REAL-TIME OBJECT TRACKING VIA OPTIMAL FEATURE SUBSPACE

Xu Min[†,⋆] *Yu Zhou*[†] *Shu Liu*[‡] *Xiang Bai*[†]

[†]Department of Electronics and Information Engineering, [‡]School of Software Engineering,
Huazhong University of Science and Technology, Wuhan, China,
[⋆] Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
{minxueric, zhouyu.hust, liushuhust, xiang.bai}@gmail.com

ABSTRACT

In this paper, we present a real-time tracking approach based on the *Optimal Feature Subspace* (OFS). OFS is an optimal subspace of a random feature space, which can best represent the target and making it most distinguished in the whole scene. Initially, we randomly crop patches inside the bounding box to generate an efficient feature template set. Then a greedy algorithm fusing the cues of both target and background is proposed to seek the OFS at every frame. In the forthcoming frame, considering the correlation of different dimensions, we compute the *Mahalanobis distance* of candidate patches to the appearance model in the obtained subspace to locate the target. The experimental results on several challenging video clips demonstrate that our approach outperforms the state-of-the-art methods, in terms of both speed and robustness.

***Index Terms*—** Real-time object tracking, Bayesian inference, Optimal Feature Subspace

1. INTRODUCTION

Visual tracking has long been one of the most important problems in computer vision. It is a challenging problem due to many factors including occlusion, motion blur, pose change, illumination variation and background clutter.

Current tracking algorithms can be categorized into generative approaches and discriminative ones. Discriminative methods treat tracking as a classification problem which distinguishes the target from the background [1][2][3][4]. They can combine information of both the target and background. Grabner [3] uses a boosting algorithm to achieve discriminative features and in [4] another similar approach in a semi-online manner is used to handle the drifting problem. Barbenko *et al.*[5] propose a multiple instance learning algorithm which can handle ambiguities in the training data. Generative approaches focus on measurement of the distance from the search regions to the target model [6][7][8][9][10][11][12]. The IVT method [13] utilizes an incremental subspace model to cope with appearance variation. Kwon *et al.*[6] decompose the observation model into multiple basic observation models to cover a wide range of pose and illumination change. Li *et al.*[14] use sparse representation to formulate ℓ_1-tracker which is solved by the orthogonal matching pursuit algorithm.

Different from the above methods, we strive to seek an *Optimal Feature Subspace* (OFS) to represent the target in a specific scenario. To begin with, based on *Haar*-like feature, we generate a sufficient feature template set through random sampling method, although many feature templates in this set hold no effectiveness for tracking. The philosophy of our approach is essentially to make full use of both positive and negative features. Specifically, we apply the feature template set in the target patch and some other image patches far away from it, and then we get one positive feature and some negative features. By measuring the standardized distance between the positive and negative features, those feature dimensions which have strong ability to separate the target from the background can be inferred. The target can be then represented in the OFS most distinctively and accurately. Moreover, our appearance model changes gradually in pace with the target by modifying the OFS moderately at every new frame, such that our tracker can handle problems like occlusions. Eventually, the process of predicting the location of target in the next frame is simply matching the candidate to the appearance model via *Mahalanobis distance* in OFS.

In general, the main contributions of the proposed tracking approach based on OFS are:

- Our tracker adaptively compresses original target image into a low-dimension feature subspace with good property of image representation.
- Owing to the random sampling strategy, our feature template set can draw a coarse-to-fine global description of the target, which makes our OFS adaptive to a wide range of challenges.
- The low computational complexity of solving the OFS guarantees our tracker is a real-time system.

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 61222308, 61173120; by the Program for New Century Excellent Talents in University in China under Grant NCET-12-0217; by the Fundamental Research Funds for the Central Universities under Grant HUST.

The remainder of this paper is organized as follows: in Sec.2, we formulate the tracking problem; in Sec.3 the novel tracking method is proposed; in Sec.4, we evaluate our system on some challenging videos for object tracking. We draw conclusion in Sec.5.

2. PROBLEM FORMULATION

In this section, we interpret visual tracking problem as a Bayesian inference problem in a Markov model with hidden state variables. Given an observed frame set up to the t-th frame $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t\}$, the hidden state variable $\mathbf{x}_t$ can be estimated recursively,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \tag{1}$$

where $p(\mathbf{y}_t|\mathbf{x}_t)$ is the appearance model that estimates the likelihood of observation $\mathbf{y}_t$ predicted by hidden state $\mathbf{x}_t$, while $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the motion model giving prior fact on relations of states between two consecutive frames. Hidden state $\mathbf{x}$, in our work, indicates the (x, y) coordinates of the patch center. Practically, we sample abundant candidate image patches to match with the target. Hence, the optimal state $\hat{\mathbf{x}}_t$ of target at time t can be obtained by the Maximum A Posterior estimation:

$$\begin{aligned} \hat{\mathbf{x}}_t &= \arg\max_{\mathbf{x}_t^i} p(\mathbf{x}_t^i|\mathbf{y}_{1:t}) \\ &= \arg\max_{\mathbf{x}_t^i} p(\mathbf{y}_t|\mathbf{x}_t^i) p(\mathbf{x}_t^i|\mathbf{x}_{t-1}) \end{aligned} \tag{2}$$

where $\mathbf{x}_t^i$ denotes the state $\mathbf{x}_t$ of i-th sample.

We establish the motion model based on *inertia* assumption. In reality, every object has tendency to maintain its motion state. Consequently, at time t, the location of the target is expected to be the previous location plus the last displacement, and we assume it to be Gaussian distributed:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = N(\mathbf{x}_t; \mu, \Sigma) \tag{3}$$

where $\mu = \mathbf{x}_{t-1} + (\mathbf{x}_{t-1} - \mathbf{x}_{t-2})$ is the expected location. $\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$ is a diagonal covariance matrix, where σ is the standard deviation for x, y.

The other subproblem is the appearance model, and it plays a key role in our approach, which will be discussed in detail in Sec.3. In brief, we attempt to represent the target by learning an *Optimal Feature Subspace* $\hat{\mathbf{f}}$. Based on this feature representation, we compute *Mahalanobis distance* $D_M(\mathbf{x}_t^i)$ from a sampled candidate $\mathbf{x}_t^i$ to the target. Therefore, the appearance likelihood is formulated as:

$$p(\mathbf{y}_t|\mathbf{x}_t^i) \propto \exp(-D_M(\mathbf{x}_t^i)) \tag{4}$$

3. APPEARANCE MODEL

In this section, we are aimed at seeking an OFS for the appearance model. Our approach is based on feature template sampling and feature selection.

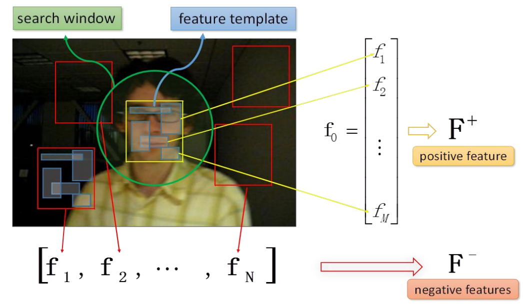

Fig. 1. The overview of generating feature template set and obtaining positive and negative features.

3.1. Feature Template Sampling

The primary consideration is that an OFS is feasible on condition that we get an over-complete feature space. Meanwhile, it's obvious that the original image patch has both auxiliary and redundant information for tracking. Based on the above knowledge, we adopt random sampling strategy to generate a qualified feature template set.

Formally, we randomly crop M patches inside the bounding box, where M is large enough to make sure the feature template set is over-complete. Every feature template corresponds to a Haar-like feature value f. Here we choose Haar-like feature [15], since it is invariant to moderate rotation and scale changes. In addition, benefiting from the *Integral Image*, the calculation speed of the feature can be accelerated remarkably. In this way, we establish a feature template set, i.e., an over-complete feature space $\mathbf{f} = [f_1, f_2, ..., f_M]^\top$, where $\mathbf{f} \in \mathbb{R}^{M\times 1}$.

At every time step t, our tracker crops out a group of N image patches $\mathbf{L}^r = \{l | \|l - \mathbf{x}\|_2 > r\}$ that are outside some radius r of the current tracker location, where l denotes the location of a image patch which has the same size as the bounding box. On one hand, we apply the feature templates on each image patch in the negative bag $\mathbf{L}^r$ to obtain the negative feature pool $\mathbf{F}^- = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N]$. On the other hand, our positive feature pool is simply the feature vector of the target image $\mathbf{F}^+ = [\mathbf{f}_0]$. Here $\mathbf{F}^+ \in \mathbb{R}^{M\times 1}$, and $\mathbf{F}^- \in \mathbb{R}^{M\times N}$.

The whole process is visualized in Fig. 1. The yellow box indicates the location of the target. Outside the green search window are the randomly sampled image patches which are represented by red boxes. The blue rectangles are also randomly generated which constitute our feature template set. Consequently, there is only one positive feature and N negative features.

3.2. Optimal feature subspace

Intuitively, a target owns visual uniqueness in the whole scene, and can be separated from the background in a low-dimension feature subspace. Furthermore, this randomized feature template set may contain those mis-aligned patches which will degrade our appearance model. Therefore, we propose to extract a sparse feature vector $\hat{\mathbf{f}}$, i.e., an OFS, from the aforementioned feature space, which is illustrated in

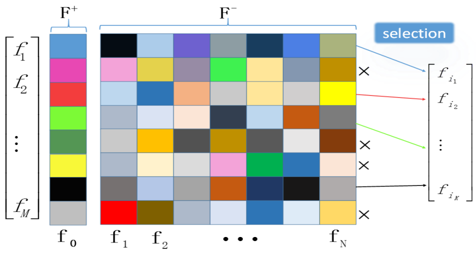

Fig. 2. Graphic representation of feature selection. In the matrix, different colors illustrate different values.

Fig.2.

Our goal is to find the optimal K-dimension feature subspace which is proper to represent the target, hence the optimization problem is formulated as:

$$\max_{\hat{\mathbf{f}} \subseteq \mathbf{f}} D(\mathbf{F}^+, \mathbf{F}^- | \hat{\mathbf{f}}). \tag{5}$$

The objective function $D(\mathbf{F}^+, \mathbf{F}^- | \hat{\mathbf{f}})$ denotes the distance between the feature pool $\mathbf{F}^-$ and $\mathbf{F}^+$ in the feature subspace $\hat{\mathbf{f}}$.

In order to solve this problem, we propose a greedy algorithm. The subspace $\hat{\mathbf{f}}$ corresponds to an index set $\Omega = \{i_1, i_2, ..., i_K\}$ which is a subset of $\{1, 2, ..., M\}$, thus the distance can be formulated as bellow:

$$D(\mathbf{F}^+, \mathbf{F}^- | \hat{\mathbf{f}}) = \sum_{i \in \Omega} D_i(\mathbf{F}^+, \mathbf{F}^-). \tag{6}$$

Actually, the criteria to select the *i*-th feature dimension is supposed to be the distance $D_i(\mathbf{F}^+, \mathbf{F}^-)$ from the negative feature pool to the positive pool in this dimension. Before computing the distance, we have to know the covariance $\mathbf{S}$ of all the dimensions:

$$\mathbf{S} = E[(\mathbf{f} - E[\mathbf{f}])(\mathbf{f} - E[\mathbf{f}])^\top], \tag{7}$$

where $E[\mathbf{f}] = \sum_{j=0}^{N} \mathbf{f}_j / (N+1)$. The covariance matrix $\mathbf{S} \in \mathbb{R}^{M \times M}$ will ont only be used in data preprocessing, but also make analysis of the correlation among different feature dimensions, which will be discussed at length in Sec.3.3. After that, the Euclidean distance between the *i*-th feature of the *j*-th negative sample patch f_{ij} and the expected positive feature f_{i0} will be standardized by:

$$D_{ij} = \frac{\| f_{ij} - f_{i0} \|}{S_{ii}} \quad \forall i \in 1, 2, ..., M, \forall j \in 1, 2, ..., N \tag{8}$$

where $\|\cdot\|$ denotes ℓ_1-norm, and S_{ii} indicates the *i*-th diagonal element of $\mathbf{S}$, i.e., the variance of the data set in *i*-th dimension. In this way, we can define the distance between positive feature pool and the negative one in the *i*-th dimension by:

$$D_i(\mathbf{F}^+, \mathbf{F}^-) = \min_j (D_{ij}), \tag{9}$$

since the minimal distance exactly reflects the discriminative capacity of one feature dimension. Based on Eqs.(6)-(9), our algorithm is summarized in Alg.1 ($D_i(\mathbf{F}^+, \mathbf{F}^-)$ is denoted as D_i for simplicity).

Algorithm 1: Greedy algorithm to optimize feature subspace

Input: Negative and positive feature pool $\mathbf{F}^-$, $\mathbf{F}^+$

1. Data preprocessing: calculate covariance matrix $\mathbf{S}$ and the standardized distance $D_{ij} = \frac{\| f_{ij} - f_{i0} \|}{S_{ii}}$
2. Distance computation: for the *i*-th dimension of feature space, $D_i(\mathbf{F}^+, \mathbf{F}^-) = \min_j (D_{ij})$
3. Distance ranking: rank the distances in descending order $D_{i_1} > D_{i_2} > \cdots > D_{i_k} > \cdots > D_{i_M}$, where $\{i_1, i_2, ..., i_M\}$ is a permutation of $\{1, 2, ..., M\}$
4. Feature selection: choose the first K dimensions to construct the optimal subspace $\hat{\mathbf{f}} = [f_{i_1}, f_{i_2}, ..., f_{i_K}]^\top$, where $\hat{\mathbf{f}} \in \mathbb{R}^{K \times 1}$

Output: *Optimal Feature Subspace* $\hat{\mathbf{f}}$

3.3. Likelihood computing

With the learned feature template $\hat{\mathbf{f}}$, the rest of the problem is readily solved. The likelihood $p(\mathbf{y}_t | \mathbf{x}_t^i)$ can be measured by the *Mahalanobis distance* $D_M(\mathbf{x}_t^i)$ between the feature $\hat{\mathbf{f}}(\mathbf{x}_t^i)$ of the candidate patch $\mathbf{x}_t^i$ in the search window and our appearance model $\hat{\mathbf{f}}_t$ at time step *t*:

$$D_M(\mathbf{x}_t^i) = \sqrt{(\hat{\mathbf{f}}(\mathbf{x}_t^i) - \hat{\mathbf{f}}_t)^\top \hat{\mathbf{S}}^{-1} (\hat{\mathbf{f}}(\mathbf{x}_t^i) - \hat{\mathbf{f}}_t)}, \tag{10}$$

where the covariance matrix $\hat{\mathbf{S}} \in \mathbb{R}^{K \times K}$ can be easily obtained by extracting the $\{i_1, i_2, ..., i_K\}$ columns and rows of $\mathbf{S}$. *Mahalanobis distance* makes our computation of likelihood much more credible, as it takes into consideration the correlation of the different dimensions in the feature subspace $\hat{\mathbf{f}}$ (e.g., two overlapped template patches in Fig.1).

3.4. Update of appearance model

Tracking with the fixed templates will be prone to failure in cases where there are illumination change, occlusion, pose variation, etc. Numerous methods have been designed to prevent the tracker drifting away from the target. Ross *et al.* [13] extend the sequential Karhunen-Loeve algorithm and propose an incremental PCA algorithm to update the appearance. Mei and Ling [16][17] apply sparse representation in tracker to handle outliers and partial occlusion. In our paper, we update our appearance model in a relatively simple and effective way. At time step $t+1$, some dimensions of optimal feature subspace $\hat{\mathbf{f}}_t$ will be removed and an equal number of new features will be added. In detail, the randomly chosen νK dimensions in $\hat{\mathbf{f}}_t$ will be maintained, and the rest $(1-\nu)K$ entries will be chosen from the $M - \nu K$ feature dimensions left in the feature template set, where ν denotes the forgetting ratio. The way of selecting the new $(1-\nu)K$ feature dimensions is pretty similar to Alg.1.

3.5. Computational complexity

One advantage of tracking via OFS is low computational complexity. In feature sampling, the positive feature pool

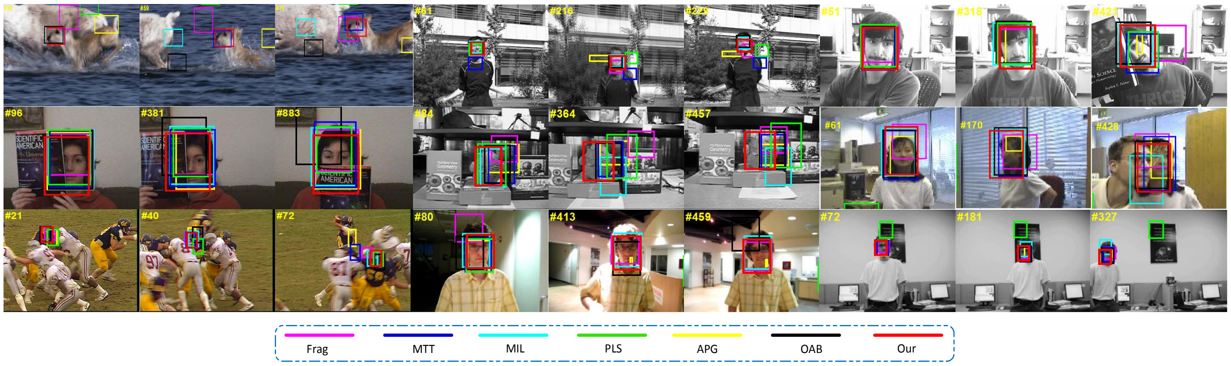

Fig. 3. Comparison with other state-of-the-art tracking methods

$\mathbf{F}^+$ is a M-dimensional vector, while the negative feature pool $\mathbf{F}^-$ is comprised of MN elements. Moreover, feature selection is just a sorting performed on M standardized Euclidean distances. In the likelihood computing, assuming that there are P candidate patches to choose from, the time cost is proportional to patch number P times feature dimension K. Therefore, the total computational complexity is only $\Theta(MN + M\log(M) + PK)$.

4. EXPERIMENTS

We perform our experiments on nine publicly available video sequences, and the authors label the ground truth center of the object for every frame. These videos provide a wide range of significant challenges including occlusion, motion blur, scale variation, rotation, pose variation and cluttered background. The tracking results are summarized in Table 1 and Fig. 3, and our tracker is gauged against six other state-of-the-art trackers with the same initial position of the target. These trackers include PLS [18], MTT[19], APG[20], OAB[3], Frag[21], MIL[15].

Our tracker is implemented in MATLAB and runs at 20 frames per second on a 2.53GHz Intel Core i3 CPU with 2GB memory. We choose $K = 100$ features from $M = 1000$ randomized features. The forgetting ratio in the update module is set to be $\nu = 0.7$. The size of search window depends on the specific video. In general, the robustness of our tracker is insensitive to parameters, since our optimal feature subspace learned from a randomly generated template set gives a coarse-to-fine description of the target.

Quantitatively, we adopt the *Average Center Location Error*(ACLE), which measures the distance between the tracked and the ground truth center locations, to evaluate a tracker's performance. In most of the nine tested videos, our tracker achieves the best performance, while in *David2* and *Girl*, our tracker also ranks top three. In terms of average ACLE among all the videos, our tracker outperforms all the other methods, which evidently shows that our approach is robust to different tough situations.

Qualitatively, Fig.3 illustrates vividly that our tracker can handle challenges better. Results on sequence *FaceOcc1*, *FaceOcc2* and *Football1* show optimal feature subspace makes our appearance model adaptive to occlusion, since patches on the occluded regions will be omitted automatically. *Deer* and *Jumping* prove that motion blur is within our tacker's capacity. This owes to Haar-like feature template set which can provide an overview characteristics of the target without paying excessive attention on meaningless details. In addition, *David* and *David2* demonstrate that our tracker copes well with situations where there are subtle pose variation and scale change, even though the scale of the bounding box in our tracker is fixed. Finally, it's revealed in *Fish* and *Girl* that our method is capable of handling appearance and illumination change, owing that the optimal feature subspace is updated in a reasonable manner.

Table 1. *Average Center Location Error* measured in pixels. Red color indicates best performance, while blue indicates second best.

Video	**Our**	APG	Frag	MIL	MTT	OAB	PLS
Deer	9.97	237.79	86.19	227.13	11.08	216.61	153.91
Jumping	4.63	34.88	17.66	7.32	63.78	7.49	42.51
Football1	5.13	35.42	22.37	8.18	21.99	29.92	24.08
David	8.30	37.42	53.88	10.74	133.19	32.28	90.22
David2	2.15	4.81	6.71	26.56	1.30	4.89	71.74
FaceOcc1	11.34	12.84	39.09	31.42	22.52	60.48	34.96
FaceOcc2	6.56	15.65	33.49	14.08	13.83	16.18	14.26
Fish	11.21	38.23	43.56	37.25	29.98	27.39	41.78
Girl	10.30	4.47	13.37	14.72	8.91	10.65	83.74
Average	7.73	46.83	35.15	41.93	34.06	45.11	61.91

5. CONCLUSION

In this paper, we propose a real-time and effective tracking algorithm based on *Optimal Feature Subspace*. Random sampling strategy guarantees the completeness of the feature space, since sampled templates are in various scales and shapes. Eliminating redundancy, the optimal feature subspace is proper and effective to represent the target. Furthermore, *Mahalanobis distance* takes correlations of different dimensions into consideration. In summary, our model performs well in terms of robustness and speed.

6. REFERENCES

[1] S. Avidan., "Support vector tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[2] S. Avidan., "Ensemble tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[3] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *British Machine Vision Conference(BMVC)*, 2006, pp. 47–56.

[4] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. European Conf. on Computer Vision(ECCV)*, 2008, pp. 234–247.

[5] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[6] Junseok Kwon and Kyoung Mu Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition(CVPR)*, 2010.

[7] Ido Leichter., "Mean shift trackers with cross-bin metrics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 695–706, 2012.

[8] X. Mei and H. Ling., "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.

[9] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang, "Incremental tensor subspace learning and its applications to foreground segmentation and tracking.," *Int'l J. Computer Vision*, vol. 91, no. 3, pp. 303–327, 2010.

[10] Jialue Fan, Ying Wu, and Shengyang Dai, "Discriminative spatial attention for robust tracking," in *Proc. European Conf. on Computer Vision(ECCV)*, 2010.

[11] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang, "Online object tracking with sparse prototypes," *IEEE Trans. on Image Processing*, vol. 22, no. 1, pp. 314–325, 2013.

[12] Yu Zhou, Xiang Bai, Wenyu Liu, and Longin Jan Latecki, "Fusion with diffusion for robust visual tracking," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[13] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[14] Hanxi Li, Chunhua Shen, and Qinfeng Shi, "Real-time visual tracking using compressive sensing," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1305–1312.

[15] B. Babenko, M.H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[16] Xue Mei and Haibin Ling, "Robust visual tracking using ℓ_1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1436–1443.

[17] Xue Mei, Haibin Ling, Yi Wu, Erik Blasch, and Li Bai, "Minimum error bounded efficient ℓ_1 tracker with occlusion detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1257–1264.

[18] Qing Wang, Feng Chen, Wenli Xu, and Ming-Hsuan Yang, "Object tracking via partial least squares analysis," *IEEE Trans. on Image Processing*, vol. 21, no. 10, pp. 4454–4465, 2012.

[19] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja, "Robust visual tracking via structured multi-task sparse learning.," *Int'l J. Computer Vision*, vol. 101, no. 2, pp. 367–383, 2013.

[20] C. Bao, Y. Wu, H. Ling, , and H. Ji, "Real time robust ℓ_1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[21] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragment-based tracking using the integral histogram," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition(CVPR)*, 2006, pp. 798–805.