LEARNING A TEMPORALLY INVARIANT REPRESENTATION FOR VISUAL TRACKING

Chao Ma*†*, Xiaokang Yang***, Chongyang Zhang***, and Ming-Hsuan Yang*†

*Shanghai Jiao Tong University, China
†University of California at Merced, USA

ABSTRACT

In this paper, we propose to learn temporally invariant features from a large number of image sequences to represent objects for visual tracking. These features are trained on a convolutional neural network with temporal invariance constraints and robust to diverse motion transformations. We employ linear correlation filters to encode the appearance templates of targets and perform the tracking task by searching for the maximum responses at each frame. The learned filters are updated online and adapt to significant appearance changes during tracking. Extensive experimental results on challenging sequences show that the proposed algorithm performs favorably against state-of-the-art methods in terms of efficiency, accuracy, and robustness.

Index Terms— temporal invariance, feature learning, correlation filters, object tracking

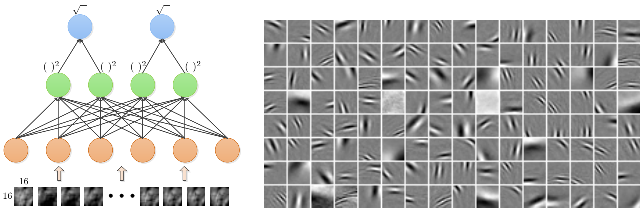
1. INTRODUCTION

Visual tracking is one of the most fundamental problems in computer vision with numerous applications [1, 2]. A typical scenario is to track an arbitrary object initialized by a bounding box in subsequent image frames. In this paper, we aim at learning an invariant compact representation from large-scale image sequences to address challenging issues of visual tracking where target objects undergo significant appearance changes due to occlusions, deformations, motion blurs, abrupt motions, illumination variations, and cluttered backgrounds.

Feature representation is of prime importance in visual object tracking with the goal of discriminating the target from the background context and has received considerable attention in the literature. In [3], discriminative local patches are selected to compute the target displacement using the Lucas-Kanade method [4]. Similarly, Collins et al. [5] propose an online ranking mechanism for feature selection by measuring the variance ratio between object and background pixels. In [6], Adam et al. propose to represent target objects using multiple random fragments. The compressive tracker [7] employs random projections to extract data-independent features as the appearance model. Grabner et al. [8] use key points to describe the regions containing targets and surrounding context. Several hand-crafted local descriptors, such as SIFT [9], SURF [10] and ORB [11], have also been exploited as target representation.

Recently, learning features from raw image pixels on large-scale dataset to deal with computer vision problems has made impressive progress compared with hand-crafted features [9, 12]. Wang and Yeung [13] propose a deep learning tracking (DLT) method to learn compact features from generic natural images to augment tracking performance. However, the DLT method merely uses the still natural images [14] for training and takes no account of the temporal slowness of target appearance in adjacent frames. Moreover, the DLT method transfers the offline learned feature encoder to initialize a neural network (NN) classifier in the first frame. The feature invariance is therefore disregarded when the NN classifier is updated on-the-fly to adapt to the target appearance changes. To address these issues, we propose to learn compact features from auxiliary large-scale video sequences as target representation. The features are learned with a temporal invariance constraint and able to handle a wide range of motion patterns in challenging testing sequences. We further take into account the correlation of appearance change between consecutive frames and develop a linear ridge regression model using correlation filters to encode the appearance template based on the learned invariant features. Since the correlation operator is readily transfered into the Fourier domain as element-wise multiplication, the proposed method effectively reduces the computational load and achieves an average tracking speed close to real-time.

We briefly discuss existing correlation tracking methods closely related to this work. Bolme et al. propose to learn a minimum output sum of squared error (MOSSE) [15] filter for visual tracking on gray-scale images, where the target appearance is represented by illumination intensities. Heriques et al. [16] propose to use correlation filters in a kernel space (CSK) which achieves the highest speed in a recent benchmark [17]. The CSK method builds on illumination intensity features and is further improved in [18] by using HOG features, i.e., the KCF tracker. In [19], Danelljan et al. exploit color attributes to represent target objects and learn an adaptive correlation filter by mapping multi-channel features into a Gaussian kernel space. Recently, Zhang et al. [20] incorporate context illumination features into filter learning and model the scale change based on consecutive correlation responses. The



(a) (b)

Fig. 1. (a) Neural network architecture with square root subspace space pooling. The input training data are small patches of size 16 × 16 pixels with temporal slowness. (b) Learned filters with the main half of components.

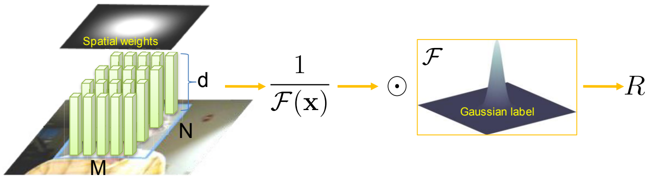


Fig. 2. The linear regression model R learned from a single frame. The feature vector $\mathbf{x}$ (with an additional layer of spatial weights) of target appearance is generated from the products between the sub-sampled patches and the learned filters in Fig. 1(b) . The operator $\mathcal{F}$ denotes the FFT transformation and $\odot$ is the Hadamard product.

DSST tracker [21] learns adaptive multi-scale correlation filters using HOG features as target representation to handle the problem of scale change. Note that our proposed algorithm differs significantly from existing methods based on correlation filters as our model builds on a temporally invariant representation learned from a large-scale dataset rather than hand-tuned features. Extensive experimental validations on 20 challenging video sequences show that the proposed algorithm performs favorably against state-of-the-art tracking methods in terms of efficiency and effectiveness.

2. PROPOSED ALGORITHM

As we aim to learn temporally invariant features resistant to appearance deformation, we first describe the architecture of the neural network for feature learning; then present the linear correlation model based on the learned representation and discuss the update scheme to adapt to appearance change during tracking.

Temporally Invariant Feature Learning. In this work, we employ an unsupervised single-layer neural network trained on Hans van Hateren's natural video repository as used in [22] to handle diverse motion transformations of objects in visual tracking. Given N training frames indexed by t, the hidden features $\mathbf{x}_t$ are learned from input data (image patches) $\mathbf{d}_t$ by solving the following unconstrained minimization problem,

$$\min_W \lambda \sum_{t=1}^{N-1} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_1 + \sum_{t=1}^{N} \|\mathbf{d}_t - W^\top W \mathbf{d}_t\|_2^2 + \lambda' \sum_{t=1}^{N} \|\mathbf{x}_t\|_1, \tag{1}$$

where W denotes the learned coefficients that connect the neurons $\mathbf{d}_t$ and $\mathbf{x}_t$; λ and λ' are regularization parameters. The hidden features $\mathbf{x}_t$ are mapped from data $\mathbf{d}_t$ by a feed-forward pass in the network as $\mathbf{x}_t = \sqrt{H(W\mathbf{d}_t)^2}$, where H denotes the square root pooling on the linear network (see [23] for details). The architecture of the network with the pooling layer is showed in Fig 1(a). In (1), the first term enforces a temporal invariance constraint on the learned filters W; the second term denotes the auto-encoder reconstruction cost [24]; and the third is the L_1 norm regularization to ensure that the obtained features have sparse activations. Note that our proposed features are more robust to diverse of motion patterns as we take the temporal slowness into account. The auto-encoder feature learning used in the DLT method is a special case of our objective function (1) when temporal invariance is disregard (i.e., $\lambda = 0$).

In our implementation, the input training data are raw pixel values of small patches of size 16 × 16 pixels, which are cropped from consecutive frames using template matching [23] to retain temporal slowness (See Fig. 1(a)). In order to reduce the computational load, we perform principal component analysis (PCA) and only keep one half of the learned filters with the main components. Fig. 1(b) shows that the learned filters bears resemblance to edge detectors. Unlike the way the DLT method transfers the filters to a neural network classifier and retrains the feature representation along with model update, we learn the filters offline and leave them unchanged during the tracking process. As a result, the proposed features are data-independent and invariant to significant appearance deformation.

Linear Correlation Tracking. We integrate the learned features into a linear correlation model similar to [15, 20, 21]. For computational efficiency, we do no perform convolution directly on an image patch representing a target using the learned filters in Fig. 1(b). Instead, we use a sub-sampling strategy with a step l to extract $M \times N$ patches of size 16×16 and compute the elementwise product between these patches and the learned filters as feature representation $\mathbf{x}$. Therefore, each feature vector of target appearance has $M \times N \times d$ dimensions (e.g., $d = 128$ in this work). A correlation filter $\mathbf{w}$ (with the same dimensions as $\mathbf{x}$) encodes the target appearance and is trained from all the circular shifts of $\mathbf{x}$ along dimensions M and N. Each shifted sample $\mathbf{x}_{m,n}$, $(m,n) \in \{0, 1, \ldots, M-1\} \times \{0, 1, \ldots, N-1\}$, is assigned a Gaussian function label $y_{m,n}$ and the filter $\mathbf{w}$ is leaned as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{m,n} \|\mathbf{w} \cdot \mathbf{x}_{m,n} - y_{m,n}\|^2 + \gamma \|\mathbf{w}\|^2, \tag{2}$$

where the regularization parameter γ is subject to $\gamma \geq 0$ and the inner product $\cdot$ is induced by a linear kernel [16] in the Hilbert space, e.g., $\mathbf{w} \cdot \mathbf{x}_{m,n} = \sum_d \mathbf{w}_{m,n,d}^\top \mathbf{x}_{m,n,d}$. Since

Table 1. Comparison of distance precision rate (%) with a threshold of 20 pixels. The first and second best results are highlighted by bold and underline.

Sequence	Ours	DLT [13]	CSK [16]	STC [20]	KCF [18]	MIL [26]	Struck [27]	CT [7]	TLD [28]	SCM [29]	TGPR [30]
basketball	**100**	88.4	**100**	56.0	92.3	28.4	12.0	29.9	2.8	66.1	99.3
car4	98.9	**100**	35.5	96.7	95.3	35.4	99.2	28.1	87.4	97.4	**100**
carDark	**100**	71.0	**100**	**100**	**100**	37.9	**100**	**100**	63.9	**100**	**100**
carScale	80.8	72.2	65.1	64.7	80.6	62.7	64.7	71.8	**85.3**	64.7	80.6
crossing	**100**	**100**	**100**	53.3	**100**	**100**	**100**	**100**	61.7	**100**	95.0
david	**100**	92.6	49.9	83.7	**100**	69.9	32.9	81.5	**100**	**100**	98.7
david3	**100**	32.1	65.9	92.5	**100**	73.8	33.7	41.3	11.1	49.6	**100**
deer	**100**	38.0	**100**	4.2	81.7	12.7	**100**	4.2	73.2	2.8	**100**
faceocc1	40.7	53.3	**94.7**	25.0	73.0	22.1	57.5	33.0	20.3	93.3	83.1
faceocc2	92.5	83.9	**100**	97.4	97.2	74.0	**100**	68.1	85.6	86.0	97.9
fish	**100**	46.8	4.2	**100**	**100**	38.7	**100**	88.2	**100**	86.3	**100**
fleetface	55.3	40.3	56.7	48.1	46.0	35.8	**63.9**	43.8	50.6	52.9	39.3
football	81.8	30.4	79.8	80.1	79.6	79.0	75.1	79.8	80.4	76.5	**100**
girl	86.0	77.8	55.4	59.4	86.4	71.4	**100**	60.8	91.8	**100**	90.4
jumping	93.3	38.3	5.1	5.4	34.2	99.7	**100**	9.6	**100**	15.3	10.9
mountainBike	**100**	88.6	**100**	**100**	**100**	66.7	92.1	17.5	25.9	96.9	**100**
suv	**98.0**	82.4	56.8	80.5	97.9	12.3	57.2	25.0	90.9	97.8	53.1
sylvester	84.8	83.9	91.0	89.7	84.3	65.1	**99.5**	90.1	94.9	94.6	94.6
trellis	99.5	34.6	81.0	73.8	**100**	23.0	87.7	38.7	52.9	87.3	98.1
woman	94.8	94.1	25.0	61.5	93.8	20.6	**100**	20.4	19.1	94.0	94.0
Average	**90.3**	67.4	68.3	68.6	87.1	51.5	78.8	46.6	64.9	78.1	86.7

Table 2. Comparison of overlap success rate (%) with a threshold of 0.5. The first and second best results are highlighted by bold and underline.

Sequence	Ours	DLT [13]	CSK [16]	STC [20]	KCF [18]	MIL [26]	Struck [27]	CT [7]	TLD [28]	SCM [29]	TGPR [30]
basketball	**99.0**	59.9	87.4	23.6	89.8	27.4	10.2	25.9	2.5	61.1	85.0
car4	39.0	**100**	27.6	22.5	36.7	27.6	39.8	27.5	79.2	97.3	39.8
carDark	93.4	67.9	99.2	99.7	72.3	17.8	**100**	0.3	52.9	99.7	**100**
carScale	44.8	**72.2**	44.8	52.8	44.4	44.8	43.3	44.8	43.7	65.1	42.1
crossing	95.0	99.2	31.7	17.5	92.5	98.3	94.2	98.3	51.7	**100**	80.8
david	62.2	82.0	23.6	58.4	62.2	22.9	23.6	42.7	**97.0**	91.3	77.1
david3	97.6	24.6	62.7	33.3	**99.2**	68.3	33.7	34.9	10.3	48.4	98.8
deer	**100**	36.6	**100**	4.2	81.7	12.7	**100**	4.2	73.2	2.8	**100**
faceocc1	65.7	92.8	**100**	24.3	**100**	76.5	**100**	85.4	83.4	**100**	98.2
faceocc2	97.9	71.9	**100**	98.0	99.6	93.6	**100**	74.4	82.9	87.4	99.3
fish	**100**	44.1	4.2	37.2	**100**	38.7	**100**	88.9	96.4	86.3	**100**
fleetface	67.8	52.8	67.6	46.3	66.9	53.7	66.6	63.8	56.7	**70.6**	61.0
football	66.3	29.6	65.7	61.9	68.2	73.8	66.0	78.5	41.2	58.6	**97.0**
girl	77.4	60.6	39.8	30.2	75.6	29.4	**98.0**	17.8	76.4	88.2	88.2
jumping	**92.0**	12.5	4.8	4.8	28.1	47.6	79.9	0.6	84.7	12.1	9.6
mountainBike	99.2	36.0	**100**	87.3	99.1	57.5	85.5	17.1	25.9	96.1	**100**
suv	**98.6**	82.5	57.5	51.3	98.5	13.0	57.5	23.1	83.9	98.4	53.5
sylvester	83.4	49.0	71.7	61.0	81.9	54.6	**92.9**	82.8	92.8	88.6	92.3
trellis	**89.1**	32.9	59.1	58.0	84.0	24.4	78.4	35.0	47.3	85.4	79.3
woman	86.6	86.4	24.5	25.8	93.6	18.8	**93.5**	15.9	16.6	85.8	**93.5**
Average	**82.7**	59.7	58.6	44.9	78.7	45.1	73.1	43.1	59.9	76.2	79.8

the label $y_{m,n}$ is not binary, the learned filter $\mathbf{w}$ contains the coefficients of a linear ridge regression [25] rather than a binary classifier. Using the fast Fourier transformation (FFT) to compute the correlation, this objective function is minimized as $\mathbf{w} = \sum_{m,n} \mathbf{a}_{m,n} \cdot \mathbf{x}_{m,n}$, and the coefficient $\mathbf{a}$ is defined by

$$A = \mathcal{F}(\mathbf{a}) = \frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\mathbf{x}_{m,n} \cdot \mathbf{x}) + \gamma}. \quad (3)$$

In (3), $\mathcal{F}$ denotes the FFT operator and $\mathbf{y} = \big\{y(m,n)|(m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}\big\}$. Fig. 2 shows the pipeline of extracting the temporally invariant features and learning the linear regression model (R) using correlation filters. The tracking task is carried out on an image patch with feature representation $\mathbf{z}$ in the new frame by computing the response map,

$$\hat{\mathbf{y}} = \mathcal{F}^{-1}\big(A \odot \mathcal{F}(\mathbf{z} \cdot \hat{\mathbf{x}})\big), \quad (4)$$

where $\hat{\mathbf{x}}$ denotes the learned target appearance model in previous frames and $\odot$ is the Hadamard product. Therefore, the new position of the target is detected by searching for the location of the maximal value of $\hat{\mathbf{y}}$.

Model Update. Since our proposed feature representation is data-independent, we update the linear regression model online to adapt to target appearance change during tracking. We adopt the update strategy similar to [15] as follows,

$$\begin{aligned} \hat{\mathbf{x}}^t &= (1-\alpha)\hat{\mathbf{x}}^{t-1} + \alpha \mathbf{x}^t, \\ \hat{A}^t &= (1-\alpha)\hat{A}^{t-1} + \alpha A^t, \end{aligned} \quad (5)$$

where t is the index of the current frame and α is the online learning rate. This update scheme is computationally efficient and exploits the temporal relationship over time to ensure that the correlation filter adapts to appearance deformation quickly.

3. EXPERIMENTAL VALIDATIONS

We evaluate the proposed method on 20 challenging sequences [17] with comparison to 10 state-of-the-art trackers, including the DLT [13], CSK [16], STC [20], KCF [18], MIL [26], Struck [27], CT [7], TLD [28], SCM [29] and TGPR [30] methods. The temporal invariance term λ in (1) is set to 50 and the sparsity term λ' is set to 150. The regularization term γ in (2) is set to 10^{-4}. The size of the search window is 1.8 times of that of the initial bounding box as surrounding context information is usually helpful to discriminate targets. The down-sample step l is set to 4. The kernel width of the Gaussian function label is set to $\sqrt{MN}/10$. The learning rate α in (5) is set to 0.015. The proposed tracking algorithm is implemented in Matlab on an Intel i7-4770 3.40 GHz CPU with 32 GB RAM. We keep the parameters unchanged on all the sequences and report that the average tracking speed is 10.6 frame per second. The source code and more experimental results are available at `https://sites.google.com/site/chaoma99/icip15tracking`.

We use two quantitative metrics for evaluation: (i) distance precision rate, which shows the percentage of frames whose center location error is within 20 pixels, and (ii) overlap success rate, which is the percentage of frames where the bounding box overlap surpasses 0.5. Table 1 and Table 2 show that our proposed method performs favorably against state-of-the-art trackers in terms of both distance precision and overlap success.

In addition, we compare the tracking results of our proposed algorithm with other four state-of-the-art trackers (DLT [13], KCF [18], STC [20], and Struck [27]) closely related to this work on 8 challenging sequences in Fig. 3. The DLT tracker learns the feature representation from large-scale

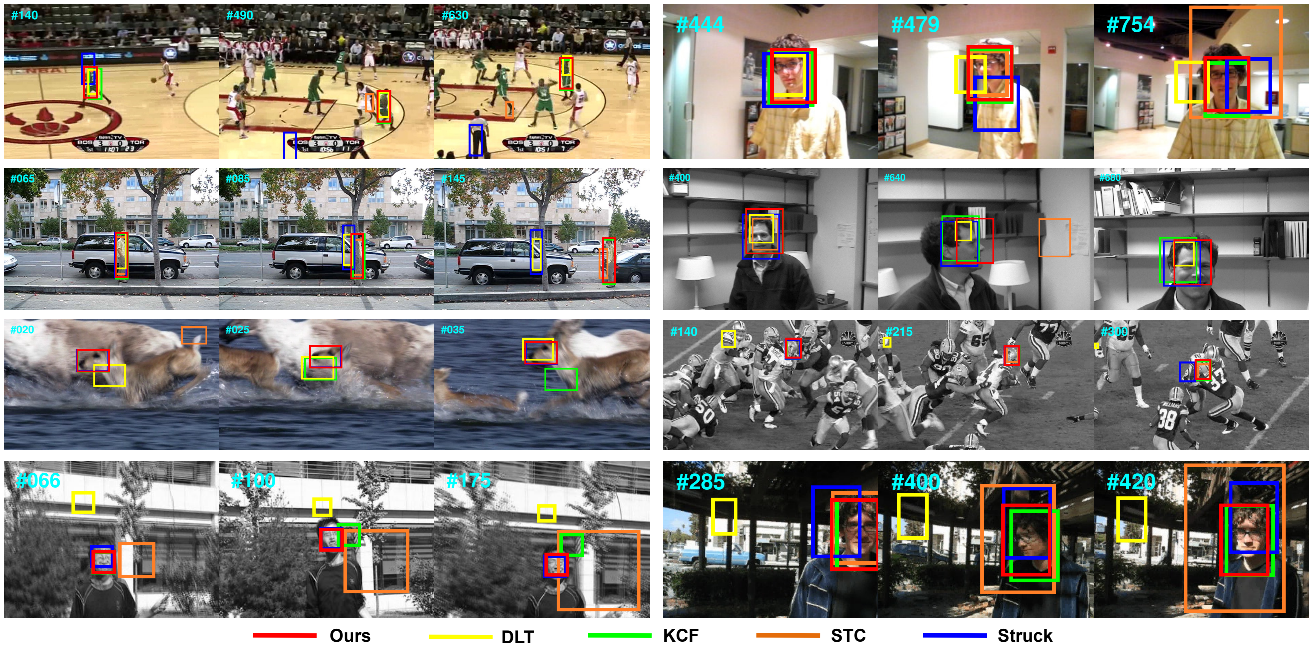


Fig. 3. Tracking results of our proposed algorithm, the DLT [13], KCF [18], STC [20], and Struck [27] methods on eight challenging sequences [17] (from left to right and top to down are *basketball*, *david*, *david3*, *fleetface*, *deer*, *football*, *jumping*, and *trellis*, respectively).

still images. Without temporal invariance, the DLT tracker drifts in the presence of rotation deformation (*david*, *trellis*), occlusion (*david3*), abrupt motion (*deer*, *football* and *jump*), and illumination change (*trellis*). The KCF tracker is based on a correlation filter learned from HOG features in Gaussian kernel space and performs well on the *basketball*, *david*, *david3*, and *trellis* sequences due to the effectiveness of the kernel correlation model. However, it fails to follow objects undergoing significant appearance change caused by abrupt motion (*deer* and *jumping*) as the hand-crafted HOG features are less effective to discriminate the targets from background. The STC tracker is also based on correlation filters and able to estimate scale changes, but does not perform well when both significant scale changes and rotation occur (*basketball*, *david*, *jumping* and *trellis*) or in the presence of abrupt motion (*deer*) as it only learns the filter from the brightness channel rather than temporally invariant features as we do. The Struck method fails to track objects undergoing rotation (*basketball* and *david*), background clutter (*trellis*), and heavy occlusion (*david3*) since it is less effective in handling significant appearance change using hand-crafted features (e.g., the Harr or HOG features).

Overall, our proposed tracker performs well against state-of-the-art methods on these challenging sequences, which can be attributed to two main reasons. First, target objects are represented by the temporally invariant features learned from large-scale video sequences rather than crafted by hand (e.g., the HOG features in the KCF tracker or simple brightness intensity in the STC tracker). The proposed feature representation is less sensitive to illumination and background clutter (*basketball* and *trellis*) and blurring caused by fast motion (*deer* and *jumping*). Second, our regression model is based on correlation filters, which consider the temporal relationship of the target and surrounding context between adjacent frames, and is updated sequentially to adapt to appearance change. Therefore, our method effectively maintains a trade-off between model stableness and adaptivity for visual tracking and is able to handle the challenges of significant rotation (*basketball*, *david* and *fleetface*) and severe occlusion (*david3*).

4. CONCLUSIONS

In this paper, we propose an effective algorithm for visual object tracking. Our method learns temporally invariant features from large-scale video sequences using a convolutional neural network. The learned features for representing target objects are robust to diverse of motion patterns and thus augment the tracking performance effectively and efficiently. We further model the temporal relationship between adjacent frames using a linear ridge regression based on correlation filters. Extensive experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods in terms of accuracy and robustness.

Acknowledgment. C. Ma is sponsored by China Scholarship Council. X. Yang and C. Zhang are supported in part by the NSFC Grants #61025005, #61129001 and #61221001, STCSM Grants #14XD1402100 and #13511504501, 111 Program Grant #B07022, and CNKT R&D Program Grant

#2012BAH07B01.

5. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[2] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” *TPAMI*, vol. 36, no. 7, pp. 1442–1468, 2014.

[3] J. Shi and C. Tomasi, “Good features to track,” in *CVPR*, 1994.

[4] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *IJCV*, vol. 56, no. 3, pp. 221–255, 2004.

[5] R. T. Collins, Y. Liu, and M. Leordeanu, “Online selection of discriminative tracking features,” *TPAMI*, vol. 27, no. 10, pp. 1631–1643, 2005.

[6] A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments-based tracking using the integral histogram,” in *CVPR*, 2006.

[7] K. Zhang, L. Zhang, and M.-H. Yang, “Real-time compressive tracking,” in *ECCV*, 2012.

[8] M. Grabner, H. Grabner, and H. Bischof, “Learning features for tracking,” in *CVPR*, 2007.

[9] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[10] D. Ta, W. Chen, N. Gelfand, and K. Pulli, “Surftrac: Efficient tracking and continuous object recognition using local feature descriptors,” in *CVPR*, 2009.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *ICCV*, 2011.

[12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, 2005.

[13] N. Wang and D. Yeung, “Learning a deep compact image representation for visual tracking,” in *NIPS*, 2013.

[14] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *TPAMI*, vol. 30, no. 11, pp. 1958–1970, 2008.

[15] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in *CVPR*, 2010.

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *ECCV*, 2012.

[17] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in *CVPR*, 2013.

[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *TPAMI*, 2015.

[19] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, “Adaptive color attributes for real-time visual tracking,” in *CVPR*, 2014.

[20] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, “Fast visual tracking via dense spatio-temporal context learning,” in *ECCV*, 2014.

[21] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, “Accurate scale estimation for robust visual tracking,” in *BMVC*, 2014.

[22] C. F. Cadieu and B. A. Olshausen, “Learning transformational invariants from natural movies,” in *NIPS*, 2009.

[23] W. Y. Zou, A. Y. Ng, S. Zhu, and K. Yu, “Deep learning of invariant features via simulated fixations in video,” in *NIPS*, 2012.

[24] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, “ICA with reconstruction cost for efficient overcomplete feature learning,” in *NIPS*, 2011.

[25] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, The MIT Press, 2012.

[26] B. Babenko, M.-H. Yang, and S. Belongie, “Robust object tracking with online multiple instance learning,” *TPAMI*, vol. 33, no. 8, 2011.

[27] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” in *ICCV*, 2011.

[28] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *TPAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[29] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparse collaborative appearance model,” *TIP*, vol. 23, no. 5, pp. 2356–2368, 2014.

[30] J. Gao, H. Ling, W. Hu, and J. Xing, “Transfer learning based visual tracking with gaussian processes regression,” in *ECCV*, 2014.