2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Non-negative Subspace Representation Learning Scheme for Correlation Filter Based Tracking

Tianyang Xu[1,2], Xiao-Jun Wu*[1], Josef Kittler[2]
1. School of IoT Engineering Jiangnan University, Wuxi, China, 214122
2. CVSSP, University of Surrey, GU2 7XH, Guildford, UK
Email: tianyang_xu@163.com, wu_xiaojun@jiangnan.edu.cn, j.kittler@surrey.ac.uk

***Abstract*—Discriminative correlation filter (DCF) based tracking methods have achieved great success recently. However, the temporal learning scheme in the current paradigm is of a linear recursion form determined by a fixed learning rate which can not adaptively feedback appearance variations. In this paper, we propose a unified non-negative subspace representation constrained leaning scheme for DCF. The subspace is constructed by several templates with auxiliary memory mechanisms. Then the current template is projected onto the subspace to find the non-negative representation and to determine the corresponding template weights. Our learning scheme enables efficient combination of correlation filter and subspace structure. The experimental results on OTB50 demonstrate the effectiveness of our learning formulation.**

I. Introduction

Visual object tracking is an important research direction in the field of computer vision and pattern recognition. It has been widely embedded in a variety of practical applications, such as intelligent surveillance, robot perception, medical image processing and other visual intelligence systems. Any target tracking algorithm is designed to find the corresponding target in continuous video frames automatically given the initial state information (position and size) of the target. Though many tracking methods have been proposed in the literature recently, target tracking still suffers from many practical problems, e.g., illumination changes, camera fuzzy, camera jitter, non-rigid deformation and partial occlusion[1][2].

Discriminative correlation filter (DCF) based tracking methods have achieved outstanding performance in recent object tracking benchmarks, i.e., OTB50, OTB100, VOT2014, VOT2015 and VOT2016[2][4][5][6][7]. The main advantage of DCFs lies in their use of circular structure of original samples and in formulating the tracking problem as ridge regression. Besides, Fast Fourier Transform (FFT) also provides a fast calculation of closed form solutions for all shift candidates in the frequency domain. However, conventional DCF based methods are limited to learn an unstable single multi-feature patch template, as shown in Fig.1, which is derived using a simple appearance representation and updating scheme. Though HOG descriptors and color features are integrated in the appearance models, existing DCF based methods lack the ability to represent and store the dynamic information of the target. On the other hand, subspace methods have been widely researched in visual object tracking. Under this framework, the time-varying targets are assumed to be distributed in a low-dimensional subspace, which contains a certain degree of appearance variation. It has been demonstrated that subspace based representation is robust to some visual challenges, e.g., illumination variations and pose transformation. As existing subspace trackers employ a particle filter framework to compare candidates with the subspace model, this strategy suffers from expensive computational costs which is increased linearly with the number of particles.

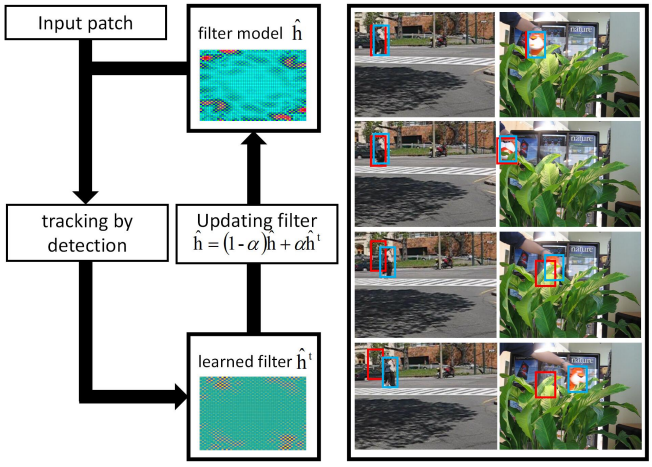


Fig. 1: a. A flow chart of the existing DCF paradigm that contains target detection, filter learning and filter updating. In the learning and updating stage, a fixed learning rate α is utilized to control the impact of the newly learned filter without considering the relationship between different frames. b. The tracking results of KCF[3] in sequence *Couple* and *Tiger2*, are shown by the red and blue rectangles. The blue designates the KCF result and the ground truth is shown in red. It demonstrates that a fixed rate learning scheme is not appropriate for all videos.

To address the above issues, we propose a novel non-negative subspace representation (NNSR) constrained learning scheme to extend DCF to subspace structure. Affine constraint is applied to construct a subspace containing several templates with auxiliary memory mechanisms. In order to preserve the information of the temporal templates in the current frame while reducing the potential drifting or false tracking, we

project the current template onto the subspace with non-negative constraints. Then we combine our NNSR learning scheme with the existing DCF paradigm. The ALM method is utilized with ADMM to solve the problem. The experimental results on the benchmark dataset, OTB50, demonstrate the effectiveness of our learning formulation. The contributions of this work are as follows: **(1)** We propose a non-negative subspace representation constrained learning scheme, which improves the adaptivity of correlation filters. **(2)** A unified optimization framework is developed to select the optimal subspace representation. **(3)** The NNSR learning scheme can directly be applied to existing DCF based methods and is shown to improve the performance in OTB50.

II. Related Work and Preliminaries

In this section we briefly introduce the correlation filter based tracking paradigm, and subspace based appearance model before presenting the proposed formulation.

DCF based methods employ circulant structure to solve a ridge regression in the frequency domain. Based on the proposal of Minimum Output Sum of Squared Error (MOSSE) filter[8], Henriques *et al.* improves the MOSSE by introducing circulant structure[9], and kernel methods[3]. Color features are fused into the correlation filter framework by Danelljan *et al.*[10] to better represent the input. SAMF[11] and DSST[12] are designed to handle scale variation. In addition, DCF based tracking methods are extended to support spatio-temporal contex learning[13][14], long-term memory[15], multi-kernel method[16], spatially regularized optimization[17][18], structural constraints[19][20], support vector representation[21] and sparse representation[22].

Assume that the original input training sample in frame t is reshaped into vector form, $\mathbf{x}^t \in \mathbb{R}^N$. Then, N training samples are generated according to circulant structure, $\mathbf{X}^t = (\mathbf{P})^{[1,2,\ldots,N]}\mathbf{x}^t = [\mathbf{x}_0^t, \mathbf{x}_1^t, \ldots, \mathbf{x}_{N-1}^t]^T$, $\mathbf{P}$ is a permutation matrix. The problem is formulated as:

$$\min_{\mathbf{h}} \sum_t z^t \sum_i \left({\mathbf{x}_i^t}^T \mathbf{h} - \mathbf{y}_i^t \right)^2 + \gamma \langle \mathbf{h}, \mathbf{h} \rangle \tag{1}$$

In order to avoid considering multiple temporal templates $\{\mathbf{x}^t\}$ with weights $\{z^t\}$, the current DCF paradigm exploits a single-frame learning scheme with a fixed updating rate. A two-stage processing framework is defined as follows: the correlation filter $\mathbf{h}^t$ is first optimized by minimizing a ridge regression by only considering the current frame:

$$\min_{\mathbf{h}^t} \sum_i \left({\mathbf{x}_i^t}^T \mathbf{h}^t - \mathbf{y}_i^t \right)^2 + \gamma \langle \mathbf{h}^t, \mathbf{h}^t \rangle \tag{2}$$

Here, a closed-form solution for this optimization problem is given by $\mathbf{h}^t = ({\mathbf{X}^t}^T \mathbf{X}^t + \gamma \mathbf{I})^{-1} {\mathbf{X}^t}^T \mathbf{y}^t$. By exploiting Fourier Transform and the circulant structure of $\mathbf{X}^t$, the solution can efficiently be calculated in the frequency domain ($\hat{\bullet}$ and *conj*$(\bullet)$ denote Fourier transform and complex-conjugate respectively):

$$\hat{\mathbf{h}}^t = \frac{conj(\hat{\mathbf{x}}^t) \odot \hat{\mathbf{y}}^t}{conj(\hat{\mathbf{x}}^t) \odot \hat{\mathbf{x}}^t + \gamma} \tag{3}$$

Then the filter model $\hat{\mathbf{h}}$ is updated by utilizing a linear recursion form after obtaining the optimal filter $\hat{\mathbf{h}}^t$ in frame t:

$$\hat{\mathbf{h}} = (1-\alpha)\hat{\mathbf{h}} + \alpha \hat{\mathbf{h}}^t \tag{4}$$

Here, α is a fixed learning rate which reflects the impact of recent information. This linear recursion updating scheme may cause drifting or false tracking when appearance changes rapidly.

The subspace based tracking methods provide a compact representation to better model the target appearance variations. These methods assume that the target appearance representations over several consecutive frames are embedded in a low-dimensional linear subspace. Under this paradigm, let $\mathbf{z}$ denote the target and $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ denote the basis templates of an underlying subspace. The current target can effectively be represented in the following form[23]:

$$\mathbf{z} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_N\mathbf{x}_N = \mathbf{X}\mathbf{c}^T \tag{5}$$

Lim *et al.*[23] proposes an incremental subspace learning method efficiently to update a low-dimensional subspace and locate the target on-line. Ross *et al.*[24] improves this incremental framework by introducing a forgetting factor which focuses more on recently acquired patches and less on earlier observations during the learning and update stages. Wang and Lu[25] utilize 2D principal component analysis directly to construct the target subspace in the original image domain. The group sparse subspace learning[26] and low-rank sparse representation[27] are proposed as options to extend linear subspace to manifold constraints.

III. Non-negative Subspace Representation Constrained Learning Scheme

In this section, we first present the construction of the template subspace $\mathbf{P}^t$ using auxiliary memory mechanisms. This subspace only contains information from the previous $t-1$ frames. Next, we calculate the projective representation of the current tracked target template, $\mathbf{x}^t$, onto the subspace with affine non-negative constraints. Then, according to the responses, we select the corresponding optimal weighting mechanisms and their combination to learn the filter model.

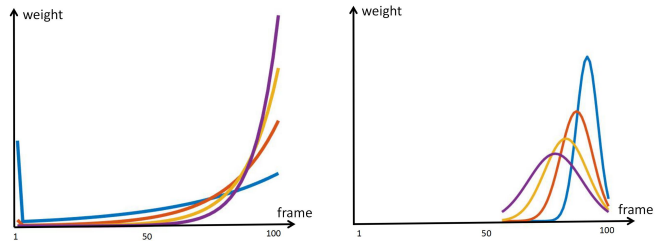
A. Construction of the Template Subspace

In frame t, we construct the template subspace by setting several specific configurations as auxiliary memory points:

$$\mathbf{f}_k^t = \sum_{i=1}^{t-1} w_k^i \mathbf{x}^i \qquad k \in \{1, 2, \ldots, K\} \tag{6}$$

Here, $\mathbf{f}_k^t$ is the k-th point, $\mathbf{x}^i$ is the tracked target template in frame i according to Equ.(3), w_k^i is the corresponding weight, K is the number of points. The weights are generated using different auxiliary memory mechanisms as follows:

Linear recursion form: As illustrated in Fig.2(a), a linear recursion form is applied with multiple learning rates. This mechanism considers the global structure between different frames[9], especially preserving the original information from



a b

Fig. 2: An intuitive illustration of auxiliary memory mechanisms to weight each frame. a. Linear recursion form with different α in (7), b. Gaussian form with different β in (8).

the first frame. The corresponding weight of the i-th filter for the k-th point ($k \in \{1, 2, \ldots, K_E\}$, K_E is the number of points using linear recursion) is calculated as:

$$w_k^i = \begin{cases} (1-\alpha_k)^{t-1}/C_{\alpha_k} & i = 1 \\ (1-\alpha_k)^{t-1-i}\alpha_k/C_{\alpha_k} & i = 2, \ldots, t-1 \end{cases} \tag{7}$$

Here, $0 < \alpha_k < 1$ is the learning rate that controls the degree of information decay, C_{α_k} is a constant to ensure that $\sum_{i=1}^{t-1} w_k^i = 1$.

Gaussian form: A Gaussian form is exploited to place emphasis on local structure. The corresponding weights of the i-th filter for the k-th point ($k \in \{1, 2, \ldots, K_G\}$, K_G is the number of points using Gaussian form) is calculated as:

$$w_k^i = \begin{cases} 0 & i = 1, \ldots, \tau_\beta \\ \dfrac{\exp\{-\frac{(i-t+2\beta_k)^2}{2\beta_k^2}\}}{C_{\beta_k}} & \text{otherwise} \end{cases} \tag{8}$$

The parameter $\beta_k \in \mathbb{N}$, $\beta_k > 0$, controls the dispersion degree of the gaussian function, C_{β_k} is a constant to ensure that $\sum_{i=1}^{t-1} w_k^i = 1$. Considering the energy distribution, $\tau_\beta = t-1-4\times\max\{\beta_k\}$ is used to reserve a fixed quantity of weights, as shown in Fig.2(b), to avoid storing entire frames.

Next, we use these K ($K = K_E + K_G$) points to construct a subspace $\mathbf{P}^t = conv\left(\{\mathbf{f}_k^t | k = 1, 2, \ldots, K\}\right)$. In order to ensure the affine structure in $\mathbf{P}^t$, NNSR learning scheme is activated when $t > t_\epsilon$ ($t_\epsilon = 10$ in our experiment).

B. Non-negative Projective Representation

Our learning scheme focuses on selecting the optimal weighting mechanisms and their combination. Exploiting subspace representation structure in Equ.(5), we formulate our learning scheme by solving the following optimization problem:

$$\begin{cases} \min\limits_{z} \dfrac{1}{2}\|\mathbf{F}z - \mathbf{x}\|_2^2 \\ \text{subject to} \quad z \geq 0 \text{ and } \mathbf{1}^T z = 1 \end{cases} \tag{9}$$

z is the tracked target template. Without loss of generality, we omit the superscript t here. $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K] \in \mathbb{R}^{n\times K}$ is a matrix with auxiliary templates as its columns, $\mathbf{x} \in \mathbb{R}^n$ and $z \in \mathbb{R}^K$. We exploit $\ell 2$-norm to measure the reconstruction error. $z \geq 0$ and $\mathbf{1}^T z = 1$ are used to guarantee that $\mathbf{F}z$ lies in $\mathbf{P}$ [28]. $\mathbf{1}$ is a column vector where all entries are one.

If $\mathbf{1}^T z = 1$ and $\|z\|_1 \leq 1$, then $z \geq 0$. So the optimization problem in Equ.(9) can be rewritten by replacing the non-negative condition and using the residual vector p ($p = \mathbf{F}z - \mathbf{x}$)[29] as follows:

$$\begin{cases} \min\limits_{p,z} \dfrac{1}{2}\|p\|_2^2 \\ \text{subject to} \quad p = \mathbf{F}z - \mathbf{x},\ \mathbf{1}^T z = 1 \text{ and } \|z\|_1 \leq 1 \end{cases} \tag{10}$$

The Augmented Lagrange Multipliers (ALM)[30] method is utilized to solve the optimization problem in Equ.(10). We introduce an additional equality constraint with corresponding slack variable and aim to solve the following optimization problem:

$$\begin{cases} \min\limits_{p,z,c} \dfrac{1}{2}\|p\|_2^2 + \lambda(\|c\|_1 - 1) \\ \text{subject to} \quad p = \mathbf{F}z - \mathbf{x}, \mathbf{1}^T z = 1 \text{ and } z = c \end{cases} \tag{11}$$

Then the Lagrange function is:

$$\begin{aligned} &L(p, z, c, \Lambda_1, \Lambda_2, \Lambda 3, u_1, u_2, u_3) \\ &= \frac{1}{2}\|p\|_2^2 + \lambda(\|c\|_1 - 1) \\ &\quad + \Lambda_1^T(\mathbf{F}z - \mathbf{x} - p) + \frac{u_1}{2}\|\mathbf{F}z - \mathbf{x} - p\|_2^2 \\ &\quad + \Lambda_2(\mathbf{1}^T z - 1) + \frac{u_2}{2}(\mathbf{1}^T z - 1)^2 \\ &\quad + \Lambda_3^T(z - c) + \frac{u_3}{2}\|z - c\|_2^2 \end{aligned} \tag{12}$$

Here $\Lambda_1 \in \mathbb{R}^n$, $\Lambda_2 \in \mathbb{R}$ and $\Lambda_3 \in \mathbb{R}^K$ are Lagrange multipliers, $u_1 > 0$, $u_2 > 0$ and $u_3 > 0$ are three penalty parameters. Alternating Direction Method of Multipliers (ADMM) is exploited to iteratively optimize these variables and the convergence can be guaranteed[31].

Step 1. Fix other variables and update p: Updating p requires solving the following problem:

$$\begin{aligned} p_* &= \arg\min_p L(p) \\ &= \arg\min_p \frac{1}{2}\|p\|_2^2 + \Lambda_1^T(\mathbf{F}z - \mathbf{x} - p) \\ &\quad + \frac{u_1}{2}\|\mathbf{F}z - \mathbf{x} - p\|_2^2 \end{aligned} \tag{13}$$

The gradient function is:

$$\nabla L(p) = p - \Lambda_1 - u_1(\mathbf{F}z - \mathbf{x} - p) = 0 \tag{14}$$

Then the closed from solution is:

$$p_* = \frac{\Lambda_1 + u_1(\mathbf{F}z - \mathbf{x})}{1 + u_1} \tag{15}$$

Step 2. Fix other variables and update z: Updating z requires solving the following problem:

$$\begin{aligned} z_* &= \arg\min_z L(z) \\ &= \arg\min_z \Lambda_1^T(\mathbf{F}z - \mathbf{x} - p) + \frac{u_1}{2}\|\mathbf{F}z - \mathbf{x} - p\|_2^2 \\ &\quad + \Lambda_2(\mathbf{1}^T z - 1) + \frac{u_2}{2}(\mathbf{1}^T z - 1)^2 \\ &\quad + \Lambda_3^T(z - c) + \frac{u_3}{2}\|z - c\|_2^2 \end{aligned} \tag{16}$$

Algorithm 1 Solving Problem Equ.(11) by ADMM[31]

Input: $\mathbf{F}$, $\mathbf{x}$, parameter λ,

Initialization: $k = 0$, $z^k = c^k = 0$, $\Lambda_1^k = 0$, $\Lambda_2^k = 0$, $\Lambda_3^k = 0$, $u_1 = u_2 = u_3 = 10$ and $\rho = 10^{-3}$.

While *not converged* **do**

1.Update p^{k+1} by:

$$p^{k+1} = \frac{\Lambda_1 + u_1(\mathbf{F}z^k - \mathbf{x})}{1 + u_1}$$

2.Update z^{k+1} by:

$$z^{k+1} = \left(u_1\mathbf{F}^T\mathbf{F} + u_2\mathbf{1}\mathbf{1}^T + u_3\mathbf{I}\right)^{-1} \cdot \left[\mathbf{F}^T(u_1\mathbf{x} + u_1p^{k+1} - \Lambda_1^k) + u_2\mathbf{1} - \Lambda_2^k\mathbf{1} + u_3c - \Lambda_3^k\right]$$

3.Update c^{k+1} by:

$$c^{k+1} = \mathcal{S}_{\frac{\lambda}{u_3}}\left(z^{k+1} + \frac{\Lambda_3^k}{u_3}\right)$$

4.Update the multipliers by:

$$\begin{aligned}\Lambda_1^{k+1} &= \Lambda_1^k + u_1(\mathbf{F}z^{k+1} - \mathbf{x} - p^{k+1})\\ \Lambda_2^{k+1} &= \Lambda_2^k + u_2(\mathbf{1}^T z^{k+1} - 1)\\ \Lambda_3^{k+1} &= \Lambda_3^k + u_3(z^{k+1} - c^{k+1})\end{aligned}$$

5.Update the parameters by:

$$u_1 = u_1(1+\rho), u_2 = u_2(1+\rho), u_3 = u_3(1+\rho)$$

6.$k = k + 1$

End While

Output: z

The gradient function is:

$$\begin{aligned}\nabla L(z) &= \mathbf{F}^T\Lambda_1 + u_1\mathbf{F}^T(\mathbf{F}z - \mathbf{x} - p) + \Lambda_2\mathbf{1}\\ &\quad + u_2\mathbf{1}(\mathbf{1}^T z - 1) + \Lambda_3 + u_3\mathbf{I}(z - c)\\ &= 0\end{aligned} \tag{17}$$

Here $\mathbf{I}$ is the identity matrix. Then the closed form solution is:

$$z_* = \left(u_1\mathbf{F}^T\mathbf{F} + u_2\mathbf{1}\mathbf{1}^T + u_3\mathbf{I}\right)^{-1} \cdot \left[\mathbf{F}^T(u_1\mathbf{x} + u_1p - \Lambda_1) + u_2\mathbf{1} - \Lambda_2\mathbf{1} + u_3c - \Lambda_3\right] \tag{18}$$

Step 3. Fix other variables and update c: Updating c requires solving the following problem:

$$\begin{aligned}c_* &= \arg\min_c L(c)\\ &= \arg\min_c \lambda(\|c\|_1 - 1) + \Lambda_3^T(z - c)\\ &\quad + \frac{u_3}{2}\|z - c\|_2^2\end{aligned} \tag{19}$$

This can be optimized in closed form by introducing a thresholding operator $\mathcal{S}_\varepsilon(x) = \text{sign}(x)\max(0, |x| - \varepsilon)$:

$$c_* = \mathcal{S}_{\frac{\lambda}{u_3}}\left(z + \frac{\Lambda_3}{u_3}\right) \tag{20}$$

We conclude the optimization steps in Algorithm 1. The algorithm terminates when $\|z^{k+1} - z^k\|_2$ falls below a threshold ϵ or $k >$ maxIter.

C. Filter Learning Model

At this stage, we reformulate Equ.(1) using the optimal representation z in Alg.(1):

$$\min_{\mathbf{h}} \sum_k z(k)\left(\mathbf{f}_k^t * \mathbf{h} - \mathbf{y}\right) + \gamma\langle\mathbf{h}, \mathbf{h}\rangle \tag{21}$$

then the optimal $\mathbf{h}$ can be learned efficiently in the frequency domain as:

$$\hat{\mathbf{h}} = \frac{\sum_{k=1}^K z(k)conj(\hat{\mathbf{f}^k}) \odot \hat{\mathbf{y}}}{\sum_{k=1}^K z(k)conj(\hat{\mathbf{f}^k}) \odot \hat{\mathbf{f}^k} + \gamma} \tag{22}$$

IV. Experiments

A. Experimental Setup

DCF Trackers: We evaluate the proposed non-negative subspace representation constrained learning scheme (NNSR) by integrating it into 4 DCFs, i.e., Kernelized Correlation Filters (KCF) tracking method [3], Spatio-Temporal Context Learning (STC) tracking method [13], Accurate Scale Estimation (DSST) tracking method [12] and Spatially Regularized Correlation Filters (SRDCF) tracking method [17].

Dataset and Evaluation Metric: The experiments are run on the OTB50 dataset [2], which contains 51 challenge test video sequences. For detailed analysis, these sequences are annotated with 11 challenging attributes including illumination variation (IV), low resolution (LR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of plane rotation (OPR), out-of-view (OV) and background clutters (BC). Two criteria proposed in [2] are exploited to evaluate the tracking performance: precision (PR) and success rate (SR). **PR:** the average center location error is used to generate a precision plot to summarize the overall performance. It shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. The percentage for a specific threshold (20 $pixels$)[32] is selected as the representative precision score. **SR:** the success plot shows the percentages of successful frames at the thresholds varied from 0 to 1, where a successful frame is defined as the intersection-over-union overlap between estimated result and the ground truth exceeding a threshold. Success rate score is the area under curve (AUC) of each success plot.

Implementation Details and Parameters: In our experiments, parameters of each tracker are fixed. The trade-off parameter λ in (11) is selected for each tracker as $\lambda = 1$ for KCF, DSST and SRDCF, $\lambda = 10$ for STC. The termination parameters for Alg.(1) is set to $\epsilon = 10^{-3}$ and maxIter$= 20$. To accelerate the optimization process, we initialize z with the corresponding value in the last frame. We set the number of points as $K = 8$ ($K_E = 4, K_G = 4$) and the corresponding weighting parameters $\{\boldsymbol{\alpha} = [0.005, 0.01, 0.02, 0.04], \boldsymbol{\beta} = [4, 6, 8, 10]\}$ for KCF, DSST and SRDCF as they all use HOG

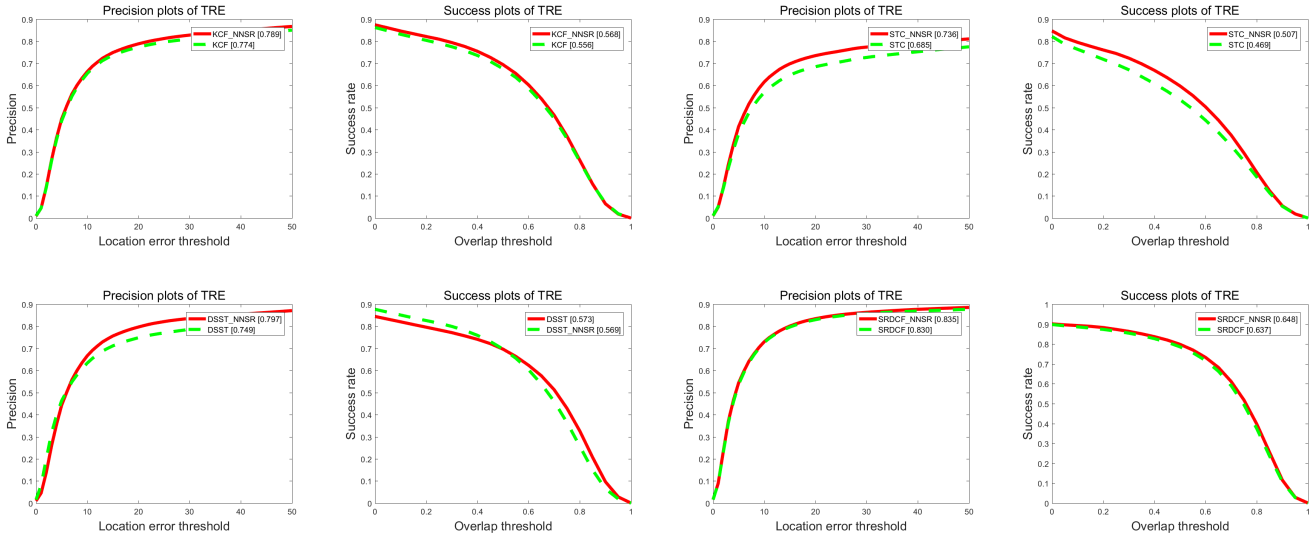


Fig. 3: Comparison of the PR/SR plots. Values in parenthesis measure the PR score and SR score.

TABLE I: Comparison of the Precision (PR) score on 11 challenging attributes. A boldface number denotes the highest score in each test.

	IV	LR	SV	**OCC**	**DEF**	MB	**FM**	**IPR**	**OPR**	**OV**	BC	**Overall**
KCF	0.729	0.502	0.728	0.758	0.757	0.627	0.579	0.728	0.749	0.642	0.776	0.774
KCF_ NNSR	**0.765**	0.502	0.728	**0.786**	**0.804**	**0.680**	**0.625**	**0.751**	**0.770**	**0.698**	0.759	**0.789**
STC	0.626	**0.603**	0.647	0.644	0.693	0.470	0.425	0.627	0.635	0.452	0.632	0.685
STC_ NNSR	**0.700**	0.593	**0.679**	**0.721**	**0.799**	**0.579**	**0.522**	**0.693**	**0.700**	**0.532**	**0.679**	**0.736**
DSST	0.724	0.563	0.714	0.722	0.717	0.576	0.514	0.722	0.717	0.573	0.692	0.749
DSST_ NNSR	**0.764**	**0.573**	**0.742**	**0.794**	**0.803**	**0.707**	**0.635**	**0.754**	**0.772**	**0.687**	**0.781**	**0.797**
SRDCF	0.774	0.594	**0.800**	0.826	0.809	**0.756**	0.714	0.773	0.805	0.726	**0.805**	0.830
SRDCF_ NNSR	0.774	**0.595**	0.789	**0.827**	**0.817**	0.740	**0.716**	**0.776**	**0.809**	**0.746**	0.774	**0.835**

features, and $\{\boldsymbol{\alpha} = [0.01, 0.02, 0.04, 0.08], \boldsymbol{\beta} = [4, 6, 8, 10]\}$ for STC with gray intensity values as features.

B. Analysis of Results

Overall performance: The comparison results over all video sequences are presented in Figure 3. It is evident that the performance of the original base trackers is improved with the NNSR learning scheme. Specifically, the improvement of precision ranges from 0.5% for SRDCF, 1.5% for KCF, 4.8% for DSST to 5.1% for STC. The success rate increases from 63.7% to 64.8% for SRDCF, 55.6% to 56.8% for KCF and 46.9% to 50.7% for STC, while DSST suffers a slight decrease of 0.4%. In general, we achieve an average gain of 3% in precision and 1.4% in success rate.

The experimental results on the challenge attributes: For a complete comparison, we also report extensive results for 11 challenging attributes in Table I and Table II. From the results, we can observe that the proposed NNSR scheme achieves very promising performance. For deformation, fast motion, out-of plane rotation and out-of-view, we obtain average percentage improvement of 6.2%, 6.7%, 3.6%, 6.8% in precision and 4.5%, 3.7%, 2.2%, 4.8% in success rate respectively. Thus our NNSR learning scheme enhances the performance of base trackers on these attributes, which suggests that our learning scheme provides a long-term efficient memory model to DCF paradigm. Other attributes such as occlusion and in plane rotation also achieve a boost compared with the original trackers in most cases. The speed comparison with base trackers is listed in Table III, NNSR learning scheme occupies about 20% computation load.

TABLE III: Speed analysis (fps)

	KCF	STC	DSST	SRDCF
Base Tracer	58.5	79.4	32.5	5.2
Base Trace + NNSR	46.5	57.4	26.5	4.4

V. Conclusion

In this paper, we proposed a general non-negative subspace representation constrained learning scheme (NNSR) to extend DCF to subspace based representation. Relevant information

TABLE II: Comparison of the Success Rate (SR) score on 11 challenging attributes. A boldface number denotes the highest score in each test.

	IV	LR	SV	OCC	**DEF**	MB	**FM**	IPR	**OPR**	**OV**	BC	Overall
KCF	0.527	0.383	0.488	0.546	0.570	0.492	0.455	0.519	0.530	0.538	**0.564**	0.556
KCF_ NNSR	**0.555**	0.383	**0.491**	**0.569**	**0.604**	**0.532**	**0.492**	**0.536**	**0.548**	**0.585**	0.557	**0.568**
STC	0.414	**0.432**	0.423	0.466	0.489	0.358	0.350	0.435	0.427	0.388	0.436	0.469
STC_ NNSR	**0.481**	0.420	**0.444**	**0.514**	**0.574**	**0.447**	**0.407**	**0.479**	**0.476**	**0.439**	**0.478**	**0.507**
DSST	**0.566**	**0.442**	**0.540**	**0.568**	0.561	0.477	0.446	**0.550**	0.539	0.500	0.533	**0.573**
DSST_ NNSR	0.551	0.404	0.493	0.553	**0.603**	**0.548**	**0.495**	0.536	**0.546**	**0.566**	**0.567**	0.569
SRDCF	0.596	**0.476**	0.616	0.632	0.628	**0.592**	0.572	**0.592**	0.609	0.591	**0.604**	0.637
SRDCF_ NNSR	**0.607**	0.473	0.616	**0.646**	**0.650**	0.584	**0.577**	0.591	**0.622**	**0.619**	0.595	**0.648**

is preserved by our multi-memory mechanisms. We introduce affine subspace to formulate an optimization problem and utilize ALM to solve it. The experimental results on a benchmark dataset demonstrate the effectiveness of our learning formulation.

VI. Acknowledgments

The paper is supported by the National Natural Science Foundation of China (Grant No.6137305561672265), UK EPSRC Grant EP/N007743/1, MURI/EPSRC/dstl Grant EP/R018456/1, and the 111 Project of Ministry of Education of China (Grant No. B12018).

References

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

[2] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE TPAMI*, vol. 37, no. 3, pp. 583–596, 2015.

[4] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE TPAMI*, vol. 37, no. 9, pp. 1834–48, 2015.

[5] M. Kristan, R. Pflugfelder, and A. L. *et al.*, "The visual object tracking vot2014 challenge results," 2014.

[6] M. Kristan, J. Matas, and A. L. *et al.*, "The visual object tracking vot2015 challenge results," in *Visual Object Tracking Workshop 2015 at ICCV2015*, Dec 2015.

[7] M. Kristan, A. Leonardis, and J. M. *et al.*, "The visual object tracking vot2016 challenge results," Oct 2016.

[8] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*, 2010, pp. 2544–2550.

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*, 2012, pp. 702–715.

[10] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR*, 2014, pp. 1090–1097.

[11] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *ECCV*, 2014, pp. 254–265.

[12] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014.

[13] K. Zhang, L. Zhang, M. H. Yang, and D. Zhang, "Fast tracking via spatio-temporal context learning," *Computer Science*, 2013.

[14] T. Xu and X.-J. Wu, "Fast visual object tracking via distortion-suppressed correlation filtering," in *Smart Cities Conference (ISC2), 2016 IEEE International*. IEEE, 2016, pp. 1–6.

[15] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *CVPR*, 2015, pp. 5388–5396.

[16] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *ICCV*, 2015, pp. 3038–3046.

[17] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV*, 2015, pp. 4310–4318.

[18] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," 2017.

[19] S. Liu, T. Zhang, X. Cao, and C. Xu, "Structural correlation filter for robust visual tracking," in *CVPR*, 2016, pp. 4312–4320.

[20] T. Xu and X. Wu, "Fast correction visual tracking via feedback mechanism," in *International Conference on Intelligent Science and Big Data Engineering*. Springer, 2015, pp. 208–219.

[21] W. Zuo, X. Wu, L. Lin, L. Zhang, and M.-H. Yang, "Learning support correlation filters for visual tracking," *arXiv preprint arXiv:1601.06032*, 2016.

[22] T. Zhang, A. Bibi, and B. Ghanem, "In defense of sparse tracking: Circulant sparse tracker," in *CVPR*, 2016, pp. 3880–3888.

[23] J. Lim, D. A. Ross, R. S. Lin, and M. H. Yang, "Incremental learning for visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2005.

[24] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.

[25] D. Wang and H. Lu, "Object tracking via 2dpca and $\ell 1-$regularization," *IEEE SPL*, vol. 19, no. 11, pp. 711–714, 2012.

[26] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *CVPR*, 2013, pp. 2371–2378.

[27] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *ECCV*, 2012, pp. 470–484.

[28] Y. J. Xu and X. J. Wu, "An affine subspace clustering algorithm based on ridge regression," *Pattern Analysis and Applications*, vol. 20, no. 2, pp. 557–566, 2017.

[29] D. Chen and R. J. Plemmons, *Nonnegativity constraints in numerical analysis*. WORLD SCIENTIFIC, 2009.

[30] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[32] B. Babenko, M. H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE TPAMI*, vol. 33, no. 8, pp. 1619–1632, 2011.