Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 828907, 10 pages
http://dx.doi.org/10.1155/2014/828907

Research Article

Visual Tracking Using Max-Average Pooling and Weight-Selection Strategy

Suguo Zhu and Junping Du

Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Junping Du; junpingdu@126.com

Received 23 April 2014; Accepted 7 July 2014; Published 20 July 2014

Academic Editor: Yantao Shen

Many modern visual tracking algorithms incorporate spatial pooling, max pooling, or average pooling, which is to achieve invariance to feature transformations and better robustness to occlusion, illumination change, and position variation. In this paper, max-average pooling method and Weight-selection strategy are proposed with a hybrid framework, which is combined with sparse representation and particle filter, to exploit the spatial information of an object and make good compromises to ensure the correctness of the results in this framework. Challenges can be well considered by the proposed algorithm. Experimental results demonstrate the effectiveness and robustness of the proposed algorithm compared with the state-of-the-art methods on challenging sequences.

1. Introduction

Visual tracking has a wide range of applications in computer vision, such as space visual surveillance, driver assistance system, and visual navigation. The challenges in designing a robust visual tracking algorithm are caused by the presence of occlusion, background clutter, and illumination change.

Recently, many visual tracking algorithms have been developed to tackle these challenges using sparsity of the image. They usually combine other sophisticated methods to track the target object. Examples are sparse representation combined with learning [1–3], mean shift [4–6], Bayesian estimation [7, 8], and particle filter [9]. However, they often ignore the pooling method and just take the max pooling or average pooling method without analyzing it; for example, Ross et al. [3] do not refer to the pooling method and employ the max pooling directly. Actually, the pooling method, as an indispensable step in sparse representation, does have an important influence on the performance of the algorithm. For instance, the pooling type is shown to matter more than the careful unsupervised pretraining of features for classification problems with little training data, and good results with random features are obtained when appropriate pooling is used [10]. Yang et al. [11] report much better classification performance on several objects or scene classification benchmarks when using the maximum value of a feature rather than its average to summarize its activity over a region of interest.

However, in spite of the successes of the previous work, there still exist many limitations. As discussed by Boureau et al. [12], the conditions max pooling and average pooling work in are different; for the former, it is the soft coding methods upon local descriptors, and, for the other, it is a hard quantization method. Boureau et al. [13] discuss in the area of visual recognition the differences of max pooling and average pooling and explain the link between the sample cardinality in a spatial pool and also provide results that max pooling method is better than the average pooling method through experiments except the resolution in visual tracking. In this paper, we propose using max pooling and average pooling together in sparse representation for the first stage of the algorithm, called max-average pooling method. We will discuss this method in Section 2.

Another popular tracking method is the sequential Monte Carlo methods, also known as particle filters, which recursively estimate target posterior with discrete sample-weight pairs in a dynamic Bayesian framework. The basic idea was introduced by Sarkar [14] and developed into various improved versions over the last decade. Particle filter performs well in visual areas. It is usually merged with other algorithms to overcome the complicated situation, such as the color visual spectrum and thermal spectrum images are fused in a joint sparse representation, which constructs the particle filter [9]. Unfortunately, by the limitation of inferred, drift will not be avoided when the similar obstacle appears. Ghosh and Manjunath [15] considers the process of learning the representation of each particle as an exclusive task to increase the robustness of the tracker and obtains good results; however, the speed of the algorithm is decreased greatly, and the particle degeneration still exists.

In our studies, we found that there are some similarities between sparse representation and particle filter: (1) the assumption is applied that the current tracking is based on correct result from the last frame, which also means that the result from every tracking is correct; (2) before the analysis of the current frame, they both sample particles; (3) they compute the weights for the particles to choose the best one. We employ the similarities to propose a novel framework for robust object tracking, which merges the same steps of sparse representation and particle filter. The proposed framework samples particles around the result tracked from the last frame, avoiding the particle degeneration. The two algorithms are illustrated in Figure 1(a) and the proposed framework is shown in Figure 1(b), where the W-S strategy equals Weight-Selection (W-S) strategy; it is proposed to balance the tracking to be more robust and correct. The W-S strategy will be hashed out in Section 4.

In this paper, we introduce a robust tracking method using a hybrid framework combined with sparse coding and particle filter, for which we propose max-average pooling to improve the traditional pooling method, and the Weight-Selection strategy to avoid drift during object tracking. The rest of the paper is organized as follows. Section 2 introduces max-average pooling method; Section 3 describes the particle filter; details on the framework of the proposed algorithm and analysis of the proposed Weight-Selection strategy are discussed in Section 4. Subsequently, the experiments are explained in Section 5. In Section 6, the summary of the paper is presented.

2. Sparse Representation with Max-Average Pooling

A typical assumption underlying sparse representation is that the tracking result in the last frame is enough accurate that the tracker could utilize it for the current prediction. Based on this assumption, we draw particles around the last result, the center point of the bounding box from the last frame, and make the dispersion of particles conforms to normal distribution. In this case, the particles will disperse around the result of the last frame.

The local patches within the target region can be represented as the linear combination of a few basic elements of the dictionary with the sparsity assumption. For the data in the current image $x_i \in X$, the set of the basis vector can be obtained by the following optimal formula:

$$\begin{aligned} \min_{W,D} \quad & \sum_{i=1}^{N} \left\| x_i - w_i D \right\|^2 + \lambda \left| w_i \right|, \\ \text{subject to} \quad & \left\| d_k \right\| \leq 1, \quad \forall k = 1, 2, \ldots, K, \end{aligned} \tag{1}$$

where a unit L2-norm constraint on w_k is typically applied to avoid trivial solutions, D denotes the dictionary and $w_i = \{w_{i1}, w_{i2}, \ldots, w_{iM}\}$, which contains many zeros to indicate the sparsity of the image, denotes the importance of the ith column in the dictionary D. Normally, the dictionary D is an over-complete basis set; that is, $K > N$. The parameter $\lambda > 0$ is a scalar regularization parameter that balances the tradeoff between reconstruction error and sparsity. The reconstruction error indicates the reliability of the representation. Note that there are two unknown parameters W and D. The purpose of this section is to obtain the weight value W, and the dictionary should be learned first.

In many tracking methods, the earlier tracking results are stored longer than the newly acquired results since they are assumed to be more accurate [8, 16, 17]. Accordingly, we take the objects from the first ten tracking results as the initial dictionary to solve the optimization problem. Firstly, with the help of the KNN algorithm, which is also called k nearest neighbor algorithm, we track the first ten frames to obtain the objects correctly and quickly. For the object tracked from the frames, we make affine transformations and save them into the dictionary. Finally, the dictionary is constituted as (width $*$ height) $*$ 10.

The average pooling scheme for histogram generation used by He et al. [18] is efficient, yet the strategy may miss the spatial information of each patch. For example, if we change the location of the left part and the right part of a human face image, the average pooling scheme neglects the exchange. While it is proven by Carneiro and Nascimento [19] that compared with average pooling the method of max pooling is more accurate, meanwhile, max pooling will be more suitable for the sufficient sparse conditions [13]. Combining the advantages of the two pooling methods, we propose max-average pooling method to solve the above problems:

$$w_i = \frac{\max\left\{w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{iN}\right\}}{\sum w_{ij}}, \quad j = 1, 2, \ldots, N, \tag{2}$$

where w_i denotes the final ith pooling result, which is produced by max-average pooling method. The proposed pooling method has two steps: maximizing of the weights in vector w_i and then averaging it. The denominator in formula (2) indicates the first step of calculating the maximization of the weights, and the numerator is the sum of $\{w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{iN}\}$, which indicates the averaging step. Utilizing the proposed max-average pooling method, we make reconstruction and calculate errors of the reconstruction with patches, the formula used for that is as follows:

$$\text{error} = \sum \left\| x_i - w_i D \right\|^2, \tag{3}$$

where error is a vector of 1 with N and N is the number of the particles. Finally, we choose the minimum one as the goal

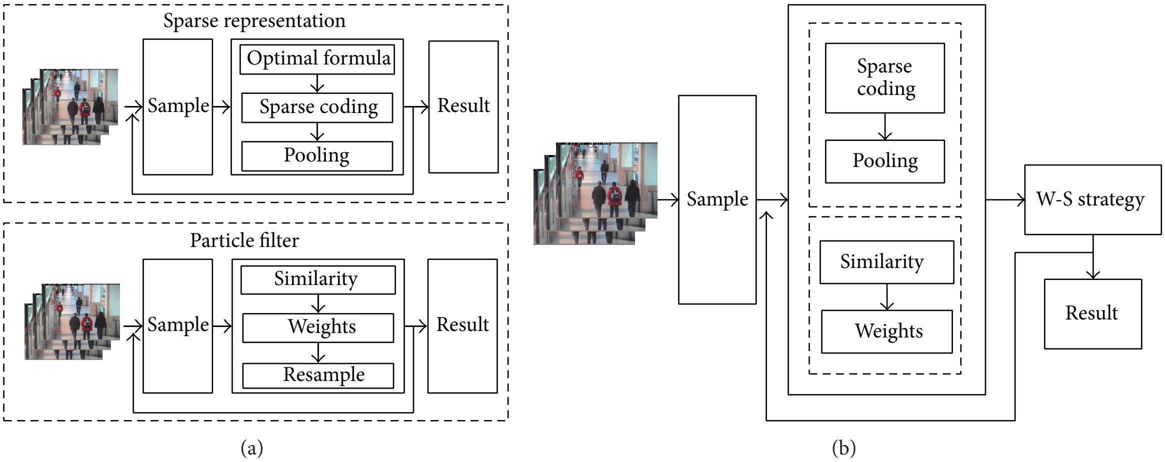


FIGURE 1: The illustrations for the algorithmic procedures and the proposed framework: (a) algorithm procedures of sparse representation and particle filter and (b) the proposed framework.

of our algorithm, which corresponds to the tracking result of the current frame. The algorithm process for one frame is as shown in the Algorithm 1.

3. Particle Filter

The particle filter is a Bayesian sequential importance sampling technique for estimating the posterior distribution of state variables characterizing a dynamic system [14].

Let X_t denote the state variable of the object at time t. The procedure of particle filter is iterative, where the initial step is to sample N particles from a set of previous particles X_{t-1} proportionally to their likelihood $\{w_{t-1}^i\}$ according to the distribution $p(x_t \mid x_{t-1}, z_{1:t})$. Subsequently, a new state X_t is generated. And that is the traditional sampling means. However, the problem is that the new particles are always affected by the likelihood $\{w_{t-1}^i\}$, which is obtained from the last state $t-1$. That situation is called particle degeneracy which causes most of the particles concentrate the positions with bigger weights. This condition is easy to cause drift. In this paper, the distribution of particles is shared with sparse representation, avoiding the weight change caused by the particle degeneracy. The reason is that there is a similar assumption between sparse representation and particle filters, which is that the tracking results before state t are all sufficiently exact that the subsequent states utilize their results for prediction.

To accomplish the tracking, we compute and update the importance weight of each particle. The posterior $p(x_t \mid z_{1:t-1})$ is approximated by the finite set of N samples with importance weights w_t^i. As is well known, if the particle x_t^i is the one the tracker predicts, the background information in its bounding box will be less than the others and the weight is bigger. Accordingly, we take two steps to calculate the weight w_t^i of each particle. An extra bounding box is settled in the original one, which is shown in Figure 2. The weights of the particles are updated separately as

$$w_{t,j}^i = w_{t-1}^i \frac{p_j\left(z_t \mid x_t^i\right) p_j\left(x_t^i \mid x_{t-1}^i\right)}{q_j\left(x_t \mid x_{1:t-1}, z_{1:t}\right)}, \qquad i = 1, 2, \ldots, N, \quad j = 1, 2, \tag{4}$$

where $x_{1:t-1}$ denotes the random samples forming posterior probability distribution, N denotes the number of the samples, $z_{1:t}$ denotes the observed values from the beginning to time t, x_t^i denotes the ith sample at time t, w_{t-1}^i denotes the ith weight vector at time $t-1$ (the last frame), $q(x_t \mid x_{1:t-1}, z_{1:t}) = p(x_t \mid x_{t-1})$ is an importance distribution, and the weights become the observation likelihood $p(z_t \mid x_t)$. Then, one weight is multiplied by the other for the final value. Consider

$$w_t^i = w_{t,1}^i \cdot w_{t,2}^i. \tag{5}$$

The above idea can be illustrated in Figure 2. The weights are calculated with two separated parts which are called bounding-1 and bounding-2. The width of bounding-2 is a quarter of bounding-1 and the height is an eighth of bounding-1. Note that bounding-1 is much bigger than bounding-2. The improvement for the calculation of weights will increase the accuracy of the prediction, especially as the position of object changes; bounding-1 will conclude the changes of object.

4. Tracking with the Proposed Algorithm

4.1. Sparse Representation Jointed with Particle Filter. Sparse representation and particle filter both have advantages and disadvantages of their own. The sparse representation method is more stable for tracking and able to consider the spatial information, while the particle filter has stronger adaptability for nonlinear and non-Gaussian distribution of continuous system than the other state-of-the-art methods. Nonetheless, for the sparse representation method, wrong reconstruction happens occasionally and it is the fatal problem which could cause drift when shape distortion occurs, while particle degeneration is also hard to solve for the particle filter.

```
Input: An image from the video sequences, the center point of the bounding box, the noise coefficient
$\lambda$, and the number of particles $N$.
(1) Sample $N$ particles around the initial point of the bounding box, and the distribution of the particles
    is the normal distribution, which is implemented by the function randn.
(2) Initialize the dictionary using tracking results of the first ten frames.
(3) Compute the formula to obtain the sparse codes: $\min_{W,\ D} \sum_{i=1}^{N} \|x_i - w_i D\|^2 + \lambda |w_i|$
    Note that $\|d_k\| \le 1$ and $k = 1, 2, \ldots, K$, where $K$ is the size of the patch' width by height.
(4) Pool the features by computing: $w_i = (\max \{w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{iN}\}) / (\sum w_{ij})$
(5) Compute the error of reconstruction:
        $error = \sum \|x_i - w_i D\|^2$
(6) Finally, obtain the particle which is corresponding to the minimum error.
```

Algorithm 1: The proposed algorithm with max-average pooling.

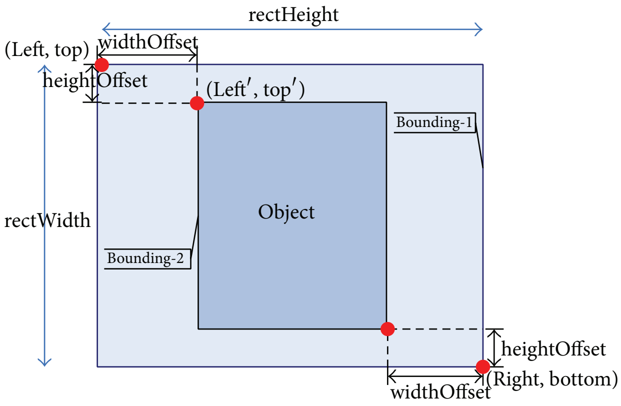


Figure 2: Two bounding boxes for weights.

In this paper, we combine the two methods for our tracking. Under the framework of particle filter, we employ the sampling method of sparse representation in the phases of initialization for each frame. We solve the optimal mathematical problem by using a popular tool called SPASM library [20] with the dictionary obtained from the first ten frames. The pooling method is used as mentioned in Section 2. Through this, we obtain one result about the object. Subsequently, we compute the weights of all particles sampled at the sparse phase using weight computation method in particle filter, and it also generates a tracking result. Actually, two tracking results are generated from the above phases, and one should be chosen to be the tracking result. The solution will be proposed in the next section.

4.2. Weight-Selection Strategy. Observing the two tracking procedures, we found that the results of sparse representation are more accurate than those of particle filter. However, during the tracking using the above method, there have been always some results jumping far away from an object; for example, Figure 3 shows this situation in the 95th frame and 96th frame: the green point denotes the result of sparse representation (SR); the yellow point denotes the result of particle filter (PF); the blue points denote the sampled particles; and the red bounding box is the tracking result. Note that the result of sparse representation suddenly “jumps” to the point away from the object when the next frame comes, while the one of particle filter just stays around the object though it is not exact enough. It is obvious that the tracking result is wrong, which means that the tracker drifts.

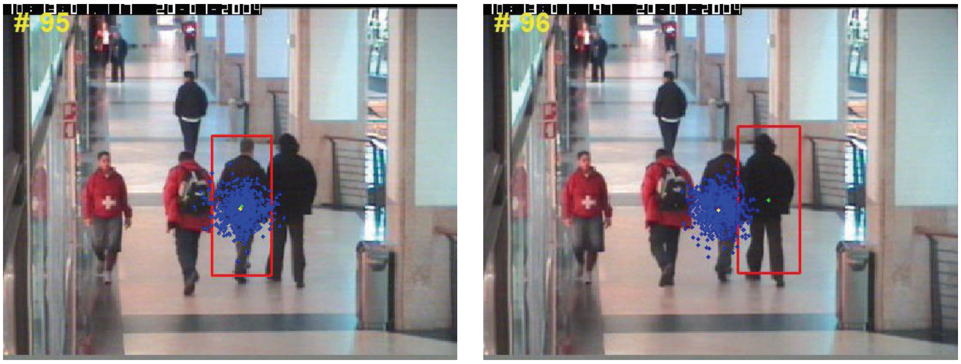
Analyzing the problem above, the proposed max-average pooling method for the sparse reconstruction sometimes is not sufficiently accurate. This problem also exists in max pooling and average pooling method. The primary reason for tracking inaccuracy, even for the drift, is the sparse reconstruction error. Figure 4(a) describes well the relationship between sparse reconstruction errors and the image sequences as occlusion comes. In image sequences, the occlusions occur from about the 63rd to the 88th frame and the 93rd to 150th frame. In this figure, the two-frame sections have the highest reconstruction errors, which indicate that when the occlusion comes, reconstruction errors will be high enough which can influence the results of object tracking. After the occlusions, the errors drop down and stay below 0.1 at almost the remaining time.

To lower the impact caused by the reconstruction errors, Zhang et al. [16] employ a regularized variant of the L1 norm to reconstruction of sparse error; Cong et al. [17] propose a criterion, SRC (sparse representation cost), to detect abnormal event; He et al. [18] discuss the impacts of reconstruction errors for signal classification. Although these methods improve the accuracy of the tracker, they increase the algorithm complexity. For the consideration of the tracking speed and the advantages of particle filter for tracking, we propose Weight-Selection (W-S) strategy to solve the above problem by choosing different results for different track conditions.

We compute the histograms in the bounding boxes which take tracking points as centers. Comparing the histograms, respectively, with the result in the last frame, we choose the one that has smaller distance. Let λ to be a weight to remedy the inaccuracy cost by the sparse representation error. If the sparse result is similar to the last result, we will make no difference; otherwise, calculate the result as

$$\text{result} = \lambda \times \text{sparse} + (1 - \lambda)\,\text{particle}, \qquad (6)$$

where sparse denotes the tracking results of sparse representation method, particle denotes results of particle filter, and



(a) The 95th frame

(b) The 96th frame

Figure 3: Tracking results in ThreePostShop2cor with the problem (green: result of SR, yellow: result of PF, blue: the sampled particles, and red: the final tracking result).

λ is set to be 0.3. Here, we do not put the particle result to replace sparse representation directly since the particle result is also not accurate at all the time. We take compromise to improve the predictive validity and achieve better data processing. When object distortion happens, the chosen weight refrains from drifting and enhances the temporary accuracy of the tracking.

Through W-S strategy, we improve the performance of our tracker. The working situation about W-S strategy is shown in Figure 4(b), which is an experiment of ThreePastShop2cor sequence. The value switches between 1 and 2 to make different decisions. If the value is 1, it means that the reconstruction error is so small that the current prediction of sparse representation is more similar to the former result and works as well as the tracking result; otherwise, the reconstruction error is incorrect and the final result is obtained by (6). Note that it also switches from 1 to 2 or from 2 to 1 even after the occlusions. That is because W-S strategy makes a positive impact on tracking even after the occlusions to decrease the effect of pose and position changing.

We demonstrate the proposed method on ThreePostShop2cor sequence. The results of the 95th and 96th frames are shown in Figure 5: the green point denotes the result of sparse representation (SR), the yellow point denotes the result of particle filter (PF), the blue points denote the sampled particles, and the red bounding box denotes the final tracking result. Note that after using the Weight-Selection strategy, we correct the jump error and improve the performance of our tracker.

4.3. Tracking Algorithm with the Proposed Method. In the following, we provide a summary of the proposed tracking algorithm.

(1) Locate the target in the first frame, either manually or by using an automated detector, and use a single particle and a bounding box to indicate this location.

(2) Initialize the dictionary with the results by tracking the target object in the first ten frames using KNN method.

(3) Advance to the next frame. Draw particles according to the dynamical model.

(4) For each particle, extract the corresponding window from the current frame, calculate its reconstruction error using max-average pooling, and choose the particle with the minimum error to be the temporary result of sparse representation method. At the same time, calculate weights of every particle and choose the best one as the result of particle filter according to the particle filter principle.

(5) Utilize Weight-Selection strategy and select the best one to be the final result.

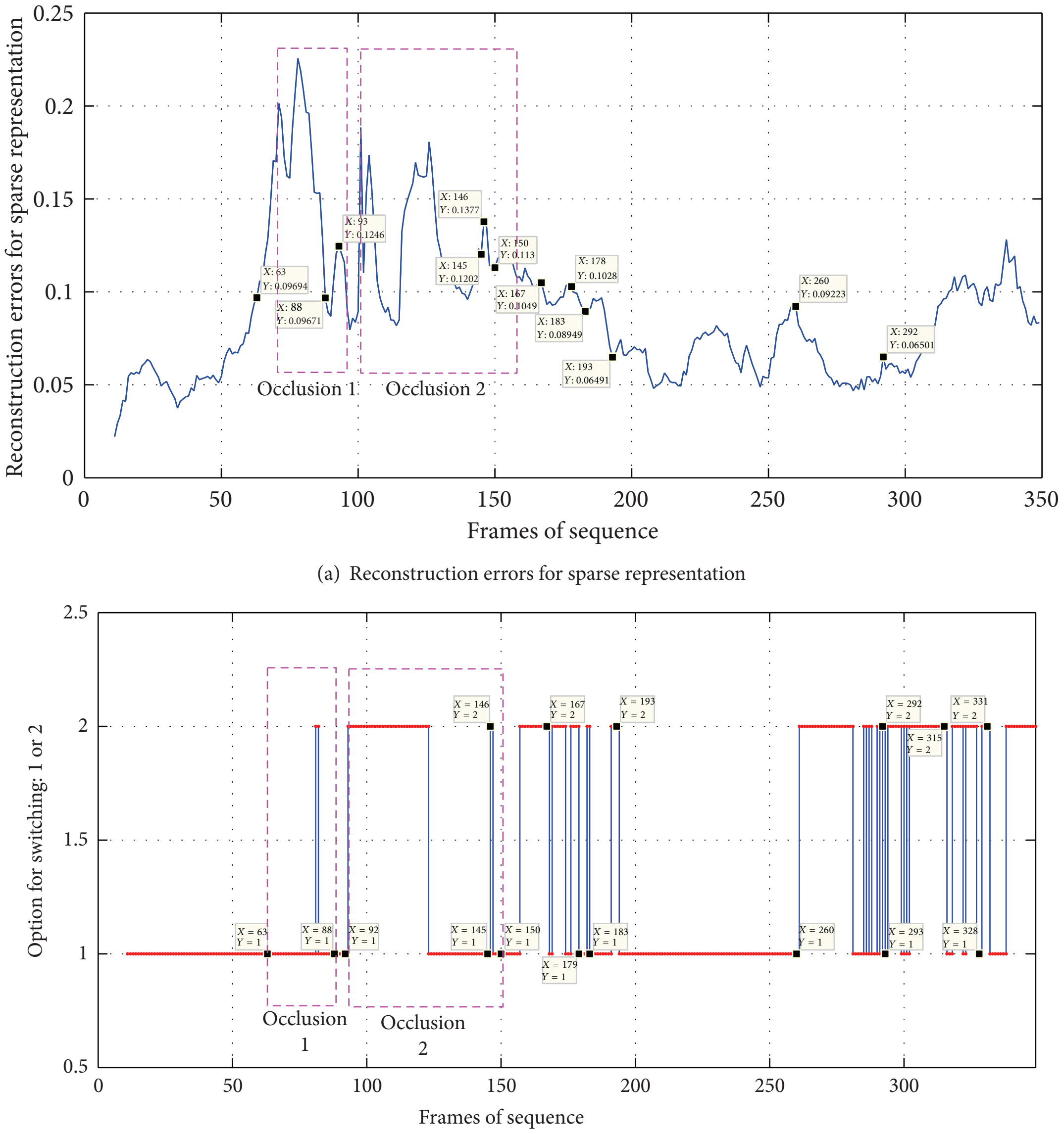
(6) Go to Step (3).

Our tracker works as a collector at the very beginning when it initializes the dictionary, which is also called templates based on tracker. Between the accuracy and the speed of the algorithm, there is actually a tradeoff. In the next section, we will discuss the implementation issues and analyze the experimental results.

5. Implementation and Experiments

The program of the proposed algorithm is implemented in Matlab r2012b and runs at about 1.5 frames per second on an Intel Core 3.2 GHz with 4 GB memory. We apply the affine transformation with six parameters to model the target motion between two consecutive frames. The sparse coding problem is solved with the SPAMS package [20] and VLFeat open source library (http://www.vlfeat.org/). For each sequence, we set 600 particles for every frame, and the location of the target object is manually labeled in the first frame. To make the scenario a bit more realistic and more challenging, we use the same parameters for all the sequence; for instance, the parameter λ used in the W-S strategy is set to be 0.3.

We evaluate the performance of the proposed algorithm on three different kinds of challenging video sequences from the previous work [8], CAVIAR data set (http://homepages.inf.ed.ac.uk/rbf/CAVIAR/) and our own.



(a) Reconstruction errors for sparse representation

(b) Weight-Selection strategy for tracking

Figure 4: Data analysis about reconstruction errors. (a) It illustrates the relationships between sparse reconstruction errors and the image sequences as occlusions come; (b) it describes the function of Weight-Selection strategy which decreases the impacts of reconstruction errors during the tracking. Value 1 or 2 represents the option for W-S strategy.

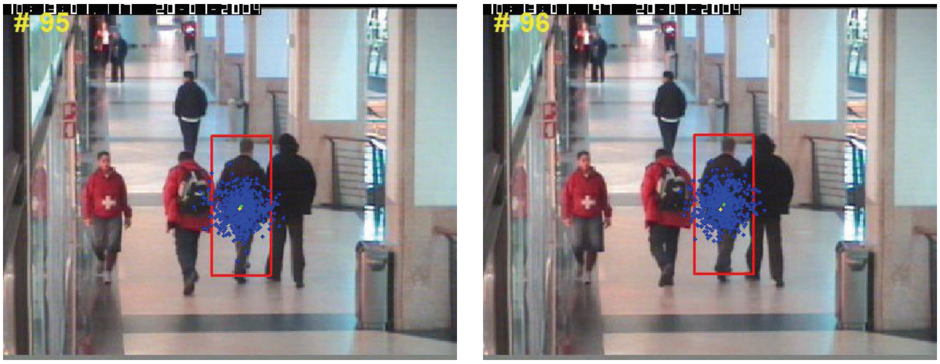
The challenges of these sequences include factors that are generally considered to be of importance by the scholars, such as occlusion, illumination change, pose/scale variation, and background clutter. The compared algorithms include two state-of-the-art tracking methods: tracking-learning-detection (TLD) tracker [2] and incremental learning for visual tracking (ILVT) tracker [3], which are provided by the authors used for fair comparison. For a better comparison, the parameters for the comparison algorithms and the proposed algorithm are set to be the same, which will ensure that the conditions are the same for all of the algorithms to track the object.

5.1. Quantitative Evaluation. Evaluation criteria are employed to quantitatively assess the performance of the trackers. Figure 6 presents the relative position errors, which is calculated (in pixels) against the manually labeled ground truth. Let

$$\text{error} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(x_{M,i}-x_{\text{gt},i}\right)^2+\left(y_{M,i}-y_{\text{gt},i}\right)^2}, \qquad (7)$$

where error denotes the relative position error of the tracking result $(x_{M,i}, y_{M,i})$ and the ground truth $(x_{\text{gt},i}, y_{\text{gt},i})$, M means the tracking algorithm M, N denotes the frame number of the tested video sequences and i indicates the ith frame. The detail of the errors is shown in Figure 6.

Overall, the proposed algorithm performs well against the state-of-the-art methods. It is able to overcome the influences of the occlusions, illumination change, and pose variation.



(a) The 95th frame

(b) The 96th frame

Figure 5: Tracking results in ThreePostShop2cor with Weight-Selection strategy (green: result of SR, yellow: result of PF, blue: the sampled particles, and red: the final tracking result).

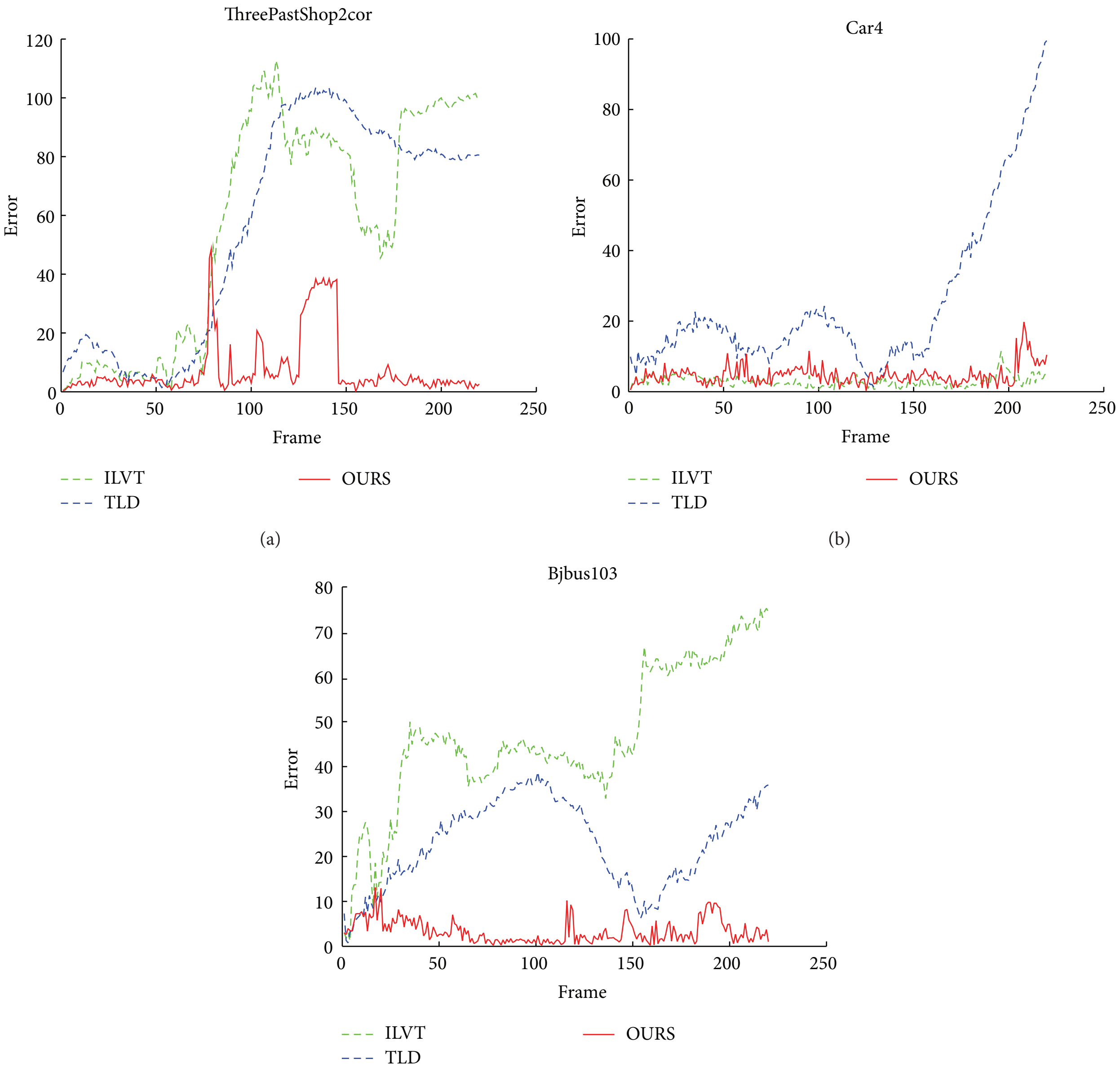
The performance of our method can be attributed to the efficient pooling method and W-S strategy.

5.2. Qualitative Evaluation. The first sequence, ThreePastShop2cor, has been used in several recent tracking papers [1, 8], and it presents the challenging issues such as occlusions and scale and pose changes. Especially for occlusions, the sequence shows a process of no occlusion, presenting occlusion and the occlusion disappeared, and the occlusions are from the 63rd to the 88th frame and the 93rd to 150th frame. The occlusion happens twice and the two persons nearby are similar to the object, especially the one on the right. In Figure 7, we use a red rectangle to denote our proposed method, blue to TLD method and green to ILVT method. Note that during the object (the person) moving from the close to the distant, the comparison methods begin to drift or deviate from the object, while the proposed method is always able to track the object; even occlusions appear twice. The TLD tracker shows inaccuracy at about the 70th frame, although its rectangle (blue) includes the object; it contains too much background information, which could lead to error for the computation of the similarity between two histograms and make the forward or backward error not exact. ILVT method occurs incorrectly as the red person blocks the target object at about the 63rd frame. The reason is that, in the 78th frame, the object disappeared temporally; at the same time, ILVT tracker could not search for it and make the new error information be the target object. For the next frames, ILVT method drifts completely. In the proposed method, max-average pooling not only considers the space construction but also chooses the best patches which could represent the characters of the object. The proposed tracker is able to track the object correctly and exactly; even the occlusions occur twice.

The second sequence, Car4, shown in Figure 6, contains a car driven in the sunlight, and, in the middle of the sequence, the shadow changes appearance of the car for a short time. The illumination change will disturb the tracking because it makes the appearance and texture of the object different. Note that before the shadow came, the trackers could always track the object, though TLD method contains more background information than ILVT method and the proposed method. As the shadow comes, TLD method is not able to maintain the earlier situation. It loses the object and the drift shows up. At the end of the sequence, TLD method is able to recapture the target after drifting into the background, but with higher tracking errors and lower success rate (Figure 8).

We notice that, during the whole tracking, ILVT method and the proposed method track the object all the time. The proposed method is not as steady as ILVT method. The reason is the influence of “jump.” As long as it jumps to the place far away from the object, the W-S strategy makes good compromise which can ensure the correctness of the method. The error of the proposed method is similar to ILVT method. If we do not consider the W-S strategy, the tracker will drift. From the overall perspective, although the proposed method is not as steady as ILVT method, it is able to track the object and its rectangle includes much less background information than ILVT method.

The last video was taken on the bus in the evening which is shown in Figure 9. We try to track the bus in front of the camera. The lighting is dark; the contrast between the target and the background is low, and it is very difficult for the general algorithm to track the target object exactly. However, the proposed method performs well in tracking the bus while the other two methods drift into the cluttered background when drastic illumination variation occurs. This can be attributed to the strategy of Weight-Selection which is able to capture the correct appearance change due to lighting changes. With the distance between the bus and the camera becoming further and further, the size of the proposed tracking box reduces. TLD method and ILVT method contain more information than the proposed method as the illumination changes drastically and the distance becomes far. As time passes by, the two compared methods lose the target object gradually. Meanwhile, the proposed method tracks the target object all the time.



(a)

(b)

(c)

FIGURE 6: Quantitative evaluation of the trackers in terms of position errors (in pixels). The center errors of the proposed algorithm (red) in comparison with ILVT (green) and TLD (blue) algorithm at three sequences.

6. Conclusions

In this paper, we propose an efficient tracking algorithm based on sparse representation and particle filter using the proposed max-average pooling and Weight-Selection strategy. The proposed method exploits both spatial and local information of the target by max-average pooling and avoids the drift resulting from sparse reconstruction errors using Weight-Selection strategy. This helps optimization of the spatial and local information. In addition, the combination of the sparse representation and particle filter avoids the particle degeneration, highlights their respective advantages, and improves the performance of the algorithm. Experimental results demonstrate the effectiveness and robustness of the proposed algorithm compared with the state-of-the-art methods on challenging sequences.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

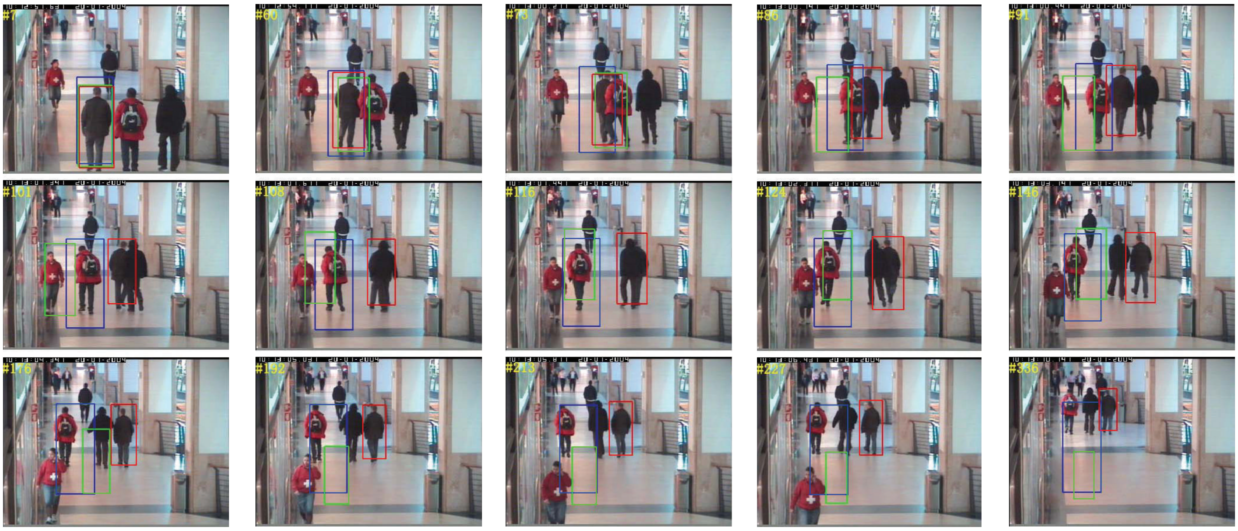


Figure 7: ThreePastShop2cor: the challenges are occlusions and pose variation (red: the proposed method; blue: TLD method; and green: ILVT method).

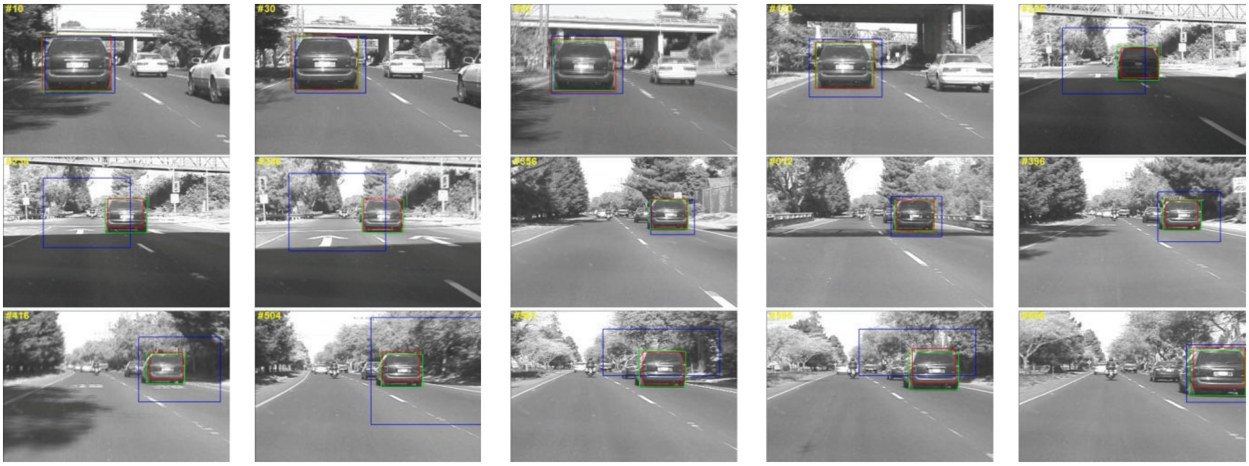


Figure 8: Car4: a car moving underneath an overpass and trees. The challenges are the illumination change and pose variation (red: the proposed method; blue: TLD method; green: ILVT method).

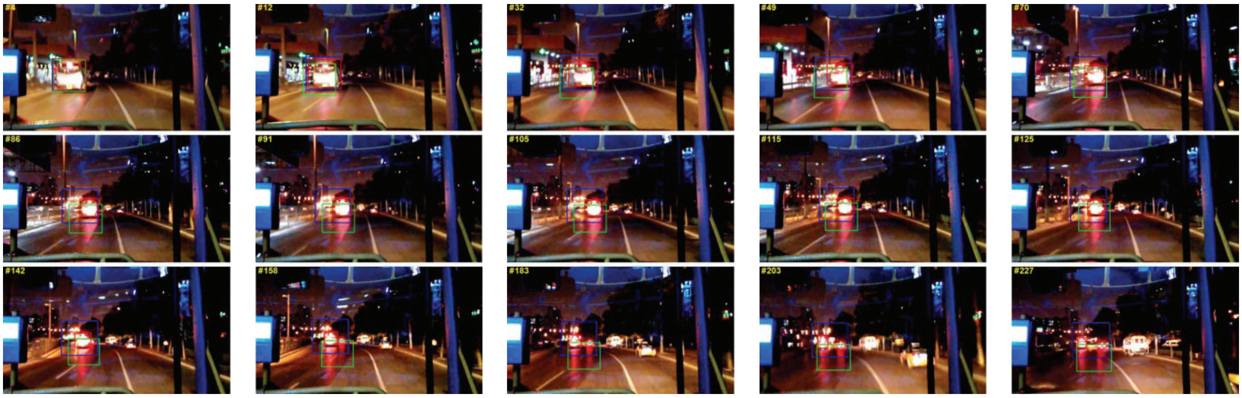


Figure 9: Bjbus103: the sequence is captured on the bus in the evening. The target object is the bus in front of the camera. The challenges are the illumination change and pose variation (red: the proposed method; blue: TLD method; green: ILVT method).

Acknowledgments

This work is supported by the National Basic Research Program of China (973Program) 2012CB821200 (Grant no. 2012CB821206) and the National Natural Science Foundation of China (Grant no. 61320106006).

References

[1] D. Wang, H. Lu, and M. Yang, "Online object tracking with sparse prototypes," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 314–325, 2013.

[2] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: tracking-learning-detection applied to faces," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 3789–3792, Hong Kong, China, September 2010.

[3] D. A. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.

[4] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[5] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1313–1320, Providence, RI, USA, June 2011.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, pp. 142–149, Hilton Head Island, SC, USA, June 2000.

[7] B. Li, W. Xiong, W. Hu et al., "Illumination estimation based on bilayer sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1423–1429, Portland, Ore, USA, 2013.

[8] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1822–1829, Providence, RI, USA, June 2012.

[9] H. P. Liu and F. C. Sun, "Fusion tracking in color and infrared images using joint sparse representation," *Science China: Information Sciences*, vol. 55, no. 3, pp. 590–599, 2012.

[10] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proceedings of IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 2146–2153, Kyoto, Japan, October 2009.

[11] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 1794–1801, Miami, Fla, USA, June 2009.

[12] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2559–2566, San Francisco, Calif, USA, June 2010.

[13] Y. Boureau, J. Ponce, and Y. Lecun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 111–118, Haifa, Israel, June 2010.

[14] P. Sarkar, "Sequential Monte Carlo methods in practice," *Technometrics*, vol. 45, no. 1, p. 106, 2003.

[15] P. Ghosh and B. S. Manjunath, "Robust simultaneous registration and segmentation with sparse error reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 425–436, 2013.

[16] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2042–2049, Providence, RI, USA, June 2012.

[17] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3449–3456, Providence, RI, USA, June 2011.

[18] S. He, Q. Yang, W. H. R. Lau et al., "Visual tracking via locality sensitive histograms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2427–2434, Portland, Ore, USA, 2013.

[19] G. Carneiro and J. C. Nascimento, "The fusion of deep learning architectures and particle filtering applied to lip tracking," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 2065–2068, Istanbul, Turkey, August 2010.

[20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.