

Robust Tracking Based on Pixel-Wise Spatial Pyramid and Biased Fusion

Huchuan Lu¹, Shipeng Lu¹, and Yen-Wei Chen^{1,2}

¹ School of Information and Communication Engineering,
Dalian University of Technology, Dalian, China

² College of Information Science and Engineering,
Ritsumeikan University, Kusatsu, Japan

Abstract. We propose a novel tracking algorithm for the balance between stability and adaptivity as well as a new online appearance model. Since the update error is inevitable, we present three tracking modules, i.e., reference model, soft reference model and adaptive model, and fuse them using biased multiplicative formula. These three contributors are built through the same appearance model with different update rate. The appearance model, Pixel-wise Spatial Pyramid, employs pixel feature vectors instead of SIFT vectors, to combine several pixel characteristics. In particular, the reserved pixel feature vectors are used to create a new codebook together with the earlier codebook. A hybrid feature map consisting of the reserved pixel vectors and anti-part of previous hybrid feature map is built to represent the new target map. Experimental results show that our approach tracks the object with drastic appearance change, accurately and robustly.

1 Introduction

Visual object tracking is one of the well-known problems in the computer vision community. Tracking intrinsically focuses on comparison problem: In general, tracking system can be thought of similarity metric-based algorithm or classification-based algorithm. Many similarity metric-based trackers have been proposed, such as probabilistic models using mean-shift [1, 2] or particle filtering [3], IVT [4] and FragTrack [5]. Classification-based algorithms [6, 7, 8], meaning to optimally discriminate the object from the current background, perform well on various challenging conditions. Our tracker based on pixel-wise spatial pyramid and biased multiplicative formula falls into the first category.

To deal with the significant appearance variations in the video sequences, due to the pose variation, shape deformation, scale change, illumination change, camera motion, and occlusions, tracking algorithm should be adaptive through the online update. The most of previous online tracking algorithms using a self-learning policy, i.e., the tracker relies on its own predictions, unfortunately faces a severe drifting problem. This trouble can be explained by the stability-plasticity dilemma [9]: If the tracker is built only with the initial information, it is the least error-prone to drift but can not survive undergoing appearance and viewpoint

changes. On the contrary, the self-learning online tracker is highly adaptive but easily drifts. Some methods have been proposed to find the trade-off between adaptivity and stability.

Grabner *et al.* [10] developed tracking as a semi-supervised learning problem using online boosting. It has shown to be less susceptible to drifting while adaptive, but it keeps the non-optimal prior. Recently, they [11] have advanced the tracker by extending the semi-supervised learning approach with adaptive priors, making it robust to track multiple similar objects.

Babenko *et al.* [8] successfully used online multiple instance learning to overcome the ambiguities of bounding boxes during tracking, and got the state-of-the-art results. Santner *et al.* [12] tackled this problem by combining several complimentary trackers operating at different timescales.

We also address the robustness and adaptivity of online appearance-based tracking regarding the reliability or credit of the tracking model in this paper. The underlying assumption is that the update can not be absolutely correct and drifting risk always exists, because the supposed objects used to update are more or less wrong, with ambiguity or label jitter. For the initial appearance model, it is the most reliable one during the tracking period, then different update rate of the model means different confidence, i.e., the more modified the model is, the less trustworthy it is. Specially, we make use of the initial (stable) appearance model, a soft stable appearance model and a novel adaptive appearance model, eventually fuse them using biased multiplicative principle (Fig. 1). Note that the appearance models are built using the same method, only with different adaptivity rates.

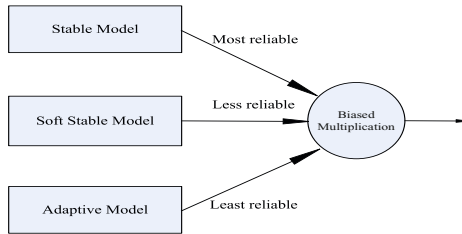


Fig. 1. Fusion of the three models

There is no doubt that effectively modeling appearance variations plays a critical role in visual tracking. Many researchers [1, 2, 3, 4, 13, 14] focus on the design of appearance model to strengthen the discriminability. Porikli *et al.* [13] proposed a covariance matrix descriptor for characterizing the appearance of an object to capture both statistical and spatial properties of object appearance. In particular, the covariance matrix descriptor offers a principal way to fuse several features through pixel feature vector style. Meanwhile, Arif *et al.* [14] employed the individual pixel feature vectors as observation in the KPCA eigenspace to create a pixel-wise appearance model which is robust to noise and occlusions, whereas previous approaches used vectorized image regions as observation.

Bag of words (bag of features) [15,16] representations have become popular for content based image classification and object localization owing to their simplicity and good performance. The main idea is to treat images as loose collections of independent local features, using the cluster label distribution in feature space as a characterization of the image. However, because these methods disregard all information about the spatial layout of the features, they have markedly limited descriptive ability. Lazebnik *et al.* [17] developed “spatial pyramid”, a simple and computationally efficient extension of the orderless bag of features image representation, and gained significantly improved performance on challenging scene categorization tasks.

The second contribution of this paper is to present a novel online learning tracker with a new appearance model and an update scheme designed for the model. We build “spatial pyramid” using pixel-wise feature vectors in the region of interest. Pixel-wise feature vector consists of several individual pixel-features. During the process of update, a codebook built by K-means is carefully modified through the distance-based scheme. We generate a hybrid feature map, i.e., the new valuable information in the current frame and the cumulative information from previous images, to absorb the essence and reject the dross as much as possible.

We briefly depict an overview of our method in section 2, describe the online learning approach with pixel-wise spatial pyramid for visual tracking in section 3, give a detailed analysis of model fusion rule in section 4, and discuss the experimental evaluation in section 5, followed by conclusions and future work in the last section.

2 Overview of the Method

This section gives an overview of our tracking system, which is summarized in Fig. 2. Our goal is to make the tracking algorithm to be adaptive to drastic appearance changes and recoverable from drifting. Therefore, we elaborately develop a discriminative appearance model and considerate update approach, then make the tracker more robust and stable using the biased multiplicative principle. The functional result is to find the biased balance point among the three tracking components with dissimilar update.

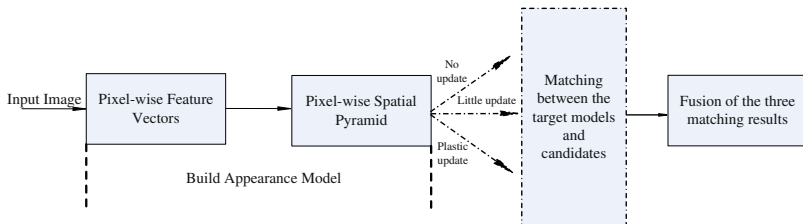


Fig. 2. An overview of our tracking algorithm

Pixel-wise feature vector, combining manifold traits instead of using one kind of feature, could be made more discriminative. We use these pixel-wise features instead of local features (e.g., SIFT [18] or HOG [19]) in the building of “spatial pyramid” to employ multiple features. The tracker’s performance proves that it is feasible and effective.

We introduce the model update rate α , where α denotes the level of the update of the appearance representation, i.e., the percentage of the reserved current pixel feature vectors unlike the adaptivity rate in [12]. The adaptivity rate in [12] denotes the number of frames a tracker needs to fully adapt to appearance changes. (i) The most reliable reference information without update does not suit appearance changes with $\alpha = 0$. (ii) The soft stable appearance model with mid-update, to some extent, copes with the appearance variations with $0.1 < \alpha < 0.4$. (iii) Frame-to-frame online tracker with a moderate ratio of update fits the appearance changes as well as possible from the premise of eliminating the noise information, with $0.4 < \alpha < 0.9$. In this paper, the soft stable model with $\alpha = 0.15$ is updated every three frames. The online model is updated each frame, with α determined by an empirical threshold.

3 Our Online Learning Tracker

3.1 Sequential Inference Model

Our on-line tracker meets the Bayesian Inference for visual tracking, which is a Markov model with hidden state variables. Using Bayes’ theorem, the tracking equation can be written as follows:

$$p(X_t/D_t) \propto p(I_t/X_t) \int p(X_t/X_{t-1})p(X_{t-1}/D_{t-1})dX_{t-1} \quad (1)$$

To benefit the building of appearance model, an affine motion sampling model as in [4] is used to attain the candidates. Hidden state variables X_t denote the affine motion parameters by six parameters $X_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}$, and $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$ denote x, y translation, rotation angle, scale, aspect ratio, and skew direction at time t , I_t describes the observed image at t frame, and $D_t = \{I_1, I_2, \dots, I_t\}$ contains observed image at the t frame and those before t frame. The term $p(X_t/X_{t-1})$ is called dynamical model, and usually modeled by a Gaussian distribution in which each parameter of X_t is treat independent. And $p(I_t/X_t)$ is called observation model, which is a probability to describe the target tracked.

As the integration in Eq. (1) is intractable analytically due to the non-Gaussian form of $p(I_t/X_t)$, we resort to particle filtering-based sampling. The particle is represented by the pixel-wise spatial pyramid.

3.2 Pixel-Wise Spatial Pyramid

The image I is represented as a two-dimensional lattice of a one-dimensional intensity image or a three-dimensional color image. Let $F(x, y)$ be the d -dimensional appearance vector extracted from I at the spatial location (x, y)

$$F(x, y) = \Gamma(I, x, y) \quad (2)$$

where Γ can be any mapping such as color, intensity, image gradient I_x, I_{xx}, \dots , edge, texture etc. The original pixel feature vector in [13] includes spatial attributes that are obtained from pixel coordinate values, but we only use the d -dimensional appearance vector and the spatial layout is exploited through the spatial pyramid. So a $M \times N$ rectangular region R forms a two-dimensional matrix of pixel feature vectors W ,

$$W_{(M*N) \times d} = [F_1, F_2, \dots, F_{M*N}] \quad (3)$$

Here, we employ the intensity and texture information to generate the five-dimensional individual pixel vector $F(x, y)$,

$$F(x, y) = [I(x, y), |I_x(x, y)|, |I_y(x, y)|, |I_{xx}(x, y)|, |I_{yy}(x, y)|] \quad (4)$$

In order to introduce spatial information, we follow the scheme proposed by Lazebnik *et al.* which is based on pyramid matching [20]. Spatial pyramid matching is a simple yet effective approach to compare similarity between images. The image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction. Let X and Y be two sets of vectors in a d -dimensional feature space which are obtained from two images. In general, SIFT vectors are used, but here they are pixel feature vectors. Pyramid matching is implemented by taking a weighted sum of the number of matches that occur at each level of resolution. Supposing we have constructed a sequence of resolutions $0, 1, \dots, L$, then we have 2^l sub-regions for the l th resolution. Let H_X^l and H_Y^l denote the histograms of X and Y at resolution l , so the histogram intersection can be computed as $I(H_X^l, H_Y^l) = \sum_{i=1}^{2^l} \min(H_X^l(i), H_Y^l(i))$. Then the overall similarity between X and Y is defined as

$$S(H_X, H_Y) = \sum_{l=0}^L w_l I(H_X^l, H_Y^l) \quad (5)$$

where the weight is $w_l = \max(\frac{1}{2^l}, \frac{1}{2^{L-l+1}})$. The details can be found in [19]. In our case, the set of pixel feature vectors are quantized by K-means with a codebook size 25, and the number of levels is limited to $L = 2$ to prevent over fitting.

3.3 Update Scheme

Though the trackers such as [1, 5] without update perform well under some circumstances, the update of tracking model is essential to cope with appearance changes. The template tracking methods is often updated by the approach based on matching score. For example, the template update mechanism in [2] is defined as

$$q^{i+1} = \alpha \pi q^i + (1 - \alpha)(1 - \pi)p(y_i) \quad (6)$$

where $\alpha = 0.85$ is a weighting factor to control the speed of the updates, q^i is the template at frame i , and $\pi = \rho[p(y_i), q]$ is the Bhattacharyya coefficient between the current template and the optimal candidate found in the i^{th} frame. The rule indicates that the update of the template will become minimal, if the template and the optimal candidate are well-matched. However, this method is not suitable in our tracker, because both the codebook and the histogram representation need to be updated. On the other hand, since there is no weak learner in our tracker, our appearance model also cannot evolve like the methods in [6, 7].

We develop a distance-based scheme to update the codebook and hybrid feature map to obtain the target appearance model, as depicted in Fig. 3.

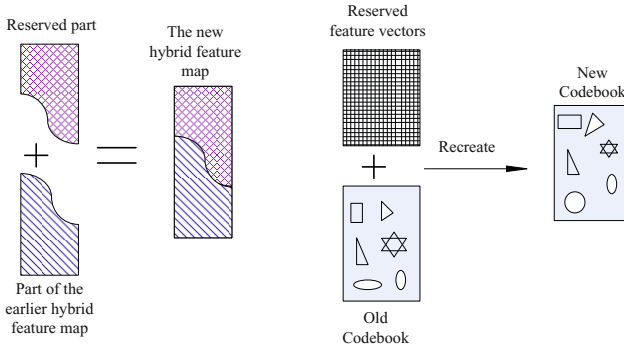


Fig. 3. Update Scheme

Each pixel feature vector in the current frame has a minimal distance to the code words. Note that the new pixel feature vectors those are nearer to the previous code words in the current frame are more likely to be the target elements. All the current pixel feature vectors are sorted in a queue ascendingly according to the minimal distances. During the process of update, for the adaptive module, an empirical threshold is used to keep the valuable information and remove the noise, particularly part of the occlusion sector. The top 15 percent in the queue is captured as the reserved feature vectors for the soft stable module. The current valuable information, i.e., part of the pixel feature vectors, together with the previous cluster centers (code words) are employed to generate a new codebook.

Since the spatial pyramid matching focuses on the matching between images, we propose a hybrid feature map according to the spatial layout to keep the cumulated instrumental information. The pixel feature vectors currently reserved are part of the hybrid feature map, and the remainders are the opposite spatial part in the earlier hybrid feature map. Figure 4 shows some of the hybrid maps in Girl and Shaking sequences (f denotes the frame number). The left shows the object region, the center depicts the hybrid maps of the adaptive module, and the right displays the hybrid maps of the soft stable module. It can be found that the hybrid maps of the adaptive module catch more appearance changes than the maps of soft stable module do, because of the different update rates of them.

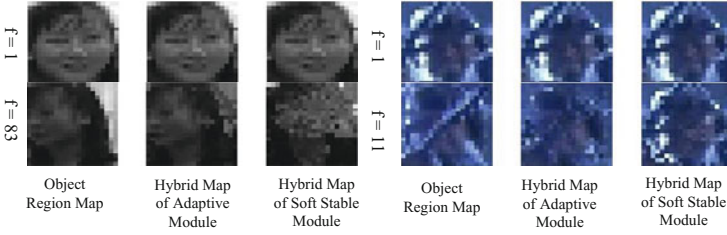


Fig. 4. Hybrid Maps of Girl and Shaking

4 Biased Multiplicative Formula

In order to take advantage of the reference information, the soft reference model, as well as online updated appearance model at the same time, we use biased multiplicative formula to fuse them. Suppose that the similarity metric matching scores are S_r, S_{sr}, S_o , i.e., likelihood scores between the candidates and the above three models. The fusion equation is defined as

$$S_f = S_r * S_{sr} * S_o \quad (7)$$

Firstly, this can be interpreted that the (soft) reference model is used to verify the judgment of the online appearance model. Furthermore, this fusion scheme can find the balance key between the three tracking modules. Finally, to make the tracker less prone to drift, we choose the candidate which has a bigger S_r or $S_r * S_{sr}$.

5 Implementation and Experiments

During the experiments, we compare our algorithm to current state-of-the-art methods, i.e., IVT, FragTrack and MILTrack, on publicly available datasets. Babenko *et al.* showed superior results comparing their method (MILTrack) to On-line Boosting and FragTrack. IVT is a successful online learning algorithm using incremental subspace representation. FragTrack benefitting from the division-combination patches scheme is robust to occlusion and perform well on several challenging sequences.

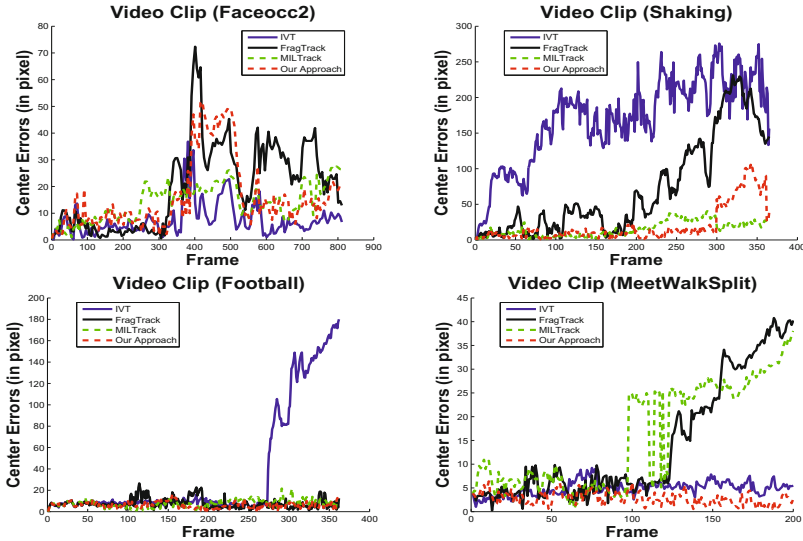
Throughout the experiments, we use seven challenging video sequences regarding e.g. moving cameras, occlusions, background clutters, 3-D motion and illumination changes. The ground truth for sequences: Girl and Faceocc2 are from [8]. ShopAssistant2cor and MeetWalkSplit come from the CAVIAR database. Shaking, Football and Skating1(low frame rate) are from [21].

5.1 Quantitative Evaluation

In this experiment, we would like to benchmark our method on the following sequences: Faceocc2, Shaking, Football and MeetWalkSplit. Table 1 and Fig. 5 depict the results based on the mean pixel error: our method yields the best

Table 1. Average center location errors in pixels

Sequence	IVT	FragTrack	MILTrack	Our approach
Faceocc2	7.5	19.9	14.3	16.1
Shaking	170.4	70.6	15.1	19.8
Football	37.3	7.5	7.5	6.1
MeetWalkSplit	4.9	14.2	16.0	2.8

**Fig. 5.** Error curves of some testing video sequences

scores in two sequences: MeetWalkSplit and Football, and gains the second best results in the sequence: Shaking. IVT performs best in Faceocc2 seq., but fails in Shaking and Football sequences, due to the severe illumination variation, out of plane rotation or viewpoint changes. FragTrack fails in Shaking seq. because of the drastic illumination changes. Though our tracker locates the object accurately in the first part of Shaking seq., it drifts in the tail because there is a combination of pose change and drastic illumination change. Our tracker even loses the target in the middle of Faceocc2 seq. because of the severe occlusion, but then recovers due to the utilization of stable and soft stable modules.

5.2 Performance of the Individual Tracking Module

Here we investigate the behavior of our three appearance modules respectively on two sequences, Girl and Shaking. The average pixel error is given in Fig. 6. The reference module works well when the appearance of the object is close to it. The soft reference module and adaptive module seem to perform poor in

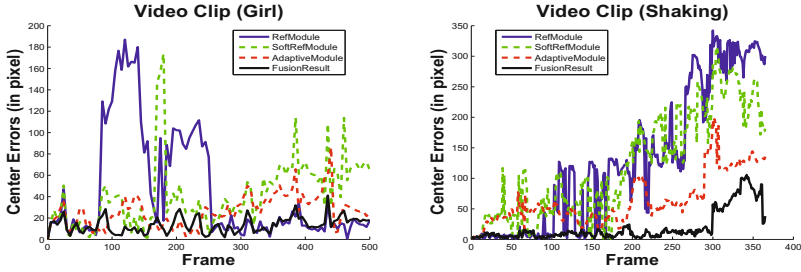


Fig. 6. Evaluation of the separate modules and fusion tracker

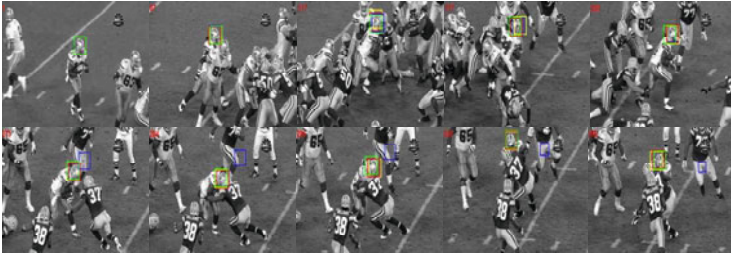


Fig. 7. Comparison between IVT, FragTrack, MILTrack and our method in Football

the sequences. But through the biased multiplicative formula, the fusion result becomes much more accurate and robust. Unfortunately, in the later part of sequence Shaking, all the results of three modules are far from the ground truth, leading drifting of the final output.

5.3 Qualitative Evaluation

We evaluate the performance of our tracking method through comparing with IVT, FragTrack and MILTrack. The bounding boxes for target of IVT, FragTrack, MILTrack and our approach are blue, yellow, green and red respectively.

Background clutter. In Fig. 7, we test Football seq. that includes severe background clutter, of which appearance is similar to that of the target. In the case of IVT, the bounding box drifts when two players collided with each other and cannot recover as illustrated in the second row of Fig. 7. Our method, FragTrack and MILTrack overcome this problem, and our approach is more accurate than them.

Occlusion. Figure 8 shows the tracking results for the pedestrian in ShopAssistant2cor. While the person of ShopAssistant2cor is severely occluded in the frames from 185 to 220, all the methods successfully track the target. But after that, the three other trackers only locate part of the object. Note that there



Fig. 8. Comparison between IVT, FragTrack, MILTrack and our method in ShopAssistant2cor

are also persons with similar color distribution comparing to the object. Our approach never drifts throughout the sequence.

3-D motion and moving camera. We present the tracking results of Girl in Fig. 9. Since our method and IVT employ the affine motion model, so we can find more accurate location of the object. Note that, in order to evaluate the motion model, the initial bounding boxes in IVT and our approach are made to be a little smaller than the boxes in MILTrack and FragTrack. The results demonstrate that IVT can track the girl in the first few frames, but fails after the girl rotates. Our method, FragTrack and MILTrack can locate the girl, but FragTrack fails during the frames from 20 to 60, MILTrack drifts in the end.

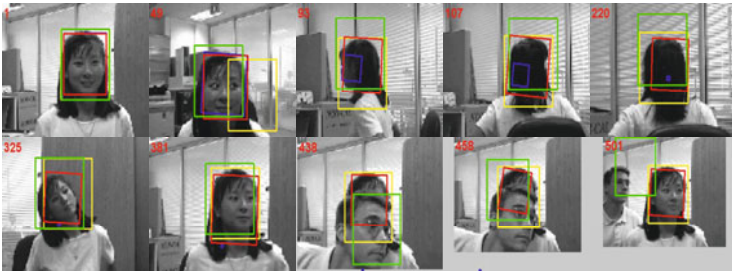


Fig. 9. Comparison between IVT, FragTrack, MILTrack and our method in Girl

Illumination change and pose variations. We assess the capability of the tracking methods regarding illumination change and drastic pose variations in Skating1(low frame rate). As shown in Fig. 10, our method covers these challenges, while IVT drifts after a few frames, the other two methods locate part of object during some of the frames. Note that there are also abrupt motion and occlusion in this sequence. For example, abrupt motion and serious occlusion can be found in frame 75 and 141 respectively.

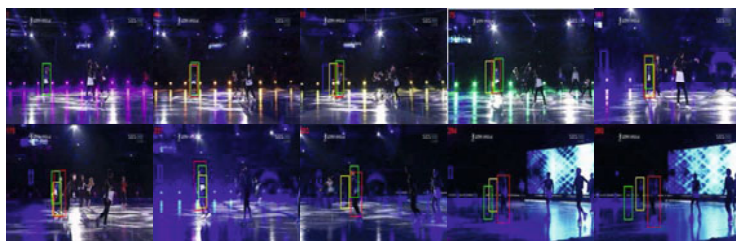


Fig. 10. Comparison between IVT, FragTrack, MILTrack and our method in Skating1(low frame rate)

6 Conclusion and Future Work

In this paper, we present a new algorithm to track object whose appearance changes drastically. We fuse three tracking modules with different update rate through biased multiplicative formula to achieve the balance between robustness and adaptivity of the tracker. Particularly, the pixel-wise spatial pyramid including several appearance features and spatial layout and the hybrid feature map accommodating the cumulated valuable pixel feature vectors play the crucial role during tracking process. We demonstrate comparative performance with the state-of-the-art tracking methods in sequences of challenging circumstances.

Future work: Through the experiments, we find that the K-means clustering is not strong enough to quantize the pixel feature vectors, because the dimension of the vector is high or the different information locates in different feature space. Other quantification methods, e.g. Gaussian Mixture Model (GMM) or Histogram Intersection Kernel (HIK), could be used to create a better codebook.

Acknowledgement. The work was supported by the Fundamental Research Funds for the Central Universities, No. DUT10JS05, and the National Natural Science Foundation of China (NSFC), No.61071209.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR, vol. 2, pp. 142–149 (2000)
2. Cannons, K., Wildes, R.: Spatiotemporal oriented energy features for visual tracking. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 532–543. Springer, Heidelberg (2007)
3. Wang, H., Suter, D., Schindler, K.: Effective appearance model and similarity measure for particle filtering and visual tracking. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 606–618. Springer, Heidelberg (2006)
4. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. *Int. J. Comput. Vision* 77(1), 125–141 (2008)
5. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR, pp. 798–805 (2006)

6. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: BMVC, vol. 1, pp. 47–56 (2006)
7. Avidan, S.: Ensemble tracking. In: CVPR, vol. 2, pp. 494–501 (2005)
8. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR, pp. 983–990 (2009)
9. Grossberg, S.: Competitive learning: From interactive activation to adaptive resonance. In: NNNI, pp. 213–250 (1998)
10. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
11. Stalder, S., Grabner, H., Gool, L.V.: Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: ICCV (2009)
12. Jakob, S., Christian, L., Amir, S., Thomas, P.: Prost: Parallel robust online simple tracking. In: CVPR (2010)
13. Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on lie algebra. In: CVPR, pp. 728–735 (2006)
14. Arif, O., Vela, P.: Non-rigid object localization and segmentation using eigenspace representation. In: ICCV (2009)
15. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR, pp. 1–8 (2008)
16. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 490–503. Springer, Heidelberg (2006)
17. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
18. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60(2), 91–110 (2004)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
20. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV, vol. 2, pp. 1458–1465 (2005)
21. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)