2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Spatial Graph Regularized Multi-kernel Subtask Cross-correlation Tracker

Baojie Fan,

***Abstract*— Some impressing multi-kernel or multi-task correlation filter trackers only focus on boosting the discrimination of multi-channel features, or exploiting the interdependence among different tasks. However, the cooperation and complementary of both technologies are missed, and the spatial structure among or inside target regions is also ignored. Therefore, this paper proposes a spatial graph regularized hierarchical subtask multi-kernel cross-correlation tracker (GHMK) via Gaussian process regression view, which enjoys the merits of multi-subtask multi-kernel learning and Gaussian process regression to jointly learn kernel cross-correlation filters, and makes them complement and boost each other. The interdependence and discrimination among multi-kernel multi-subtask are jointly exploited by group structure sparsity, which is also used to evaluate spatial feature selection. The spatial graph is constructed via cross similarity to maintain the geometric structure among or inside hierarchy subtasks. Besides, the developed model is general, and provides a unified solution from GPR for CF trackers without boundary effect. Comprehensive experiments demonstrate its favorable and competitive performance against the state-of-the-art trackers.**

I. INTRODUCTION

Recent years have witnessed the astonishing advancements in discriminative correlation filter (DCF) based trackers, such as CSR-DCF [1], f-dKCF [2], BACF [3] and ECO [4]. With the trade-off between computational load and tracking accuracy, DCF-based trackers show superior performance. Its core inherit is to train a kernelized ridge regression (KRR) and performing operations in the Fourier domain. Tang et al [5] extend the traditional single kernel correlation filter tracker to multiple kernel version MKCFup, to boost the discriminative ability of multi-cue features. But they still fail to fully explore the cooperation and complementary of different features. In [6], MCPF takes the merit of particle filter and multi-task correlation filter to exploit the interdependence among different particles with multi-channel features, and jointly learns the multi-task correlation particle filter (MCPF). It concatenates all the multi-layer features into a single kernel space to represent the entire target samples, this operation brings much redundancy and buries the discriminative characters of different features, even some distractive features are included. Besides, the circular correlation make it very difficult to exploit the target structure, and bring the unwanted boundary effects. SRDCF [7] only relax the boundary effect to some extent, but can not eliminate the effect thoroughly and mine the structure relationship.

B. J. Fan is working at Automation College in NJUPT: jobfbj@gmail.com. This work is supported by Ministry of Science and Technology of the People's Republic of China (2019YFB1310300), National Natural Science Foundation of China (No. 61876092, U2013210), State key Laboratory of Robotics (No.2019-O07) and State key Laboratory of Integrated Service Network (ISN20-08).

Fig. 1: The tracking results of related trackers in challenging sequences

To solve above problems, we develop a spatial graph regularized hierarchical subtask multi-kernel cross correlation tracker via Gaussian process regression view (GPR). Specifically, the hierarchical subtasks are used to learn the multi-kernel GPR model by computing Gaussian covariance matrix, and then estimate the regression values of the hierarchical subtask candidates. This operation can eliminate the boundary effect thoroughly as well as explore the kernel trick. Then, we divide the target region into hierarchical subregions. The trace of each subregion is defined one subtask. Different non-linear kernel spaces are assigned to multiple features of hierarchical subtasks. One kernel space is associated to each feature, and boost their discriminative ability and interdependence. In the following, we takes the cooperations and interdependencies among hierarchical subtasks into account to learn multi-subtask multi-kernel cross-correlation filters jointly. It is achieved by mixed group sparse norm regularized multi-subtask learning, which also choose the suitable spatial features. The spatial graph regularized term is introduced to not only exploit the intrinsic structure relationship among hierarchical subtasks, and but also preserve their spatial layout structure. The proposed general model can be applied to many kinds of kernel functions and transformations of samples, and provide a unified solution from GPR for CF trackers without boundary effect. Finally, we evaluate extensive experiments on six benchmark datasets to demonstrate its favorable performance, shown in Figure 1. The main contributions of our work are three-fold:

(1) We develop a spatial graph regularized hierarchical subtask multi-kernel cross correlation regression tracker. It

has general property to incorporate any kernel functions with any transformation of target samples. The proposed model provides a unified solution via GPR for CF trackers without boundary effect.

(2) A spatial Lapalcian graph is constructed via subtask cross similarity to not only exploit the intrinsic structure relationship among hierarchical subtasks, and but also preserve their spatial layout structure inside target region.

(3) The proposed tracking model enjoys the merits of hierarchical subtask multi-kernel learning and Gaussian process regression, which simultaneously explores the interdependence and complementary among multiple subtasks and boost the discriminative ability of different features.

II. Related Works

In this section, we briefly review the tracking algorithms closely related to the proposed model. The reader is referred to more thorough reviews in [8].

Trackers based on correlation filter: Starting with KCF tracker [9], many improved DCF methods are developed from different views, such as MKCFup [5] for different features, the spatial-temporal regularized method SRDCF [7]. The background-aware tracking algorithm [3] enhances the filter discriminativity. GPRT [10] with gaussian process regression. CACF [11] to mitigate boundary effects. LGCF [12] exploits locally reliable structure. Ma et al.[13] combine three layers of convolution filters to develop a hierarchical ensemble DCF tracker. C-COT [14] efficiently integrates multi-resolution shallow and deep feature maps by the continuous convolution operator, its efficient extension is ECO [4]. In [6], a global multiple tasks correlation particle filter tracker (MCPF) takes the merit of particle filter and multi-task correlation filter.

Trackers based on deep learning network: Some impressive deep tracking networks are designed and pre-trained for target-specific location. A fully convolutional Siamese tracking network (SiamFC) [15] is trained off-line to learn the similarity metric. An adaptive decision-making tracker [16] learn the policy of suitable tracking feature selection in a reinforcement learning fashion. Dynamic Siamese tracking network (DSiam) [17] introduces the transformation for training samples. Siamese region proposal network (Siamese-RPN) [18] introduces region proposal subnetwork for the classification and position regression via one-shot learning. Generative adversarial learning is introduced to enhance the quality and number of samples, such as VITAL [19].

III. Background

Consider the target tracking as a prediction problem: given a training dataset $\mathcal{D} = [x_i, y_i], i = 1, 2, ..., N$, $X = [x_1, ..., x_N] \in R^{D\times N}$ is the input data, and $Y = [y_1, ..., y_N] \in R^N$ is the desired vector corresponding to X. This problem can be addressed from two views: kernel ridge regression and Gaussian process regression.

A. Gaussian Process Regression Correlation Filter Tracker

The Gaussian process regression model can solve the prediction problem with the probabilistic model $Y|F \sim \mathcal{N}(F, \sigma_n I)$, $F \sim \mathcal{N}(0, K)$, where $F = [f(x_1), f(x_2), ..., f(x_N)]$, K is the covariance function given by $K_{ij} = k(x_i, x_j)$, $k(x_i, x_j) = \sigma_f^2 exp[\frac{-(x_i - x_j)^2}{2l^2}] + \sigma_n^2 \delta(x_i, x_j)$, where σ_f, σ_n are the weighted parameters for the covariance and noise model, respectively. $\delta(x_i, x_j)$ is the Kronecker delta function. Then, we calculate the posterior predictive distribution of the test samples $x_j^*, f^* = f(x_j^*), j = 1, ..., M$, $F^* = [f(x_1^*), f(x_2^*), ..., f(x_M^*)]$.

$$\begin{bmatrix} F \\ F^* \end{bmatrix} |(x, x^*) \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{xx} & K_{xx^*} \\ K_{x^*x} & K_{x^*x^*} \end{bmatrix}\right) \tag{1}$$

$$\begin{bmatrix} F^* \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{x^*x^*} & K_{xx^*} \\ K_{x^*x} & K_{xx} + \sigma_n^2 I \end{bmatrix}\right) \tag{2}$$

$$\begin{aligned} F^*|Y \sim \mathcal{N}(&K_{x^*x}(K_{xx} + \sigma^2 I)^{-1} Y, \\ &K_{x^*x^*} - K_{x^*x}(K_{xx} + \sigma^2 I)^{-1} K_{xx^*}^T) \end{aligned} \tag{3}$$

where $K_{xx} = [k(x_1, x_1), ..., k(x_i, x_j), ..., k(x_N, x_N)]$, $K_{x^*x} = [k(x^*, x_1), ..., k(x^*, x_N)]$, $K_{x^*x^*} = k(x^*, x^*)$.

The GPR can be introduced into kernel correlation filter tracking process. The circular training samples are used to straightforwardly construct the covariance matrix $K_{x_i x_j}$, where i, j are the frame index. The predicted expectation $E(x_t)$ of the test sample x_t in current frame is reported as $E(x_t) = K_{x_t x_i}(K_{x_i x_i} + \sigma^2 I)^{-1} y_i, i < t$ are the circular training samples in the previous frames. $f(x_t) = \sum_i \beta_i K_{x_t, x_i}(K_{x_i, x_i} + \sigma_n^2 I)^{-1} y_i$ can be efficiently represented as the linear weighted combination of $E(x_t)$ as

$$f(x_t) = \sum_i \beta_i K_{x_t, \sum_{i=1}^{t-1} \beta_i x_i} (\sum_{i=1}^{t-1} \beta_i K_{x_i, x_i} + \sigma_n^2 I)^{-1} y_i \tag{4}$$

where β_i is the weight parameter.

From the above analysis, we observe that: If KRR uses the same kernel as the covariance function in GPR. Meanwhile, if the regularisation parameter λ in KRR is the same as the variance σ_n^2 in GPR. Then, the estimate of KRR function coincides with the GPR posterior mean. Their cooperation can jointly take the advantages of both views.

IV. Spatial Graph Regularized Hierarchical Subtask Multi-kernel Cross-correlation Tracker

Inspired from the above founding, we develop a spatial graph regularized hierarchical subtask multi-kernel cross correlation tracker via GPR view, Defining the pairwise training sample set $\mathcal{D} = [x^i, y^i], i = 1, ..., P$, $x^i \in R^{n_x \times n_y}$ is the input, $y^i \in R^{m_x \times m_y}$ is the label. z represents the test samples. The sample set $\boldsymbol{x} = [x_1, ..., x_N] \in R^{n_x \times n_y \times N}$ is generated from training sample x^i, with the predefined transform function $\mathcal{T}()$. Notice that, $\mathcal{T}()$ is flexible, not limited to the circular shifts. The kernel cross-corrrelator for the test sample z with its desired output $y_z \in R^{m_x \times m_y}$ is defined as $y_z = K_{z,\boldsymbol{x}} * w$, where w is a correlation filter,

$K_{z,\boldsymbol{x}}$ is a single kernel matrix with its vector $k(z,\boldsymbol{x}) = [k(z,x_1),...,k(z,x_N)]$. Besides, we introduce the Laplacian graph regularized term to consider the spatial structure relationship among samples. The graph regularized kernel correlation filter w can be learned from GPR view by

$$\begin{aligned} \underset{\alpha}{argmin} \sum_{i=1}^{P} \|K_{x^i x^i} w - y^i\|_F^2 + \lambda_1 \|w\|_2^2 + \lambda_2 \boldsymbol{w^T} L \boldsymbol{w} \\ s.t. w = \sum_{i,j=1}^{P} \alpha_{i,j} k(x^i, x^j) \end{aligned} \tag{5}$$

where $K_{x^i x^i} = [k(x^i, x_1^i), ...k(x^i, x_N^i)]$ is the single kernel matrix. $[x_1^i, ..., x_N^i]$ are the generated samples from training sample x^i. $L = D - C$ is the Laplacian matrix, D is a diagonal matrix with each element $d_{ii} = \sum_{j=1}^{P} C_{ij}$. C is an affinity matrix with each element C_{ij} denoting the similarity weight of samples x^i and x^j. Therefore, the closed-form solution of α is obtained in Fourier domain as

$$\alpha = [\sum_{i=1}^{P} \hat{K}_{x^i x^i} \odot (\hat{K}_{x^i x^i} + \lambda_1 I + \lambda_2 L)]^{-1} \sum_{i=1}^{P} \hat{K}_{x^i x^i} \odot \hat{y}^i. \tag{6}$$

Notice that, Eq.(6) can be a unified solution for the existing CF trackers from GPR without boundary effect. The ultimate detection and update formula are as follows from GPR view.

$$\begin{aligned} f(x_t) &= K_{x_t Z_{t-1}} \odot H_{t-1}^{-1} \odot J_{t-1} \\ Z_t &= \mu x_{t-1} + (1-\mu) Z_{t-1} \\ H_t &= \mu K_{x_t x_t} \odot (K_{x_t x_t} + \lambda_1 I + \lambda_2 L) + (1-\mu) H_{t-1} \\ J_t &= \mu K_{x_t x_t} \odot y_t \end{aligned} \tag{7}$$

In the following, we incorporate hierarchical subtask multi-kernel learning into the model in (5), to boost the discriminative ability of multi-cue features and exploit the interdependence among hierarchical subtasks. Therefore, the extended multi-kernel vector $k(z,\boldsymbol{x})$ is defined as $k(z,x_k) = \boldsymbol{d}^T \boldsymbol{k}(z,x_k) = \sum_{b=1}^{B} d_b k_b(x_t, x_k)$, where $k_b, b = 1, ...B$ is the base kernel, B is the number of base kernels. $\boldsymbol{d} = [d_1, ...d_b, ..., d_B]^T, \sum_{b=1}^{B} d_b = 1$, $\boldsymbol{k}(z,x_k) = [k_1(z,x_k), ...k_b(z,x_k), ..., k_B(z,x_k)]^T$. Besides, we divide the entire target samples $T = [T^1, ..., T^m]$ into hierarchical image patches with different sizes $T_h = [T_h^1, ..., T_h^m], h = 1, ..., H$, the trace of each target patch is defined as one subtask. So, the graph regularized hierarchical subtask multi-kernel cross correlation tracking model is presented as

$$\begin{aligned} \underset{w,d_b}{min} \| \sum_{b=1}^{B} d_b K_{xx}^b w - y\|_F^2 + \lambda_1 \|w\|_{2,1} \\ + w^T \sum_{b=1}^{B} d_b K_{xx}^b w + \lambda_3 w^T L w. \end{aligned} \tag{8}$$

where $K_{xx} = [k_1(x,x_1), ...k_B(x,x_N)]$ is the multi-kernel matrix. $[x_1^i, ..., x_N^i]$ are the generated samples from training sample x. $\|w\|_{2,1} = \|\sqrt{\sum_{j=1}^{c} w_j \odot w_j}\|_1$ is the group structured sparsity, where c is the number of cues or tasks. It not only introduces hierarchical subtask learning to exploit the interdependence and cooperation for multi-kernel multi-cue features, but also evaluates the spatial structure feature selection on filter $w_\psi = diag(\psi)w$,$diag(\psi)$ is the binary diagonal indicator to enable some elements in w, $\|\psi\|_0 = \|w_\psi\|_0 = \|w\|_0$, approaching by$\|w\|_1$.

A. Optimizing Algorithm

The alternating iteration approach can be used to solve it, which optimizes w and d_b iteratively until the stopping criteria reaches. Slack variable w^s and augmented Lagrange multiplier q are introduced to obtain the Lagrange function of Eq.(8) as

$$\begin{aligned} \underset{w,d_b}{min} \| \sum_{b=1}^{B} d_b K_{xx}^b w - y\|_F^2 + \lambda_1 \| \sqrt{\sum_{j=1}^{c} w_j^s \odot w_j^s} \|_1 + \\ \lambda_2 w^T \sum_{b=1}^{B} d_b K_{xx}^b w + \lambda_3 w^T L w + \gamma \|w - w^s + \frac{q}{\gamma}\|_2^2. \end{aligned} \tag{9}$$

ADMM method is adopt to iteratively solve Eq.(9).

Solving w, when others are fixed: The Eq.(9) is transformed into the following convex optimizing formula, and it can be solved efficiently in Fourier domain.

$$\begin{aligned} \underset{w}{min} \| \sum_{b=1}^{B} d_b K_{xx}^b w - y\|_F^2 + \lambda_2 w^T \sum_{b=1}^{B} d_b K_{xx}^b w \\ + \lambda_3 w^T L w + \gamma \|w - w^s + \frac{q}{\gamma}\|_2^2. \end{aligned} \tag{10}$$

The closed-form solution of w is presented with FFT as

$$\hat{w} = \frac{\sum_{b=1}^{B} \mathcal{F}(d_b K_{xx}^b) \odot \hat{y} + \gamma \hat{w}_s - \hat{q}}{\sum_{b=1}^{B} \mathcal{F}(d_b K_{xx}^b) \odot (\mathcal{F}(d_b K_{xx}^b) + \lambda_2 + \lambda_3 L) + \gamma} \tag{11}$$

The division in Eq.(11) is an element-wise operation. Denoting $W_M^t = \mathcal{F}(d_b K_{xx}^b) \odot \hat{y}$, $W_N^t = \mathcal{F}(d_b K_{xx}^b) \odot (\mathcal{F}(d_b K_{xx}^b)$, they are updated as follows:

$$\begin{aligned} W_M^t =& \mu \mathcal{F}(d_b K_{xx}^b) \odot \hat{y} + (1-\mu) A_M^{t-1} \\ W_N^t =& \mu \mathcal{F}(d_b K_{xx}^b) \odot (\mathcal{F}(d_b K_{xx}^b + \lambda_2 + \lambda_3 L) \\ & + (1-\mu) A_N^{t-1}. \end{aligned} \tag{12}$$

$\hat{w}$ can be evaluated and updated efficiently frame by frame as

$$\hat{w} = \frac{W_M^t + \gamma \hat{w}_s - \hat{q}}{W_N^t + \gamma}. \tag{13}$$

Solving w^s, when other parameters are given: w^s is updated by solving the following optimizing problem

$$\underset{w^s}{argmin} \lambda_1 \| \sqrt{\sum_{j=1}^{c} w_j^s \odot w_j^s} \|_1 + \gamma \|w - w^s + \frac{q}{\gamma}\|_2^2. \tag{14}$$

In order to efficiently calculate the closed-form solution, we transform the index of group structure sparsity in Eq.(14) from channels to pixel coordinates. Eq.(14) is rewritten as

$$\underset{w_{ij}^s}{argmin} \lambda_1 \sum \|w_{ij}^s\|_2^2 + \gamma \|w_{ij} - w_{ij}^s + \frac{q_{ij}}{\gamma}\|_2^2. \tag{15}$$

where i, j are the row and column coordinates. The optimal solution w_{ij}^s is presented as

$$w_{ij}^s = max(0, 1 - \frac{\lambda_1}{\gamma \| w_{ij} + \frac{q_{ij}}{\gamma} \|_2} (w_{ij} + \frac{q_{ij}}{\gamma}). \quad (16)$$

Solving q, γ, when other variables are given: They are updated by the following formula

$$\begin{cases} q = q + \gamma(w - w^s) \\ \gamma = max(\gamma_{max}, 1.05\gamma) \end{cases} \quad (17)$$

Solving d_b, when other variables are given: Given w, Eq.(8) is transformed as a convex and constrained optimizing problem.

$$\underset{d_b}{argmin} \sum_{b=1}^{B} \| d_b K_{xx}^b w - y_d \|_F^2 + \lambda_2 w^T d_b K_{xx}^b w. \quad (18)$$

where $y_d = \frac{y}{B}$. By evaluating the first derivation of d_b and setting it to zero, we can obtain the solution of d_b.

$$d_b = \frac{(K_{xx}^b w)^T (2y_d - \lambda_2 w)}{2(K_{xx}^b w)^T (K_{xx}^b w)} = \frac{D_M^t}{D_N^t} \quad (19)$$

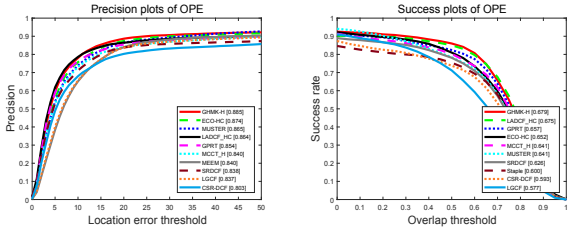
where $D_M^t = (K_{xx}^b w)^T (2y_d - \lambda_2 w)$, $D_N^t = 2(K_{xx}^b w)^T (K_{xx}^b w)$, they are updated by

$$\begin{aligned} D_M^t &= \mu (K_{xx}^b w)^T (2y_d - \lambda_2 w) + (1 - \mu) D_M^{t-1} \\ D_N^t &= 2\mu (K_{xx}^b w)^T (K_{xx}^b w) + (1 - \mu) D_N^{t-1}. \end{aligned} \quad (20)$$

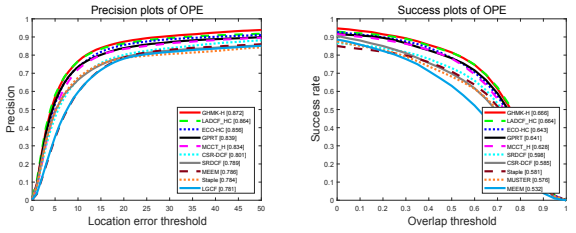
The optimal $\hat{d}_b$ can be efficiently evaluated and updated in Fourier domain with $\mathcal{F}(K_{xx}^b w) = \mathcal{F}(K_{xx}^b) \odot \mathcal{F}(w)$.

V. Experiments

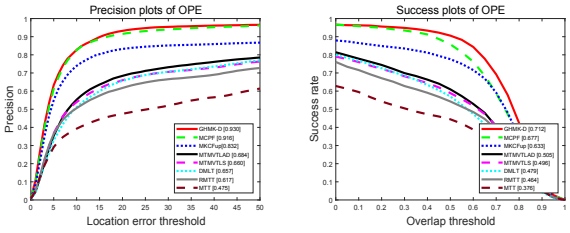
Implementation Details: We evaluate two versions of the developed tracking model equipped with different features: hand-crafted and deep features, denoted as GHMK-H and GHMK-D, respectively. Gaussian kernel is used for both types of features. The hand-crafted features include 31-channels HOG and 10-channels color names(CN), the cell size in both HOG and CN features is 4×4. The deep features are extracted from the middle convolutional layers–conv3,conv4 and conv5 layers in VGGNet-M-2048. We set the values of parameters as follows: the updating rate $\mu = 0.005$, the parameters of gaussian kernel in Eq.(1) $\sigma_f = 1, \sigma_n = 0.01, l = 1.4$, $\lambda_1 = 0.1, \lambda_2 = 1, \lambda_3 = 1, \gamma_{max} = 30$, $B = 2$ for hand-crafted features, $B = 3$ for deep features. The maximum iteration in solving w is 20. The variances of affine parameters for particle filter are set to $(0.01, 0.001, 0.001, 0.01, 4, 4)$, We maintain all the parameters same for all the following experiments. We evaluate the proposed GHMK-H and GHMK-D in MATLAB 2016a on a computer with an intel I7 4.00GHz CPU and 32GB RAM. The MatConvNet toolbox is used for extracting the deep features, which is implemented on a GeForce GTX 1080Ti GPU.



(a) The tracking results of GHMK-H on OTB-2013

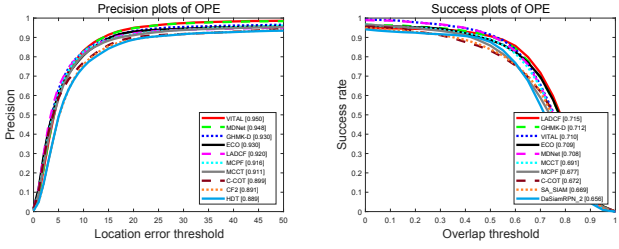


(b) The experimental results of GHMK-H on OTB-2015

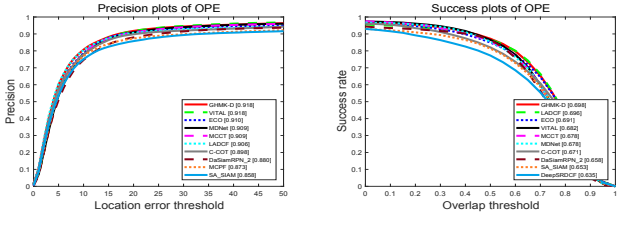


(c) Compared with related multi-task multi-kernel trackers

Fig. 2: Performance of GHMK-H with hand-crafted features on OTB benchmark dataset.



(a) The evaluation and comparison of GHMK-D on OTB-2013



(b) The evaluation and comparison of GHMK-D on OTB-2015

Fig. 3: The tracking results of GHMK-D with deep features on OTB benchmark dataset.

TABLE I: The evaluations of GHMK-D and some state-of-the-art trackers on the VOT 2016 dataset.

	GHMK-D	C-cot	Eco	Vital	Mcpf	Crest	Staple	Ebt	Csrdcf	Tcnn	Mdnet	SiamFC	BACF	TADT
A	**0.55**	**0.54**	**0.55**	**0.54**	0.52	0.514	0.51	0.441	0.52	**0.54**	0.53	**0.55**	**0.56**	0.55
R	**0.69**	0.85	**0.72**	**0.83**	0.948	1.08	1.35	0.90	1.17	0.96	1.20	1.382	1.88	1.17
EAO	**0.378**	**0.331**	**0.374**	0.323	0.319	0.283	0.295	0.291	0.276	0.325	0.257	0.277	0.223	0.299

TABLE II: Tracking comparisons on the Temple Color dataset are presented with AUC and PS.

	DSLT	GHMK-D	ECO	C-COT	CREST	MCPF	DSRDCF	CF2	HDT	Staple	Muster	MEEM	KCF(LAB)
AUC	**0.587**	**0.599**	**0.597**	0.573	0.555	0.545	0.537	0.484	0.480	0.498	0.459	0.459	0.431
PS	**0.807**	**0.809**	**0.798**	0.781	0.731	0.774	0.738	0.703	0.686	0.665	0.636	0.639	0.601

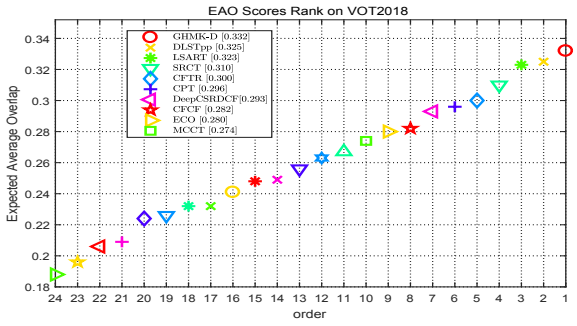


Fig. 4: The EAO plot of GHMK-D evaluated on VOT-2018, larger values represent better tracking performance.

TABLE III: Performance comparisons on the UAV123 dataset with Success and precision

Metric	GHMK-D	ARCF	ECO	SiamRPN	SRDCF	CPCF	KAOT
PRE	0.757	0.641	0.741	0.768	0.582	0.661	0.686
AUC	0.546	0.421	0.525	0.557	0.402	0.466	0.479

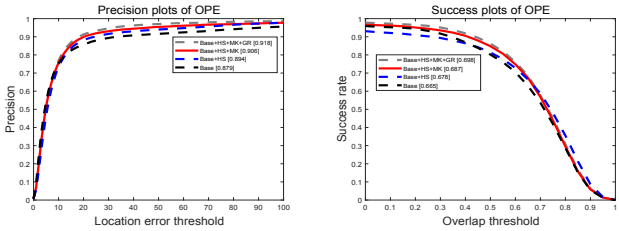


Fig. 5: The tracking contributions of different layered subtask correlation filters on OTB-2015.

A. *Comparisons with State-of-the-art Trackers*

The tracking experiments are evaluate on the benchmark datasets including OTB benchmark [8], TempleColor [20], VOT-2016 [21], VOT-2018 [22] and UAV123 [23].

OTB datasets: For hand-crafted features, trackers include KCF [9], SRDCF [7], TGPR [24], GPRT [10], LGCF [12], LADCF-HC [25], MKCFup [5], ECO-HC [4]. Figure 2.(a) and (b) show GHMK-H perform favorably on both the distance precision and overlap success rate among 29 trackers in OTB-2013, OTB-2015 and some states-of-the-arts trackers. For trackers with deep feathers including CF2 [13], MDNet [26], SA-SIAM [27], ACFN [28], ECO [4], MCPF [6], LADCF [25], VITAL [19]. Figure 3.(a) and (b) illustrate that GHMK-D has the advanced tracking performance, better precision and higher success rate than many deep trackers.

Besides, we compare GHMK-D with related state-of-the-art multi-task multi-kernel trackers including MTMVLAD [29], MTMVLS [29], DMLT [30], MTT [31], MCPF [6], MKCFup [5]. Figure 2.(c) reports that the proposed multi-task multi-kernel correlation tracker based on GPR has better accuracy and success rate.

VOT dataset: We evaluate the proposed tracker GHMK-D on the VOT-2016 [21] and VOT-2018 [22] benchmarks, respectively, and compare the result with state-of-the-art trackers. Table I demonstrates that GHMK-D has the competitive performance in VOT-2016 with EAO value 0.378. Figure 4 illustrates the favorable tracking results of GHMK-D against the top-ranked participants in VOT-2018, such as DSLTpp [32], ECO [4], MCPF [6].

Temple Color dataset: We also report the experimental results on Temple Color dataset [20], including: DSLT [32], C-COT [14], MCPF [6], ECO [4], CF2 [13]. Table II presents that GHMK-D performs favourably against some recent related tracking algorithms. It achieves the PS and AUC scores of $(0.809, 0.599)$, better than the values $(0.774, 0.545)$ of MCPF [6], $(0.703, 0.484)$ of CF2 [13].

UAV123: UAV123 [23] is a large dataset captured from low-altitude UAVs. This dataset has a total of over 110K frames and 123 video sequences. We compare the proposed GHMK-D with previous approaches on this dataset. As shown in Table III, our GHMK achieves the competitive performance with an AUC score of 0.546 and a precision score of 0.757, less than SiamRPN. However, it outperforms the representative approaches, such as CPCF [33], KAOT [34], ARCF [35], ECO [4]. This demonstrates the powerful capabilities of our method on tracking far and tiny objects.

VI. Discussion

The proposed tracking framework is general, KCF, GPRT, MCPF and MKCFup can be regarded as the special cases of our model. Different from them, the develop model can incorporate many types of kernel functions and any transformations of training samples into the tracking model, such as affine transformation, rotation, not limited to circular shifts. The hierarchical subtask multi-kernel learning jointly boosts the powerful discriminativity of different features, and mines the cooperations and interdependencies among hierarchical subtasks. The spatial graph regularized term is introduced to not only exploit the intrinsic structure relationship among hierarchical subtasks, and but also preserve their spatial layout structure.

To verify the contribution of each component in our tracking framework, we evaluate three variants of GHMK-D on OTB2015 tracking benchmark. The global subtask tracking model MKCF is regarded as the baseline tracker, represented as "Base". The hierarchical subtasks, multi-kernel and graph regularized terms are progressively added to the baseline tracker, denoted as "Base+HS", "Base+HS+MK", "Base+HS+MK+GR". Figure 5 presents the impact of progressively integrating one component at a time.

VII. Conclusions

In this paper, we develop a general graph regularized multi-task multi-kernel cross correlation tracker from GPR view. The cooperation and discriminativity among multi-cue features is exploited by multi-task multi-kernel learning with group structured sparsity and graph regulator. Experiments demonstrate its favorable performance against the state-of-the-art trackers.

References

[1] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan, "Discriminative correlation filter with channel and spatial reliability," in *CVPR*, 2017, pp. 4847–4856.

[2] Linyu Zheng, Ming Tang, Yingying Chen, Jinqiao Wang, and Hanqing Lu, "Fast-deepkcf without boundary effect," in *CVPR*, 2019, pp. 4020–4029.

[3] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey, "Learning background-aware correlation filters for visual tracking.," in *ICCV*, 2017, pp. 1144–1152.

[4] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg, et al., "Eco: Efficient convolution operators for tracking.," in *CVPR*, 2017, pp. 6638–6646.

[5] Ming Tang, Bin Yu, Fan Zhang, and Jinqiao Wang, "High-speed tracking with multi-kernel correlation filters," in *CVPR*, 2018, pp. 4874–4883.

[6] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang, "Multi-task correlation particle filter for robust object tracking," in *CVPR*. IEEE, 2017, pp. 4819–4827.

[7] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV*, 2015, pp. 4310–4318.

[8] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 9, pp. 1834–1848, 2015.

[9] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[10] Linyu Zheng, Ming Tang, and Jinqiao Wang, "Learning robust gaussian process regression for visual tracking.," in *IJCAI*, 2018, pp. 1219–1225.

[11] Matthias Mueller, Neil Smith, and Bernard Ghanem, "Context-aware correlation filter tracking," in *CVPR*, 2017, pp. 1387–1395.

[12] Heng Fan and Jinhai Xiang, "Robust visual tracking via local-global correlation filter.," in *AAAI*, 2017, pp. 4025–4031.

[13] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Robust visual tracking via hierarchical convolutional features," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[14] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *ECCV*. Springer, 2016, pp. 472–488.

[15] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip H S Torr, "Fully-convolutional siamese networks for object tracking," 2016, pp. 850–865.

[16] Chen Huang, Simon Lucey, and Deva Ramanan, "Learning policies for adaptive tracking with deep feature cascades," *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 105–114, 2017.

[17] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang, "Learning dynamic siamese network for visual object tracking," in *ICCV*. IEEE, 2017, pp. 1781–1789.

[18] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu, "High performance visual tracking with siamese region proposal network," in *CVPR*, 2018, pp. 8971–8980.

[19] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson Lau, and Ming-Hsuan Yang, "Vital: Visual tracking via adversarial learning," *arXiv preprint arXiv:1804.04273*, 2018.

[20] Pengpeng Liang, Erik Blasch, and Haibin Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

[21] M. Kristan and et al., "The visual object tracking vot2016 challenge results," in *ECCVW*, 2016.

[22] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al., "The sixth visual object tracking vot2018 challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[23] Matthias Mueller, Neil Smith, and Bernard Ghanem, "A benchmark and simulator for uav tracking," in *European conference on computer vision*. Springer, 2016, pp. 445–461.

[24] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing, "Transfer learning based visual tracking with gaussian processes regression," in *ECCV*. Springer, 2014, pp. 188–203.

[25] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking," *arXiv preprint arXiv:1807.11348*, 2018.

[26] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016, pp. 4293–4302.

[27] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng, "A twofold siamese network for real-time object tracking," in *CVPR*, 2018, pp. 4834–4843.

[28] Jongwon Choi, Hyung Jin Chang, Sangdoo Yun, Tobias Fischer, Yiannis Demiris, Jin Young Choi, et al., "Attentional correlation filter network for adaptive visual tracking.," in *CVPR*, 2017, vol. 2, p. 7.

[29] Xue Mei, Zhibin Hong, Danil Prokhorov, and Dacheng Tao, "Robust multitask multiview tracking in videos," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 11, pp. 2874–2890, 2015.

[30] Shunli Zhang, Yao Sui, Sicong Zhao, Xin Yu, and Li Zhang, "Multi-local-task learning with global regularization for object tracking," *Pattern Recognition*, vol. 48, no. 12, pp. 3881–3894, 2015.

[31] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja, "Robust visual tracking via multi-task sparse learning," in *CVPR*. IEEE, 2012, pp. 2042–2049.

[32] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang, "Deep regression tracking with shrinkage loss," in *ECCV*, 2018, pp. 353–369.

[33] Changhong Fu, Xiaoxiao Yang, Fan Li, Juntao Xu, Changjing Liu, and Peng Lu, "Learning consistency pursued correlation filters for real-time uav tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1–8.

[34] Yiming Li, Changhong Fu, Ziyuan Huang, Yinqiang Zhang, and Jia Pan, "Keyfilter-aware real-time uav object tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1–7.

[35] Ziyuan Huang, Changhong Fu, Yiming Li, Fuling Lin, and Peng Lu, "Learning aberrance repressed correlation filters for real-time uav tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2891–2900.