Volume 3, Issue 5, May 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Compound Order Tracking Based on Multi Sparse Representation

Khemraj Deshmukh
Department of Electronics and Telecommunication
India.

Vishal Moyal
Department of Electronics and Telecommunication
India.

Abstract—Object tracking has been a very active research topic in the last few years due to its growing importance in various applications. Performance of object tracking can be reduces due to factors such as pose variation, illumination change, occlusion, and motion blur. Existing online algorithms often encounter problems such as drift and lack of data for learning. In this paper we presented an efficient online tracking algorithm with an appearance model based on multi scale image feature space with data independent basis. Proposed appearance model do not change the feature of object in iteration. Compound order samples of both background and foreground has been taken. These samples are reduced in size using sparse measurement matrix. Classification and updating of these samples is done in compressed domain. A MATLAB code has been implemented as per the proposed method and it gives us good result of tracking in simple and difficult sequences.

Keywords—Compound Order Learning, Online Tracking, Compressed sensing.

I. INTRODUCTION

Tracking is an essential component of several vision applications. Still, robust and accurate tracking of a deforming, non-rigid and fast moving object without getting restricted to particular model assumptions presents a major challenge [3].

Although many tracking methods employ static appearance models that are either defined manually or trained using only the first frame these methods are often unable to cope with significant appearance changes. These challenges are particularly difficult when there is limited a priori knowledge about the object of interest. In this scenario, it has been shown that an adaptive appearance model, which evolves during the tracking process as the appearance of the object changes, is the key to good performance [4]. Training adaptive appearance models, however, is itself a difficult task with many questions yet to be answered. Such models often involve many parameters that must be tuned to get good performance (e.g. "forgetting factors" that control how fast the appearance model can change), and can suffer from drift problems when an object undergoes partial occlusion.

Grabner et al. [1] propose an online semi-supervised boosting method to alleviate the drift problem in which only the samples in the first frame are labeled and all the other samples are unlabeled. Babenko et al. [4] introduce multiple instance learning into online tracking where samples are considered within positive and negative bags or sets. Recently, a semi-supervised learning approach [11] is developed in which positive and negative samples are selected via an online classifier with structural constraints.

Hanxi Li[9] proposed a Real Time Compressed Sensing Tracking (RTCST) by exploiting the signal recovery power of Compressed Sensing (CS). Dimensionality reduction and a customized Orthogonal Matching Pursuit (OMP) algorithm are adopted to accelerate the CS tracking. This algorithm is presented a good idea of compressive tracking with very sparse representation.

In this paper, we propose an effective and efficient tracking algorithm with an appearance model based on features extracted in the compressed domain. The main components of our compound order tracking algorithm are shown by Figure 1. Our appearance model is generative as the object can be well represented based on the features extracted in the compressive domain. It is also discriminative because we use these features to separate the target from the surrounding background via a naive Bayes classifier. In our appearance model, features are selected by an information-preserving and non-adaptive dimensionality reduction from the multi-scale image feature space based on compressive sensing theories [10, 11]. It has been demonstrated that a small number of randomly generated linear measurements can preserve most of the salient information and allow almost perfect reconstruction of the signal if the signal is compressible such as natural images or audio [10–12]. We use a very sparse measurement matrix that satisfies the restricted isometry property (RIP) [13], thereby facilitating efficient projection from the image feature space to a low-dimensional compressed subspace. For tracking, the positive and negative samples are projected (i.e., compressed) with the same sparse measurement matrix and discriminated by a simple naive Bayes classifier learned online. The proposed compressive tracking algorithm runs at real-time and performs favorably against state-of-the-art trackers on challenging sequences in terms of efficiency, accuracy and robustness.

II. PROPOSED APPROACH

Figure 1 shows the complete work flow of compound order compact tracking algorithm. This method has mainly five steps, which are as follows:

1. Initial Target marking

2. Collection of Classifiers
3. Sparse Decomposition
4. Finding Maximal in Positive Classifiers
5. Updating of Classifier till the last frame

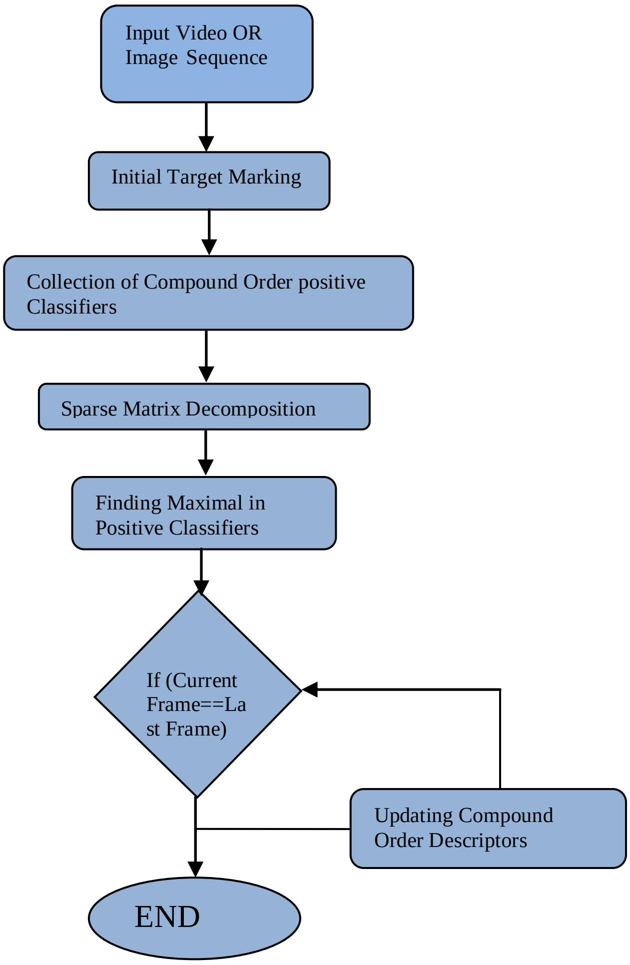


Fig. 1 Block diagram of proposed Tracking method

Our image representation consists of a set of Haar-like features that are computed for each image patch [14], [15]; this is discussed in more detail in Section 3.6. The appearance model is composed of a discriminative descriptor which is able to return $p(y = 1|X)$ (we will use $p(y|X)$ as shorthand), where X is an image patch (or the representation of an image patch in feature space) and y is a binary variable indicating the presence of the object of interest in that image patch. At every time step t, our tracker maintains the object location l_t^*. Let $l(X)$ denote the location of image patch X (for now let's assume this consists of only the (X, Y) coordinates of the patch center, and that scale is fixed; below we consider tracking scale as well). For each new frame we crop out a set of image patches $X^s = \{x: ||l(x - l_{t-1}^*)|| < s\}$ that are within some search radius s of the current tracker location, and compute $p(y|x)$ for all $x \in X^s$. We then use a greedy strategy to update the tracker location:

$$l_t^* = l\left(\underset{x \in X^s}{\operatorname{argmax}} p(y|x)\right)$$

In other words, we do not maintain a distribution of the target's location at every frame, and our motion model is such that the location of the tracker at time t is equally likely to appear within a radius s of the tracker location at time $(t + 1)$:

$$p(l_t^*|l_{t-1}^*) \alpha \begin{cases} 1 & \text{if } ||l_t^* - l_{t-1}^*|| < s \\ 0 & \text{otherwise} \end{cases}$$

This could be extended with something more sophisticated, such as a particle filter, as is done in [16], [17]; however, we again emphasize that our focus is on the appearance model.

Once the tracker location is updated, we proceed to update the appearance model. We crop out a set of patches $X^r = \{x : ||l_t^* - l_{t-1}^*|| < r\}$, where r < s is a scalar radius (measured in pixels), and label this bag positive (recall that in MIL we train the algorithm with labeled bags). In contrast, if a standard learning algorithm were used, there would be two options: set r = 1 and use this as a single positive instance, or set r > 1 and label all these instances positive. For negatives we crop out patches from an annular region $X^{r,\beta} = \{ x : r < \ ||l_t^{*} - l_{t-1}^{*}|| \ < \beta\}$, where r is same as before, and is another scalar. Since this generates a potentially large set, we then take a random subset of these image patches and label them negative. We place each negative example into its own negative bag, though placing them all into one negative bag yields the same result.

Incorporating scale tracking into this system is straightforward. First, we define an extra parameter λ to be the scale space step size. When searching for the location of the object in a new frame, we crop out image patches from the image at the current scale, , as well as one scale step larger and smaller, $l_t^s \pm$; once we find the location with the maximum response, we update the current state (both position and scale) accordingly. When updating the appearance model, we have the option of cropping training image patches only from the current scale, or from the neighboring scales as well; in our current implementation we do the former

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

A. Page Layout

Your paper must use a page size corresponding to A4 which is 210mm (8.27") wide and 297mm (11.69") long. The margins must be set as follows:

- Top = 19mm (0.75")
- Bottom = 19mm (0.75")
- Left = Right = 19mm (0.75")

III. COMPOUND ORDER LEARNING

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified. Traditional discriminative learning algorithms for training a binary descriptor that estimates $p(y|x)$ a training data set of the form $\{(x_1,y_1), \dots\dots, (x_n,y_n)\}$ where x_i is an instance (in our case a feature vector computed for an image patch), and $y_i \in (0,1)$ is a binary label. In Compound Order Learning training data has the form $\{(x_1,y_1), \dots\dots, (x_n,y_n)\}$ where a bag $X_i = \{x_{i1}, \dots\dots, x_{im}\}$ and y_i is a bag label. The bag labels are defined as:

$$y_i = \max_j (y_{ij})$$

where y_{ij} are the instance labels, which are not known during training. In other words, a bag is considered positive if it contains at least one positive instance. Numerous algorithms have been proposed for solving the MIL problem [11], [18], [19]. The algorithm that is most closely related to our work is the MILBoost algorithm proposed by Viola et al. in [18]. MILBoost uses the the gradient boosting framework [20] to train a boosting descriptor that maximizes the log likelihood of bags:

$$\mathcal{L} = \sum_i (\log p(y_i|X_i))$$

Notice that the likelihood is defined over bags and not instances, because instance labels are unknown during training, and yet the goal is to train an instance descriptor that estimates $p(y|x)$ We therefore need to express $p(y_i|X_i)$, the probability of a bag being positive, in terms of its instances. In [18] the Noisy-OR (NOR) model is adopted for doing this:

$$p(y_i|X_i) = 1 - \prod_j (1 - p(y_i|X_i))$$

although other models could be swapped in (e.g. [21]). The equation above has the desired property that if one of the instances in a bag has a high probability, the bag probability will be high as well. As mentioned in [18], with this formulation, the likelihood is the same whether we put all the negative instances in one bag, or if we put each in its own bag. Intuitively this makes sense because no matter how we arrange things, we know that every instance in a negative bag is negative.We refer the reader to [18] for further details on MILBoost. Finally, we note that MILBoost is a batch algorithm (meaning it needs the entire training data set at once) and cannot be trained in an online manner as we need in our tracking application. Nevertheless, we adopt the loss function in Equation 4 and the bag probability model in Equation 5 when proposed algorithm will be developed.

IV. ONLINE COMPOUND ORDER TRACKING

The algorithms in [23] and [22] rely on the special properties of the exponential loss function of AdaBoost, and therefore cannot be readily adapted to the MIL problem. We now present our novel online boosting algorithm for MIL. As in [40], we take a statistical view of boosting, where the algorithm is trying to optimize a specific objective function J. In this view, the weak descriptors are chosen sequentially to optimize the following criteria:

$$(h_k, \alpha_k) = \underset{h \epsilon H, \alpha}{\operatorname{argmax}} J(H_{k-1} + \alpha h)$$

where H_{k-1} is the strong descriptor made up of the first $(k-1)$ weak descriptors, and H is the set of all possible weak descriptors. In batch boosting algorithms, the objective function J is computed over the entire training data set.

In our case, for the current video frame we are given a training data set $\{(x_1,y_1),\ldots\ldots,(x_n,y_n)\}$, where $X_i = \{x_{i1},\ldots,x_{im}\}$. We would like to update our descriptor to maximize log likelihood of this data (Equation 4). We model the instance probability as

$$p(y|x) = \sigma(H(x))$$

Where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function; the bag probabilities $p(y|X)$ are modeled using the NOR model in Equation 5. To simplify the problem, we absorb the scalar weights α_t into the weak descriptors, by allowing them to return real values rather than binary.

At all times our algorithm maintains a pool of M > K candidate weak stump descriptors h. To update the descriptor, we first update all weak descriptors in parallel, similar to [22]. Note that although instances are in bags, the weak descriptors I n a MIL algorithm are instance descriptors, and therefore require instance labels y_{ij} . Since these are unavailable, we pass in the bag label y_i for all instances x_{ij} to the weak training procedure. We then choose K weak descriptors h from the candidate pool sequentially, by maximizing the log likelihood of bags:

$$h_k = \arg\max_{h\in(h_g,\ldots h_M)} \mathcal{L}(H_{k-1}+h)$$

See Algorithm 2 for the pseudo-code of Online MILBoost and Fig. 3 for an illustration of tracking with this algorithm

Online Compound Order Tracking can be enhanced in terms of speed and memory required by the use of some dimensionality reduction technique. Proposed algorithm uses the sparse decomposition method to reduce the matrix dimension.

A random matrix $R \in \mathbb{R}^{nXm}$ whose rows have unit length projects data from the high-dimensional image space $X \in \mathbb{R}^m$ to a lower-dimensional space $V \in \mathbb{R}^n$.

$$V = RX$$

where $n \ll m$. Ideally, we expect R provides a stable embedding that approximately preserves the distance between all pairs of original signals. The Johnson- Lindenstrauss lemma [16] states that with high probability the distances between the points in a vector space are preserved if they are projected onto a randomly selected subspace with suitably high dimensions. Baraniuk et al. [17] proved that the random matrix satisfying the Johnson-Lindenstrauss lemma also holds true for the restricted isometry property in compressive sensing. Therefore, if the random matrix R in (1) satisfies the Johnson-Lindenstrauss lemma, we can reconstruct x with minimum error from v with high probability if x is compressive such as audio or image. We can ensure that v preserves almost all the information in x. This very strong theoretical support motivates us to analyze the high-dimensional signals via its low-dimensional random projections. In the proposed algorithm, we use a very sparse matrix that not only satisfies the Johnson-Lindenstrauss lemma, but also can be efficiently computed for real-time tracking.

A typical measurement matrix satisfying the restricted isometry property is the random Gaussian matrix $R \in \mathbb{R}^{nXm}$ where $r_{ij} \sim N(0,1)$, as used in numerous works recently [29, 5, 30]. However, as the matrix is dense, the memory and computational loads are still large when m is large. In this paper, we adopt a very sparse random measurement matrix with entries defined as

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 \text{ with probability } \frac{1}{2s} \\ 0 \text{ with probability } 1-\frac{1}{s} \\ -1 \text{ with probability } \frac{1}{2s} \end{cases}$$

Achlioptas [24] proved that this type of matrix with s = 2 or 3 satisfies the Johnson-Lindenstrauss lemma. This matrix is very easy to compute which requires only a uniform random generator. More importantly, when s = 3, it is very sparse where two thirds of the computation can be avoided. In addition, Li et al. [31] showed that for $S = O(m)X \in \mathbb{R}^m$ this matrix is asymptotically normal. Even when $S = m/\log(m)$, the random projections are almost as accurate as the conventional random projections where $r_{ij} \sim N(0,1)$. In this work, we set $s = m/4$ which makes a very sparse random matrix. For each row of R, only about $c, c \leq 4$, entries need to be computed. Therefore, the computational complexity is only $O(cn)$ which is very low. Furthermore, we only need to store the nonzero entries of R which makes the memory requirement also very light.

For each sample $Z \in \mathbb{R}^{wXh}$, to deal with the scale problem, we represent it by convolving z with a set of rectangle filters at multiple scales $\{h_{1,1}, \dots \dots \dots, h_{w,h}\}$ defined as

$$h_{i,j}(x,y) = \begin{cases} 1, & 1 \leq x \leq i, 1 \leq y \leq j \\ 0, & \text{otherwise} \end{cases}$$

where and are the width and height of a rectangle filter, respectively. Then, we represent each filtered image as a column vector in $\mathbb{I}$ and then concatenate these vectors as a very high-dimensional multi-scale image feature vector $X = \{x_1, \dots\dots, x_m\}^T \in$ where $m = (w$. The dimensionality is typically in the order of 106 to 1010. We adopt a sparse random matrix in (2) with $s = n$ to project x onto a vector $v \in$ in a low-dimensional space. The random matrix R needs to be computed only once off-line and remains fixed throughout the tracking process. For the sparse matrix in (2), the computational load is very light. As shown by Figure 2, we only need to store the nonzero entries in and the positions of rectangle filters in an input image corresponding to the nonzero entries in each row of . Then, v can be efficiently computed by using to sparsely measure the rectangular features which can be efficiently computed using the integral image method [14].

A. Document Descriptor Construction and Detail

The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes. shown in Figure 3.4, each element vi in the low-dimensional feature $v \in \mathbb{R}^n$ is a linear combination of spatially distributed rectangle features at different scales. As the coefficients in the measurement matrix can be positive or negative (via (2)), the compressive features compute the relative intensity difference in a way similar to the generalized Haar-like features (See also Figure 2). The Haar-like features have been widely used for object detection with demonstrated success. The basic types of these Haar-like features are typically designed for different tasks [1, 21]. There often exist a very large number of Haar-like features which make the computational load very heavy. This problem is alleviated by boosting algorithms for selecting important features [20, 21]. Recently, Babenko et al. [8] adopted the generalized Haar-like features where each one is a linear combination of randomly generated rectangle features, and use online boosting to select a small set of them for object tracking. In our work, the large sets of Haar-like features are compressively sensed with a very sparse measurement matrix. The compressive sensing theories ensure that the extracted features of our algorithm preserve almost all the information of the original image. Therefore, we can classify the projected features in the compressed domain efficiently without curse of dimensionality. In the matrix R, dark, gray and white rectangles represent negative, positive, and zero entries, respectively. The blue arrows illustrate that one of nonzero entries of one row of R sensing an element in x is equivalent to a rectangle filter convolving the intensity at a fixed position of an input image.

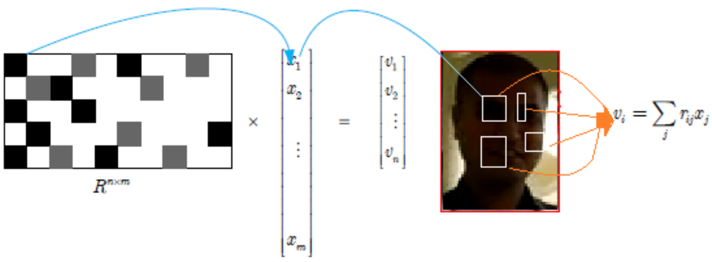


Fig 2. Graphical representation of compressing a high-dimensional vector x to a low dimensional vector v.

For each sample $z \in \mathbb{R}^m$, its low-dimensional representation is $v = \{v_1, \dots\dots, v_n\}^T \in \mathbb{R}^n$ with $m \gg n$. We assume all elements in v are independently distributed and model them with a naive Bayes descriptor [32],

$$H(v) = \frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)} = \sum_{i=1}^{n} \log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right)$$

where we assume uniform prior, $p(y = 1) = p(y = 0)$, and $y \in \{0, 1\}$ is a binary variable which represents the sample label. Diaconis and Freedman [23] showed that the random projections of high dimensional random vectors are almost always Gaussian. Thus, the conditional distributions $p(v_i|y = 1)$ and $p(v_i|y = 0)$ in the descriptor $H(v)$ are assumed to be Gaussian distributed with four parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ where

$$p(v_i|y=1) \sim N((\mu_i^1, \sigma_i^1), \quad p(v_i|y=0) \sim N((\mu_i^0, \sigma_i^0)$$

The scalar parameters in above equation are incrementally updated.

V. Online Compound Order Tracking

In this section, we describe the application of the proposed method to different challenging video inputs online available. We use two criteria, tracking success rate and location error with respect to object center, for quantitative evaluations. The performance of proposed algorithm is measured on a 2.10 GHz Pentium Dual-Core laptop machine with 1.0 GB main memory, running on Windows-7 operating system. All programs were developed in MATLAB, version R2010a. We perform our experiments on 5 publicly available video sequences, as well as 4 of our own. For all sequences we labeled the ground truth center of the object for every 5 frames, and interpolated the location in the other frames. Objective evaluations are performed in the tracking outputs. Proposed method also quantitatively compared with different model on different image Sequences. The test algorithms include: incremental visual tracker (IVT),1 variance ratio tracker (VRT),2 online boosting tracker (BoostT),3 l_1 tracker (L1T),4 and proposed algorithm (COCT),5.

Figure 3 shows screenshots of some tracking example. Evaluation result is presented in the form of table. Table 1 shows success rate for different sequences and different applied algorithms. Table 2 shows average centre location error for different sequences and different applied algorithms.

TABLE I
SUCCESS RATE (SR) (%)

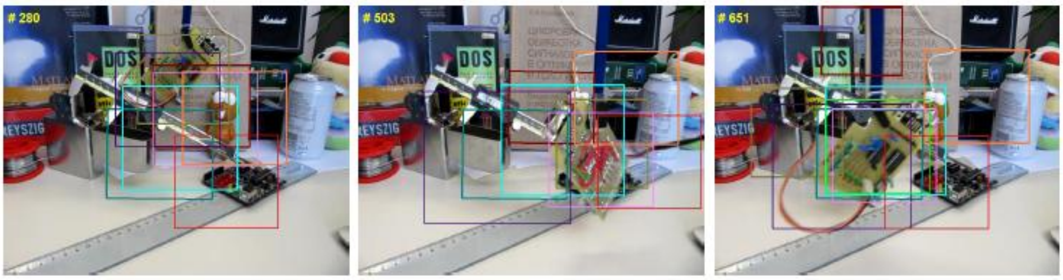
Algorithms ➡ Sequences ⬇	**IVT**	**VRT**	**BoostT**	**L1T**	**COCT**
Ant	30%	40%	35%	70%	89%
Bolt	10%	16%	1%	2%	79%
Car	50%	9%	91%	29%	90%
Cycle	11%	38%	12%	74%	75%
Khem	-	-	-	-	85%
Toy	13%	20%	22%	57%	23%

TABLE III
AVERAGE CENTRE LOCATION ERRORS (IN PIXELS)

Algorithms ➡ Sequences ⬇	**IVT**	**VRT**	**BoostT**	**L1T**	**COCT**
Ant	51	59	46	21	5
Bolt	196	157	227	58	8
Car	167	113	154	61	14
Cycle	38	14	12	74	11
Khem					41
Toy	22	37	22	57	13

VI. Conclusions

The version of this template is V2. Most of the formatting instructions in this document have been compiled by Causal Productions from the IEEE LaTeX style files. Causal Productions offers both A4 templates and US Letter templates for LaTeX and Microsoft Word. The LaTeX templates depend on the official IEEEtran.cls and IEEEtran.bst files, whereas the Microsoft Word templates are self-contained. Causal Productions has used its best efforts to ensure that the templates have the same appearance. In this paper we presented a way of updating an adaptive appearance model of a tracking system. We argued that using Compound order Learning to train the appearance classifier results in more robust tracking. A very sparse measurement matrix was adopted to efficiently compress features from the foreground targets and background ones. The tracking task was formulated as a binary classification problem with online update in the compressed domain.



(a) Board

(b) Toy

Fig 3: Screenshot of some Tracking example

REFERENCES

[1] 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).

[2] Thomas Brox and Joachim Weickert, " Level Set Based Image Segmentation with Multiple Regions," In Pattern Recognition, Springer LNCS 3175, pp. 415-423, T¨ubingen, Germany, Aug. 2004.

[3] Fatih Porikli, Oncel Tuzel and Peter Meer "Covariance Tracking using Model Update Based on Lie Algebra," in Proceedings of CVPR 2006.

[4] Babenko, B., Yang, M.-H., Belongie, S.: Robust object tracking with online multiple instance learning. PAMI 33, 1619–1632 (2011).

[5] Avidan, S.: Support vector tracking. PAMI 26, 1064–1072 (2004).

[6] Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. PAMI 27, 1631–1643 (2005).

[7] Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via online boosting. In: BMVC, pp. 47–56 (2006).

[8] Ross, D., Lim, J., Lin, R., Yang, M.-H.: Incremental learning for robust visual tracking. IJCV 77, 125–141 (2008).

[9] Hanxi Li, Chunhua Shen, and Qinfeng Shi "Real-time Visual Tracking Using Sparse Representation," In 2010.

[10] Donoho, D.: Compressed sensing. IEEE Trans. Inform. Theory 52, 1289–1306 (2006).

[11] Candes, E., Tao, T.: Near optimal signal recovery from random projections and universal encoding strategies. IEEE Trans. Inform. Theory 52, 5406–5425 (2006).

[12] Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. PAMI 31, 210–227 (2009)

[13] Candes, E., Tao, T.: Decoding by linear programing. IEEE Trans. Inform. Theory 51, 4203–4215 (2005)

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in CVPR, vol. 1, 2001, pp. 511–518.

[15] P. Doll´ar, Z. Tu, H. Tao, and S. Belongie, "Feature mining for image classification," in CVPR, 2007.

[16] J. Wang, X. Chen, and W. Gao, "Online selecting discriminative tracking features using particle filter," in CVPR, vol. 2, 2005, pp. 1037–1042.

[17] Z. Khan, T. Balch, and F. Dellaert, "A rao-blackwellized particle filter for eigentracking," in CVPR, vol. 2, 2004.

[18] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in NIPS, 2005, pp. 1417–1426.

[19] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in NIPS, 2003, pp. 577– 584.

[20] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," The Annals of Statistics, vol. 29, no. 5, pp. 1189– 1232, 2001.

[21] B. Babenko, P. Doll´ar, Z. Tu, and S. Belongie, "Simultaneous Learning and Alignment: Multi-Instance and Multi-Pose Learning," in Faces in Real-Life Images, 2008.

[22] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic a statistical view of boosting," The Annals of Statistics, vol. 28, no. 2, pp. 337–407, 2000.

[23] S. Stalder, H. Grabner, and L. van Gool, "Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition," in Online Learning in Computer Vision (OLCV) Workshop, 2009.

[24] Achlioptas, D, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins". J. Comput. Syst. Sci 66, 671–687 (2003).

