Here, $T \in (0,1)$ is a threshold in the determining the rule, and we set is as 0.5 in our method. The details of on-line ensemble SVM tracker are described as Algorithm 2.

Algorithm 2. On-line Ensemble SVM Classifiers for Tracking

Input: I_n Video frames for processing (n=1,…, L)
R Rectangle region of the target region
Output: Rectangles of target object's region R_n (n=1,…, L)

Initialization for the first frame I_n (n=1):

- Use the target and other random chosen regions without overlap with the target in the first frame to form a ground truth classifier $f_0(\mathbf{x}) = \mathbf{w}_0\mathbf{x} + b_0$, which will not be updated during tracking. This classifier can also be other off-line trained classifier.
- Extract the positive and the negative samples as **Algorithm 1.**
- Train a SVM classifier $f_1(\mathbf{x}) = \mathbf{w}_1\mathbf{x} + b_1$ by using the extracted samples.
- Initialize the samples' weights $\omega_i = 1/N$, $i = 1, 2, ..., N$.
- For m = 0 to 1
  a) Make $\{\omega_i\}_{i=1}^N$ a distribution
  b) Chose the most strong SVM with the largest α_m by using equation (2)
  c) If α_m <0, α_m =0 and break
  d) Remove the chosen SVM
  e) Update samples' weight $\omega_i = \omega_i \exp[-\alpha_m \cdot y_i \, \mathrm{P}_m(\mathbf{x}_i, y_i)]$, $i = 1, 2, ..., N$
- Normalize α_i to make $\sum_i \alpha_i = 1$. The output of the ensemble one is $\mathrm{F}(\mathbf{x}) = \alpha_0 f_0(\mathbf{x}) + \alpha_1 f_1(\mathbf{x})$

For each new frame I_n (n>1):

- Use F($\mathbf{x}$) to search the target region and extract samples $S = S^+ \cup S^-$.
- Push some of S^+ into the positive sample queue by random sampling.
- Check whether a new SVM should be built by r_m in (1).
- Choose the last K SVMs to update as **Algorithm 1**. Here we set K=5.
- Radom chose M samples S' from the sample history queue, M equals to the number of S^+ . New group of samples are determined as: $S'' = S \cup S'$
- Initialize the samples' weights $\omega_i = 1/N$, $i = 1, 2, ..., N$.
- For m = 0 to $K_{\max}$ ($K_{\max}$ =10 in ours)
  a) Make $\{\omega_i\}_{i=1}^N$ a distribution
  b) Chose the most strong SVM with the largest α_m by using equation (2)
  c) If α_m <0, α_m =0 and break
  d) Remove the chosen SVM from the SVM queue
  e) Update samples' weights $\omega_i = \omega_i \exp[-\alpha_m \cdot y_i \, \mathrm{P}_m(\mathbf{x}_i, y_i)]$, $i = 1, 2, ..., N$
- Normalize α_i to make $\sum_i \alpha_i = 1$. The output of the ensemble one is: $\mathrm{F}(\mathbf{x}) = \sum_{m=0}^{K} \alpha_m f_m(\mathbf{x})$

Compared with a single on-line SVM, the ensemble tracker can get a more reliable result, especially when the appearance of the target changes frequently (as fig. 4,5). From fig. 4 and fig.5, we can clearly find that a single on-line SVM is useful. However, it record all the history information as its support vectors to achieve the global optimization, which may cause it can difficultly handle large appearance variation in a short period. This phenomenon is also appeared in the incremental

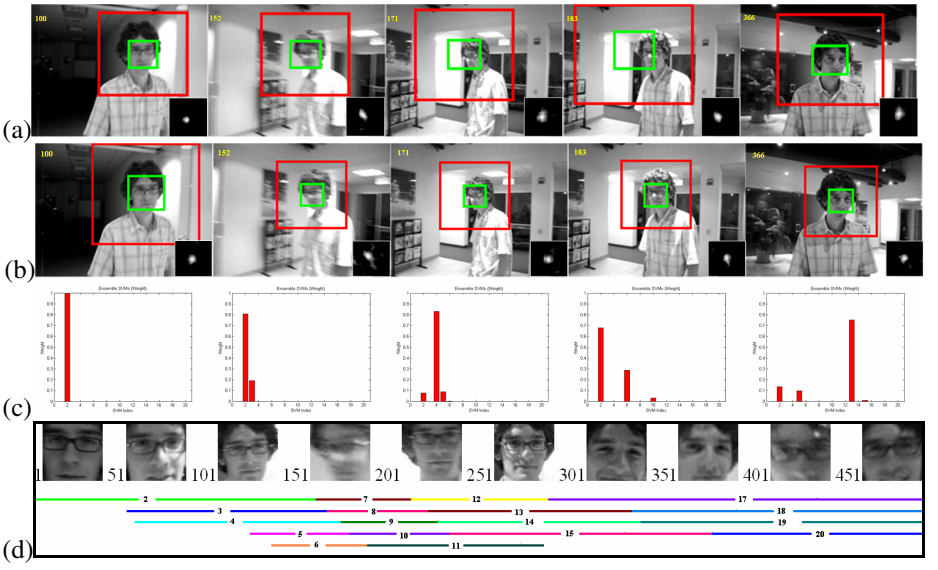


Fig. 4. (a) Tracking results of frame 100,152,171,183 and 366 by using single on-line linear SVM tracker. (b) Tracking results of on-line ensemble SVMs tracker. The confidence map of the search region is on the right-bottom of each frame. (c) The ensemble weight of each SVM in the mixture model. (d) The updating period of each SVM (from being generated to stop updating). Mind that No.1 SVM is the ground truth SVM, and it will not be updated during tracking.

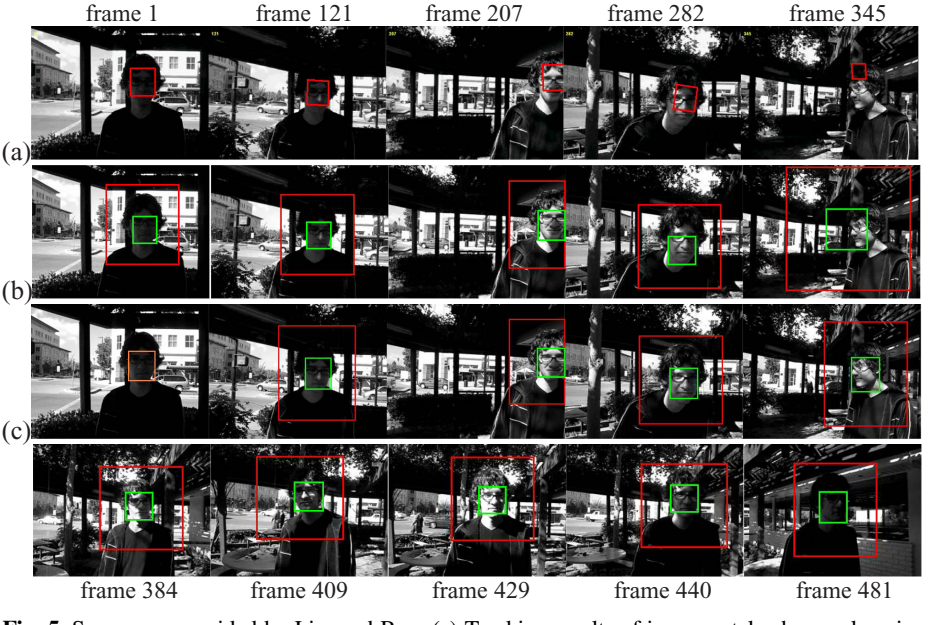


Fig. 5. Sequences provided by Lim and Ross (a) Tracking results of incremental subspace learning tracker. The tracker failed after frame 345. (b) Tracking results of our single on-line linear SVM tracker. The tracker almost failed on frame 345, and then it drifts away from the target. (c) Tracking results of on-line ensemble SVMs tracker. The tracker finished the whole video with accurate results.

subspace learning tracker shown as fig.5 (a). The ensemble SVM tracker, which we proposed here, can choose the SVM classifiers with the best discriminative ability to the chosen samples, and on-line combines them together by adjusting their weight. Using this method, the ensemble classifier can use the history information more reasonable, at the same time the final tracker has an especially strong distinguish ability, which makes the tracking result more reliable.

4 Experiments

In this section, several experimental results are carried out by our algorithm. Region patterns we used here are some common features: histograms of oriented gradients (HOG) [14] and local binary patterns (LBP) [5]. Integral histograms [15] are built for extracting region feature efficiently. Similar to the method used in [14], we construct a 9 bins HOG histogram for each cell, each block contains four cells with a 36-D HOG feature vector that is normalized to an L2 unit and a 59-D feature vector for LBP. Different form [14], the pixels including in a cell is not a constant, because the object region is scalable. The positive samples are captured by scaling the target region from 0.8 to 1.5 and rotating it form -8 degree to +8 degree. The negative samples are captured within the context region, and the negatives can have some overlaps with the positive one (below 1/3 area of positive sample). The sample rate between each negative region is set as 5 pixels per step in our method. In our method, the scale problem is solved by a naive way. The suitable scale is got by searching different scales around the centre of local maximum region.

First of all, we captured some videos by ourselves to demonstrate the robustness of our framework (fig. 6(a)).Then we carried out our method on some frequently used public available sequences (fig. 6(b,c)). Compared with some other popular methods,

Fig. 6. (a) A glass bottle with illumination and appearance change while moving in cluttered background. Beside the target, there is another bottle, which looks like the target bottle. However, the tracker can discriminate them very well without confusion. (b) A moving doll with large pose and illumination change, frames 1, 454, 728, 1162 and 1343. (c) A moving vehicle with disturbance of the light around, frames 1, 129, 245, 295 and 391.

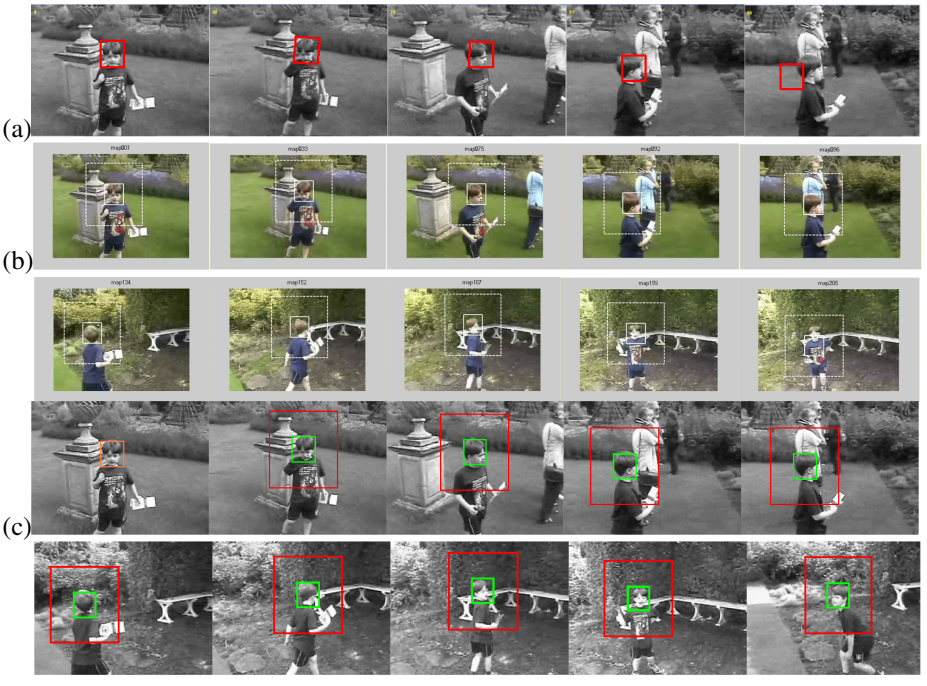


Fig. 7. Results of a boy with large head pose change. (a) Results of incremental subspace learning tracker. The tracker failed after frame 96 (b) Results of ensemble tracker, it runs well for 203 frames, but failed later, which may cause by size problem. (c) Results of our method.

the incremental subspace learning tracker [13] of Ross and Lim and the ensemble tracker [16] of Shai Avidan, we get the results in fig. 7. The incremental learning tracker is based on updating the sample mean and the eigenbasis over time. However, when the variation is very large, the updating can't adapt the change quickly enough and an imprecise position may be got. So after several frames' updating, the target may drift very quickly because of error accumulation. The ensemble is powerful, however it is a pixel based tracker (as [18]), the information for a pixel is little and the feature vector may have a large variation when the target is colorful, and the tracker may get confused when the color of the target and the background is similar. Our method, as mentioned before, is based on the region patterns which is more stable while tracking. Meanwhile, it contains and chooses the most useful "key frames" of the target by ensemble of SVMs, which have the most discriminative ability. Because of that, the performance of the tracker is especially good on some challenging videos with large appearance variation.

5 Conclusion

In this paper, we build a novel framework to track general object. The ensemble SVM tracker proposed here is made up of several SVM classifiers, which are proved especially strong in selecting and recording the "Key Frames" of the object. These classifiers are generated and updated during different periods with different historical

information. By on-line adjusting each SVM's weight, the ensemble classifier can distinguish the target and the background better than any single component. With the selected useful historical information and the strong discriminative ability, the tracker performs especially well on some difficult videos with large appearance variation.

Acknowledgments

This work was done when the author visited visual computing group in Microsoft Research Asia. The author would like to appreciate all the researchers in that group for their supports. Thanks for Lim and Ross's image sequences and matlab code of incremental subspace learning tracker provided in their website. Furthermore, thanks to Shai Avidan's help and the results of ensemble tracker provided by him.

References

1. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. PAMI 20(10), 1025–1039 (1998)
2. Black, M.J., Jepson, A.: EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision 26(1), 63–84 (1998)
3. Isard, M., Blake, A.: Condensation-Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
4. Perez, P., et al.: Color-Based Probabilistic Tracking. In: ECCV, pp. 661–675 (2002)
5. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI 24, 971–987 (2002)
6. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. PAMI 25(5), 564–577 (2003)
7. Vacchetti, L., Lepetit, V., Fua, P.: Fusing online and offline information for stable 3D tracking in real-time. In: CVPR 2003, vol. 2, pp. 241–248 (2003)
8. Nummiaroa, K., Koller-Meierb, E., Gool, L.V.: An Adaptive Color-Based Particle Filter. Image and Vision Computing 99–110 (2003)
9. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. PAMI 25(10), 1296–1311 (2003)
10. Avidan, S.: Support Vector Tracking. PAMI 26(8), 1064–1072 (2004)
11. Matthews, I., Ishikawa, T., Baker, S.: The Template Update Problem. PAMI 26, 810–815 (2004)
12. Okuma, K., Taleghani, A.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 28–39. Springer, Heidelberg (2004)
13. Ross, D., Lim, J., Yang, M.H.: Probabilistic visual tracking with incremental subspace update. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 470–482. Springer, Heidelberg (2004)
14. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR 2005, vol. 1, pp. 886–893 (2005)
15. Porikli, F.: Integral histogram: a fast way to extract histograms in Cartesian spaces. In: CVPR 2005, vol. 1, pp. 829–836 (2005)
16. Avidan, S.: Ensemble tracking. In: Proceedings of CVPR 2005. vol.2, pp. 494–501 (2005)
17. Grabner, H., Bischof, H.: On-line Boosting and Vision. In: CVPR 2006, vol. 1, pp. 260–267 (2006)
18. Wu, Y., Huang, T.S.: Color Tracking by Transductive Learning. In: Proceedings of CVPR 2000, vol. 1, pp. 133–138 (2000)