Online Object Tracking using Sparse Prototypes by Learning Visual Prior

S.Divya, Dr.K.Latha

Abstract—**Object tracking is becoming a key ingredient in analysis of video imagery. For efficient and robust object tracking, visual prior of generic real world images are transferred for tracking the objects. The real world images are learned offline in an over-complete dictionary. The VOC2010 and CalTech101 data sets containing large variety of objects are used for learning visual prior. For visual tracking of online objects the learned visual prior is transferred for object representation using l1/l2 Sparse coding and multi-scale max pooling. With the object representation, the tracking task is formulated within the Bayesian inference framework with the use of Sparse prototypes. In order to reduce tracking drift, we present a method that takes occlusion and motion blur into account rather than simply include image observations for model update.**

***Index Terms*—Learning prior, l1/l2 sparse coding, Object Representation, Object Tracking and Sparse Prototype**

I. Introduction

Object tracking is an important task within the field of computer vision. The proliferation of high-powered computers, the availability of high quality and inexpensive video cameras, and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. There are three key steps in video analysis: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior. Therefore, the use of object tracking is pertinent in the tasks of: motion-based recognition, automated surveillance, video indexing, human-computer interaction, traffic monitoring, and vehicle navigation. Tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. Additionally, depending on the tracking domain, a tracker can also provide object-centric information, such as orientation, area, or shape of an object.

Tracking objects can be complex due to loss of information caused by projection of the 3D world on a 2D image, noise in images, complex object motion, nonrigid or articulated nature of objects, partial and full object occlusions, complex object shapes, scene illumination changes, and real-time processing requirements.

The main challenge in developing a robust tracking algorithm is to account for large appearance variations of the target object and background over time. In this paper, we tackle this problem with both prior and online visual information. Novel Tracking Algorithm is used to report the appearance variations. The central theme of our approach is to exploit generic visual prior for object tracking. Although object tracking is usually an online task and visual information of the target may be scarce before the task starts, some useful prior can be still exploited offline particularly on the patch level. At this level, the images share similarity. These images are initially classified using a spatial-pyramid image representation based on sparse codes (SC) of SIFT features and learning the visual prior with a dictionary.

The sparse coding at different scales gives the object representation. With this representation, positive and negative samples from a frame are distinguished and the tracking task is carried within Bayesian inference framework using Sparse prototype. For most tracking algorithms in the literature, either strong prior information of the target object is assumed or no prior knowledge is exploited. However, the tracking results after a long duration period are usually unpredictable as only online visual information is used. The algorithm used here exploits the strength of both approaches. In particular, the proposed algorithm learns generic visual prior offline and transfers such knowledge to online object tracking.

The contributions of our tracking algorithm are summarized as follows. First, we learn a generic visual prior offline without assuming any specific knowledge of the target object for tracking. Second, the prior is represented by an over complete dictionary and learned by sparse coding from local patches. It is different from the widely used orthogonal dictionary (subspace) learned from holistic images by principal component analysis (PCA) or its variants. Third, with the learned dictionary, sparse coding, and multiscale max pooling, a high-level object representation is constructed and tracking is carried out using Sparse Prototype.

II. Related Work

There is a rich literature in object tracking, and a thorough review on this topic can be found in [15]. To deal with the problem of large object and background appearance variations, most recent tracking algorithms have focused on

S.Divya is with Anna University:: Regional Campus, Tiruchirapalli, Tiruchirapalli-620024;(e-mail: divyasudevan.dct@gmail.com).

Dr.K.Latha, Assistant Professor is with Anna University:: Regional Campus, Tiruchirapalli, Tiruchirapalli-620024; (email:erklatha@gmail.com).

978-1-4673-4866-9/13/$31.00 ©2013 IEEE

developing robust object representation schemes. Based on a specific prior of the target, an object model can be learned offline. Black and Jepson [3] learn a subspace model to represent target objects at fixed views. Avidan [1] uses a set of vehicle and nonvehicle images collected offline to learn a classifier for car tracking. All these methods heavily depend on the specific prior. However, in most real-world tracking applications, it is difficult to enumerate all possible appearance variations of objects. Therefore, such tracking algorithms have limited application domains. Numerous adaptive appearance models have been recently proposed for object tracking. In these algorithms, object representation can be initialized and updated with online observations without any prior. Jepson et al. [6] learn a Gaussian mixture model via an online expectation maximization algorithm to account for target appearance variations during tracking. To overcome the problem of partial occlusion, sparse representation has been also utilized for object tracking [9].In [12] the authors extend the conventional particle filtering framework with multiple dynamic and observation models to account for target appearance variation caused by change of pose, illumination, scale, and partial occlusion.

Sparse coding algorithms model an observed example as a linear combination of a few elements from an over complete dictionary. The recent development of sparse coding/ representation has attracted much interest and has been used in image denoising [10], image classification [13], and object tracking [11]. These methods have proven that a learning dictionary from data outperforms prechosen (fixed) ones (e.g., wavelet) since the former can significantly reduce reconstruction error. Different from the representations based on PCA and its variants, sparse models do not impose that the bases in the dictionary be orthogonal, which allows more flexibility to adapt the representation to the data [10]. In this paper, we propose a robust generative tracking algorithm with adaptive appearance model which handles partial occlusion and other challenging factors. By exploiting the advantage of subspace representation, our algorithm is able to process higher resolution image observations, and performs more efficiently with favorable results than the existing method based on sparse representation of templates [12].

III. Learning Visual Priorwith Sparse Coding

We first present how visual prior is learned from numerous images of various object classes. Although we can get a large number of real-world images, there is no straightforward method to use and correspond to generic visual prior in the tracking literature. In this paper, sparse coding is used to learn the visual prior from large image sets of objects with an over complete dictionary.

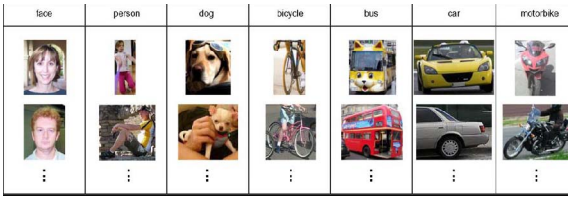

Fig.1 Sample Images for learning Visual Prior

A. *Image Set*

This paper aims to associate object recognition (based on visual prior) and object tracking (based on prior or online information). On the square level, small images share structural similarity. This is why we make use of such prior information offline from existing data sets and use it for online visual tracking. The VOC20101 and Caltech1012 data sets, which consist of a large variety of objects, are used for learning visual prior. We use object classes that are common in surveillance scenarios from the two data sets, including nonrigid (e.g., face, person, and dog) and rigid (e.g., bicycle, bus, car, and motorbike) objects. Some images of these classes are shown in Fig. 1.

B. *Learning Dictionary*

Since sparse coding based on SIFT descriptor has been proved to outperform sparse coding on raw image patches in computer vision [14], we also choose SIFT as the basic appearance descriptor in our tracking method. We extract the SIFT descriptors from overlapped patches of each gray scale image and learn the dictionary in an unsupervised manner. Let$X = [x_1 \ldots . x_n] \varepsilon\, R^{m*n}$ be the SIFT descriptors we extract from the image set, where m and n are the dimensionality of each SIFT descriptor and the number of SIFT descriptors, respectively. The dictionary is denoted as $D = [d_1 \ldots . d_n] \varepsilon\, R^{m*k} (k \gg m)$. The images in the dictionary is classified as

$$\min_{D,\{a_i\}} \frac{1}{2} \sum_{i=1}^{n} ||x_i - Da_i||^2 + \beta \sum_{i}^{n} ||a_i||_1$$
$$\text{Subject to } ||d_j||_2^2 \leq 1 \; \forall_j \varepsilon \{1, \ldots\ldots, k\} \qquad (1)$$

Where $a_i \; \varepsilon \; R^k$ is the sparse coefficient vector of x_i Parameter β is a tradeoff between reconstruction error and sparsity. To enlarge the sparsity, we can increase β and vice versa. Although there is a large number of SIFT descriptors extracted from the data set, is learned offline with the sparse coding method proposed in [9].

IV. Object Representation From Learned Prior

The learned visual prior is represented by the over complete dictionary D. For object tracking we transfer this prior by representing object with D. For each SIFT descriptor a sparse coefficient vector is learned by performing l1/l2 sparse coding on the dictionary. Then, an object is represented by applying multiscale max pooling.

A. *l1/l2 Sparse Coding*

To represent an object, we first extract the SIFT descriptors from their image patches and then encode them with the learned dictionary. Let $X = [x_1 \ldots . x_n] \varepsilon\, R^{m*n}$ denote the SIFT descriptors extracted from an object image, the l1/l2 sparse coefficient vector is calculated by

$$\min_{a_i} \frac{1}{2} || x_i - Da_i||_2^2 + \lambda_1 ||a_i||_1 + \frac{\lambda^2}{2} ||a_i||_2^2 \qquad (2)$$

When ƛ2 = 0,it leads to the l1-norm sparse coding problem, which has been widely used in [9] .With l1/l2 sparse coding, the SIFT descriptors from different objects can be encoded by different bases in the dictionary. Thus, sparse coding can achieve a much lower reconstruction error.

B. *Multiscale Max Pooling*

For the tracking task, we need to define an object-level feature for a target or a background sample over the sparse representation matrix. For representing an object with a set of descriptors, we use a pooling function that operates on each row of A and obtain a vector bЄR^k .To make the representation more robust to local spatial translations, we use the max pooling function on the absolute sparse codes

$$b_i = max\{|a_{i,1}|, \ldots\ldots, |a_{i,N}|\} \quad (3)$$

wherebi is the th element of b and $a_{i,}$j is the element of the ith row and the jth column of A .

We use multiscale max pooling to obtain the object-level representation by preserving spatial information and local invariance [11].This pooling process searches different locations and different scales of the object image and combines all local maximum responses. In this paper, it is implemented by dividing the whole object image into M non overlapped spatial cells, applying max pooling on the coding results of descriptors in each cell and concatenating the pooled features from all the spatial cells.

$$z = [b_1^T, \ldots\ldots, b_M^T] \quad (4)$$

V. Object Trackingvia Sparse Prototypes

After extraction of SIFT features, sparse coding, and multiscale max pooling, we obtain a spatial pyramid representation for each object image.

A. Proposed Tracking Algorithm

In this paper, Object tracking is considered as a Bayesian inference task in a Markov model with hidden state variables [3]. The sparse representation of the object is shown in Fig.2. Given a set of observed images Y_T= {y_1….. y_T} at the t-th frame the hidden state variable xt is estimated recursively as,

$$p(x_t|Y_t) \alpha p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|Y_{t-1}) dx_{t-1} \quad (5)$$

where$p(x_t|x_{t-1})$ represents the dynamic (motion) modelbetween two consecutive states, and$p(y_t|x_t)$ denotes observation model that estimates the likelihood of observing $y_t$at statex_t . The optimal state of the tracked target given all the observations up to t-th frame is obtained by the maximum a posteriori estimation over N samples at time t by

$$\widehat{x_t} = arg_{x_t^i} max\, p(y_t^i|x_t^i) p(x_t^i|x_{t-1}), i = 1,2, \ldots\ldots, N \quad (6)$$

Where x_t^i indicates the i-th sample of the state x_t , and y_t^i denotes the image patch predicated by x_t^i.

B. Dynamic Model

In this paper, we apply an affine image warp to model the target motion between two consecutive frames. The six parameters of the affine transform are used to model $p(x_t|x_{t-1})$of a tracked target. Let $X_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \emptyset_t\}$, where $x_t, y_t, \theta_t, s_t, \alpha_t, \emptyset_t$denote x, y translations, rotation angle, scale, aspect ratio, and skew respectively. The state transition is formulated by,

$p(X_t|X_{t-1}) = N(X_t, X_{t-1}, \varphi)$ where φ is the diagonal covariance matrix.

C. Observation Model

If no occlusion occurs, an image observation yt can be assumed to be generated from a subspace of the target object spanned by U and centered at μ. However, it is necessary to account for partial occlusion in an appearance model for robust object tracking. We assume that a centered image observation $\overline{y_t}(\overline{y_t} = y_t - \mu)$ of the tracked object can be representedby a linear combination of the PCA basis vectors U and few elements of the identity matrix I (i.e., trivial templates). If there is no occlusion, the most likely image patch can be effectively represented by the PCA basis vectors and coefficients tend to be zeros. If partial occlusion occurs, the most likely image patch can be represented as a linear combination of PCA basis vectors and very few numbers of trivial templates.

For each observation corresponding to a predicted state, we solve the following equation efficiently using the proposed algorithm as summarized in Table I,

$$L(z^i, e^i) = \min_{z^t, e^t} \frac{1}{2} \|\bar{y}^i - Uz^i - e^i\| + \lambda \|e^i\| \quad (7)$$

and obtain $z^i$and e^i , where i denotes the i-th sample of the state x.

The observation likelihood can be measured by their construction error of each observed image patch,

$$p(\bar{y}^i|x^i) = \exp(-||\bar{y}^i - Uz^i||_2^2) \quad (8)$$

However, Eq. 8 does not consider occlusion. Thus, we use a mask to factor out non-occluding and occluding parts,

$$p(\bar{y}^i|x^i) = \exp\left[-\left(w^i \cdot (\bar{y}^i - Uz^i)||_2^2 + \beta \sum (1 - w^i)\right)\right] \quad (9)$$

Where$w^i = [w_{1,}^i w_2^i, \ldots., w_d^i]$ is a vector that indicates the zero elements of e^i

D. Tracking Architecture Design

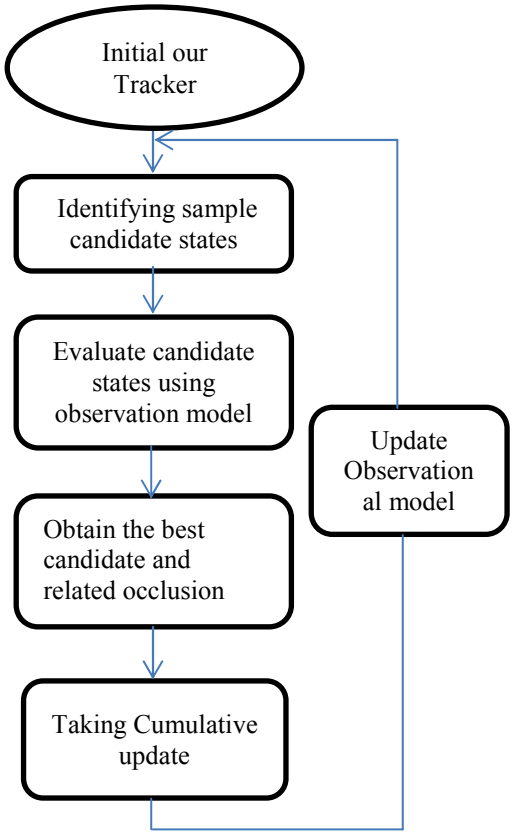

Fig.2 Our Tracking Algorithm

E. Update of Observation Model

It is essential to update the observation model for handling appearance change of a target object for visual tracking. We explore the trivial coefficients for occlusion detection since the corresponding templates are used to account for noise. First, each trivial coefficient vector corresponds to a 2D map as a result of reverse raster scan of an image patch. A non-zero element of this map indicates that pixel is occluded (referred as occlusion map). Second, we compute the ratio , of the number of nonzero pixels and the number of occlusion map pixels. We use two thresholds tr1 and tr2 to describe the degree of occlusion. If < tr1, we directly update the model with this sample. If tr1 << tr2, it indicates that the target is partially occluded. We then replace the occluded pixels by its corresponding parts of the average observationμ, and use this recovered sample for update. Otherwise if > tr2, it means that a significant part of the target object is occluded, and we discard this sample without update.

VI. Proposed Tracking Algorithm

1: Input: Video frames y_1….. y_T
2: Output: Target states x_1…..x_T.
3: for t=1 to T do
4: if t=1 then
5: Transfer prior for object representation.
6: Initialize the classifier with parameter set w1.
7: else
8: Transfer prior for object representation.
9: Estimate xt from t-th frame.
10: Store target observation corresponding to zt .
11: if the number of target observations is equal to some predefined threshold then
12: Collect a number of negative samples in the current frame.
13: Use the target observations (positive samples) and negative samples to update wt.
14: Clear the target observation set.
15: else
16:w_t=w_{t-1}
17: end if
18: end if
19: end for

VIII Experimental Results

We evaluate our tracker on 3 challenging image sequences (some of them are publicly available) against several state-of-the- art algorithms. The challenging factors in these sequences include, pose, occlusion, cluttered background, image blur.

A.Implementation

The proposed algorithm consists of an offline prior learning module and an online object tracking component. In the offline phase, the SIFT descriptors are densely extracted from 16 x16 patches on a grid with step size of 8 pixels from each selected image (based on intensity). With about 200 000 SIFT descriptors, we learn a 128x1024 dictionary. We compare our tracker with several object tracking algorithms, i.e., the incremental visual tracker (IVT) [13], the L1 tracker (L1T) [12], the MIL tracker (MILT) [2], the visual tracking decomposition tracker (VTD) [7]

B. Image blur and low contrast

In the car sequence, the target also undergoes partial occlusion and poses variation other than image blur caused by camera motion. Our tracker performs well in this sequence, whereas the L1T method quickly fails when the car is partially occluded by a bus. The IVT and perform better than L1T, but they lose track of the target when image blur occurs. The MILT, VTD, method are able to track the target in this sequence although with some errors in the last frames.

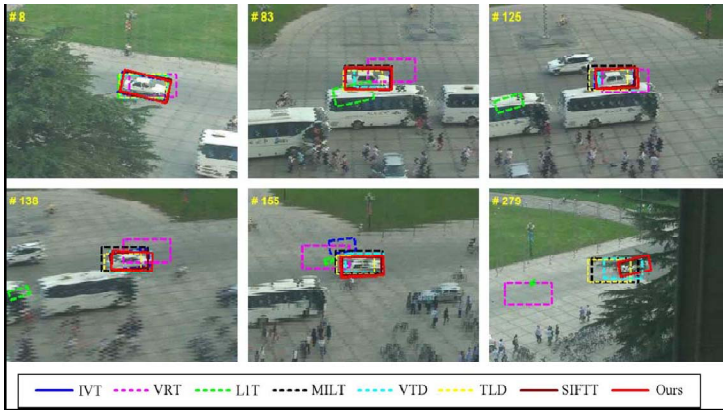

Fig.3 Car Sequence

B. Occlusion

In the CAVIAR sequence, it is difficult to keep track of the target after occlusion because there are other objects with similar appearances in the scene. The IVT, L1T, MILT, VTD, methods do not perform well, whereas the VRT algorithm performs slightly better. In contrast, our method exploits visual prior and represents objects by coding results of local image patches described by SIFT features for learning a target-specific classifier. The initial classifier also facilitates the proposed method to keep track of the target when heavy occlusion occurs.

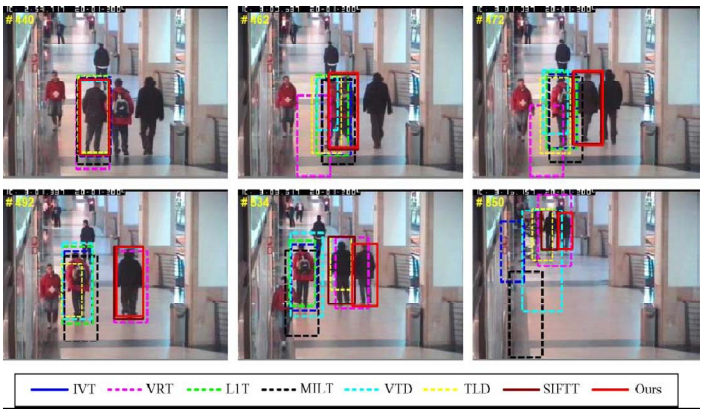

Fig.4 CAVIAR

C. Illumination change

In the shaking sequence, the target undergoes pose variation besides illumination change. The L1T and MILT algorithms are also able to track the target, whereas the IVT,

and TLD methods drift from the target quickly. Our tracker uses an online update mechanism to account for the appearance variation of the target and background over time and retains a detector to alleviate visual drift problem. In addition, the object representation based on sparse coding and multiscale max pooling is less sensitive to illumination and pose change, thereby achieving good tracking performance.

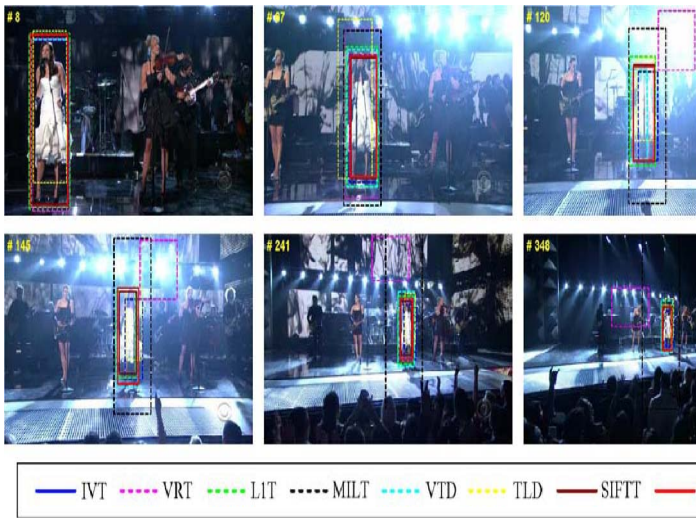

Fig.5 Singing

B.Quantitative Evaluation

Aside from the qualitative comparison, we compute the tracking success rate and center location error using the ground truth manually labeled at every five frames. We employ the criterion used in the PASCAL Visual Object Classes challenge [5] to determine whether each tracking result is a success.

The rightmost column and the second column from the right of Table1 show the implementation results of our tracker algorithm.

Methods / Test Sequences	IVT	L1	MIL	Ours Algm Eq.6	Ours Algm Eq.9
Car	0.92	0.84	0.34	0.92	0.92
Singer	0.66	0.70	0.34	0.84	0.82
Caviar	0.28	0.28	0.25	0.28	0.89

Table.1 Overall rating of tracking methods

IX Conclusion

This paper has exploited generic visual prior learned from real-world images for online tracking of specific objects. We have presented an effective method that learns and transfers visual prior for robust object tracking. With a large set of natural images, we represent visual prior with an over complete dictionary. We transfer the learned prior to tracking tasks by sparse codes and sparse prototypes and represent the object with the multiscale max pooling method. Compared with the related state-of-the-art tracking methods, the proposed tracking algorithm is demonstrated to robustly perform in complex environments where the target and background undergo different kinds of variations.

REFERENCES

[1] S. Avidan, “Ensemble tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2005, vol. 2, pp. 494–501.

[2] B. Babenko, M.-H.Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2009, pp. 983–990.

[3] M. Black andA. Jepson, “Eigentracking: Robust matching and trackingof articulated objects using a view-based representation,” in Proc. Eur.Conf. Comput. Vis., 1996, pp. 329–342.

[4] D. Comaniciu, V. R. Member, and P. Meer. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5):564–575, 2003.

[5] M. Everingham, L.VanGool,C.Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (voc) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, Jun. 2010.

[6] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, “Robust online appearance models for visual tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[7] J. Kwon and K. Lee, “Visual tracking decomposition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2010, pp. 1269–1276.

[8]X. Li and W. Hu, “Robust visual tracking based on incremental tensor subspace learning,” in Proc. IEEE Int. Conf. Comput. Vis., 2007, pp. 1–8.

[9] H. Lee, A. Battle, R. Raina, and A. Ng, “Efficient sparse coding algorithms,” Adv. Neural Inf. Process. Syst., vol. 19, pp. 801–808, 2007.

[10] J. Mairal, F. Bach, and J. Ponce, “Task-driven dictionary learning,”INRIA, Rocquencourt, France, Tech. Rep. 7400, 2010.

[11] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Nonlocalsparse models for image restoration,” in Proc. IEEE Int. Conf. Comput.Vis., 2009, pp. 2272–2279.

[12] X. Mei and H. Ling, “Robust visual tracking using l1 minimization,” in Proc. IEEE Int. Conf. Comput. Vis., 2009, pp. 1436–1443.

[13] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” Int. J. Comput.Vis., vol. 77, no. 1–3, pp.125–141, 2008.

[14] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2009, pp. 1794–1801.

[15] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” ACM Comput. Surveys, vol. 38, no.4, p. 13, 2006.