Robust Visual Tracking using Multi-Frame Multi-Feature Joint Modeling

Peng Zhang*, Shujian Yu*, *Student Member, IEEE,* Jiamiao Xu, Xinge You†, *Senior Member, IEEE,* Xiubao Jiang, Xiao-Yuan Jing, and Dacheng Tao, *Fellow, IEEE.*

***Abstract*—It remains a huge challenge to design effective and efficient trackers under complex scenarios, including occlusions, illumination changes and pose variations. To cope with this problem, a promising solution is to integrate the temporal consistency across consecutive frames and multiple feature cues in a unified model. Motivated by this idea, we propose a novel correlation filter-based tracker in this work, in which the temporal relatedness is reconciled under a multi-task learning framework and the multiple feature cues are modeled using a multi-view learning approach. We demonstrate the resulting regression model can be efficiently learned by exploiting the structure of *blockwise diagonal matrix*. A fast *blockwise diagonal matrix* inversion algorithm is developed thereafter for efficient online tracking. Meanwhile, we incorporate an adaptive scale estimation mechanism to strengthen the stability of scale variation tracking. We implement our tracker using two types of features and test it on two benchmark datasets. Experimental results demonstrate the superiority of our proposed approach when compared with other state-of-the-art trackers. MATLAB code is available from our project homepage http://bmal.hust.edu.cn/project/KMF2JMTtracking.html.**

***Index Terms*—Visual Tracking, Multi-task Learning, Multi-view Learning, Blockwise Diagonal Matrix, Correlation Filters.**

I. Introduction

VISUAL tracking is one of the most important components in computer vision system. It has been widely used in visual surveillance, human computer interaction, and robotics [1], [2]. Given an annotation of the object (bounding box) in the first frame, the task of visual tracking is to estimate the target locations and scales in subsequent video frames. Though much progress has been made in recent years, robust visual tracking, which can reconcile different varying circumstances, still remains a challenging problem [1]. On the one hand, it is expected that the designed trackers can compensate for large appearance changes caused by illuminations, occlusions, etc. On the other hand, the real-time requirement in real applications impedes the usage of overcomplicated models.

Briefly speaking, there are two major categories of modern trackers: generative trackers and discriminative trackers [1], [3]. Generative trackers typically assume a generative process of the target appearance and search for the regions most similar to the target model, while discriminative trackers usually train a classifier to distinguish the target from the background. Among discriminative trackers, correlation filter-based trackers (CFTs) drawn an increasing number of attentions since the development of Kernel Correlation Filter (KCF) tracker [4]. As has been demonstrated, KCF can achieve impressive performance on accuracy, robustness and speed on both the Online Tracking Benchmark (OTB) [5] and the Visual Object Tracking (VOT) challenges [6].

Despite the overwhelming evidence of success achieved by CFTs, two observations prompt us to come up with our tracking approach. First, almost all the CFTs ignore the temporal consistency or invariance among consecutive frames, which has been demonstrated to be effective in augmenting tracking performance [7], [8]. Second, there is still a lack of theoretical sound yet computational efficient model to integrate multiple feature cues. Admittedly, integrating different channel features is not new under the CFTs umbrella. However, previous work either straightforwardly concatenating various feature vectors [9], [10] (i.e., assuming mutual independence of feature channels) or inheriting high computational burden which severely compromises the tracking speed [11], [12].

In this paper, to circumvent these two drawbacks simultaneously, we embark on the basic KCF tracker and present a multi-frame multi-feature joint modeling tracker ($(\text{MF})^2$JMT) to strike a good trade-off between robustness and speed. In $(\text{MF})^2$JMT, the interdependencies between different feature cues are modeled using a multi-view learning (MVL) approach to enhance the discriminative power of target appearance undergoing various changes. Specifically, we use the view consistency principle to regularize the objective function learned from each view agree on the labels of most training samples [13]. On the other hand, the temporal consistency is exploited under a multi-task learning (MTL) framework [14], i.e., we model M ($M \geq 2$) consecutive frames simultaneously by constraining the learned objectives from each frame close to their mean. We extend $(\text{MF})^2$JMT to its kernelized version (i.e., K$(\text{MF})^2$JMT) and develop a fast *blockwise diagonal matrix* inversion algorithm to accelerate model training. Finally, we adopt a two-stage filtering pipeline [9], [15], [16] to cope with the problem of scale variations.

*The first two authors contributed equally to this work and should be regarded as co-first authors.

†Corresponding author.

P. Zhang, J. Xu, X. You and X. Jiang are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei, China. e-mail: youxg@mail.hust.edu.cn.

S. Yu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA. e-mail: yusjlcy9011@ufl.edu.

X.-Y. Jing is with the State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China and also with the School of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210023, China. e-mail: jingxy_2000@126.com.

D. Tao is with the UBTech Sydney Artificial Intelligence Institute and the School of Information Technologies, in the Faculty of Engineering and Information Technologies at The University of Sydney, Darlington, NSW 2008, Australia. e-mail: dacheng.tao@sydney.edu.au.

To summarize, the main contributions of our work are twofold. First, a novel tracker, which can integrate multiple feature cues and temporal consistency in a unified model, is developed for robust visual tracking. Specifically, instead of simply concatenating multiple features into a single vector, we demonstrate how to reorganize these features by taking into consideration their intercorrelations. Moreover, we also present an advanced way to reconcile temporal relatedness amongst multiple consecutive frames, rather than naively using a forgetting factor [10], [17]. Second, a fast *blockwise diagonal matrix* inversion algorithm is developed to speed up training and detection. Experiments against state-of-the-art trackers reveal the underline clues on the importance of "intelligent" feature integration and temporal modeling, and also illustrate future directions for the design of modern discriminative trackers.

The rest of this paper is organized as follows. In section II, we introduce the background knowledge. Then, in section III, we discuss the detailed methodology of $(\text{MF})^2$JMT and extend it under kernel setting. A fast algorithm for *blockwise diagonal matrix* inversion is also developed to speed up training and detection. Following this, an adaptive scale estimation mechanism is incorporated into our tracker in section IV. The performance of our proposed approach against other state-of-the-art trackers is evaluated in section V. This paper is concluded in section VI.

$Notation$: Scalars are denoted by lowercase letters (e.g., x), vectors appear as lowercase boldface letters (e.g., $\mathbf{x}$), and matrices are indicated by uppercase letters (e.g., X or $\mathbf{X}$). The (i,j)-th element of X (or $\mathbf{X}$) is represented by X_{ij} (or $\mathbf{X}_{ij}$), while $[\cdot]^T$ denotes transpose and $[\cdot]^H$ denotes Hermitian transpose. If X (or $\mathbf{X}$) is a square matrix, then X^{-1} (or $\mathbf{X}^{-1}$) denotes its inverse. $\mathbf{I}$ stands for the identity matrix with compatible dimensions, $diag(\mathbf{x})$ denotes a square diagonal matrix with the elements of vector $\mathbf{x}$ on the main diagonal. The i-th row of a matrix X (or $\mathbf{X}$) is declared by the row vector $\mathbf{x}^i$, while the j-th column is indicated with the column vector $\mathbf{x}_j$. If $|\cdot|$ denotes the absolute value operator, then, for $\mathbf{x} \in \mathbb{R}^n$, the ℓ_1-norm and ℓ_2-norm of $\mathbf{x}$ are defined as $\|\mathbf{x}\|_1 \triangleq \sum_{i=1}^{n} |x_i|$ and $\|\mathbf{x}\|_2 \triangleq \sqrt{\sum_{i=1}^{n} x_i^2}$, respectively.

II. Related work

In this section, we briefly review previous work of most relevance to our approach, including popular conventional trackers, the basic KCF tracker and its extensions.

A. Popular conventional trackers

Success in visual tracking relies heavily on how discriminative and robust the representation of target appearance is against varying circumstances [18], [19]. This is especially important for discriminative trackers, in which a binary classifier is required to distinguish target from background. Numerous types of visual features have been successfully applied for discriminative trackers in the last decades, including color histograms [20], [21], texture features [22], Haar-like features [23], etc. Unfortunately, none of them can handle all kinds of varying circumstances individually and the discriminant capability of a unique type of feature is not stable across the video sequence [20]. As a result, it becomes prevalent to take advantage of multiple features (i.e., multi-view representations) to enable more robust tracking. For example, [24], [25] used the AdaBoost to combine an ensemble of weak classifiers to form a powerful classifier, where each weak classifier is trained online on a different training set using pixel colors and a local orientation histogram. Note that, several generative trackers also have improved performance by incorporating multi-view representations. [26] employed a group sparsity technique to integrate color histograms, intensity, histograms of oriented gradients (HOGs) [27] and local binary patterns (LBPs) [28] via requiring these features to share the same subset of the templates, whereas [29] proposed a probabilistic approach to integrate HOGs, intensity and Haar-like features for robust tracking. These trackers perform well normally, but they are far from satisfactory when being tested on challenging videos.

B. The basic KCF tracker and its extensions

Much like other discriminative trackers, the KCF tracker needs a set of training examples to learn a classifier. The key idea of KCF tracker is that the augmentation of negative samples is employed to enhance the discriminative ability of the tracking-by-detection scheme while exploring the structure of the circulant matrix to speed up training and detection.

Following the basic KCF tracker, numerous extensions have been conducted to boost its performance, which generally fall into two categories: application of improved features and conceptual improvements in filter learning [30]. The first category lies in designing more discriminative features to deal with challenging environments [31], [32] or straightforwardly concatenating multiple feature cues into a single feature vector to boost representation power [9], [33]. A recent trend is to integrate features extracted from convolutional neural networks (CNNs) trained on large image datasets to replace the traditional hand-crafted features [34], [35]. The design or integration of more powerful features often suffers from high computational burden, whereas blindly concatenating multiple features assumes the mutual independence of these features and neglects their interrelationships. Therefore, it is desirable to have a more "intelligent" manner to reorganize these features such that their interrelationships are fully exploited.

Conceptually, the theoretical extension on the filter learning also drawn lots of attentions. Early work focused on accurate and efficient scale estimation as the basic KCF assumes a fixed target size [11], [15]. The most recent work concentrated on developing more advanced convolutional operators. For example, [36] employed an interpolation model to train convolution filters in continuous spatial domain. The so-called Continuous Convolution Operator Tracker (C-COT) is further improved in [37], in which the authors introduce factorized convolution operators to drastically reduce the number of parameters C-COT as well as the number of filters.

Apart from these two extensions, efforts have been made to embed the conventional tracking strategies (like part-based tracking [38]) on KCF framework. For example, [39] developed a probabilistic tracking reliability metric to measure

how reliable a patch can be tracked. On the other hand, [40] employed an online random fern classifier as a re-detection component for long-term tracking, whereas [16] presented a biology-inspired framework where short-term processing and long-term processing are cooperated with each other under a correlation filter framework. Finally, it is worth noting that, with the rapid development of correlation filters on visual tracking, several correlation filter-based thermal infrared trackers (e.g., [41]–[43]) have been developed in recent years. This work only focuses on visual tracking on color video sequences. We leave extension to thermal infrared video sequences as future work.

III. The Multi-Frame Multi-Feature Joint Modeling Tracker ((MF)^{2}JMT)

The (MF)^{2}JMT made two strategic extensions to the basic KCF tracker. Our idea is to integrate the temporal information and multiple feature cues in a unified model, thus providing a theoretical sound and computational-efficient solution for robust visual tracking. To this end, instead of simply concatenating different features, the (MF)^{2}JMT integrates multiple feature cues using a MVL approach to better exploit their interrelationships, thus forming a more informative representation to target appearance. Moreover, the temporal consistency is taken into account under a MTL framework. Specifically, different from prevalent CFTs that learn filter taps using a ridge regression function which only makes use of template (i.e. circulant matrix $\mathbf{X}$) from the current frame, we show that it is possible to use examples from M ($M \geq 2$) consecutive frames to learn the filter taps very efficiently by exploiting the structure of *blockwise diagonal matrix*.

Before our work, there are two ways attempting to incorporate temporal consistency into KCF framework. The Spatio-Temporal Context (STC) tracker [10] and its extensions (e.g., [17]) formulate the spatial relationships between the target and its surrounding dense contexts in a Bayesian framework and then use a temporal filtering procedure together with a forgetting factor to update the spatio-temporal model. Despite its simplicity, the tracking accuracy of STC tracker is poor compared with other state-of-the-art KCF trackers (see Section V-F). On the other hand, another trend is to learn a temporally invariant feature representation trained on natural video repository for visual tracking [31]. Although the learned feature can accommodate partial occlusion or slight illumination variation, it cannot handle the scenarios where there are large appearance changes. Moreover, the effectiveness of trained features depends largely on the selected repository which suffers from limited generalization capability. Different from these two kinds of methods, we explicitly model multiple consecutive frames in a joint cost function to circumvent abrupt drift in a single frame and to reconcile temporal relatedness amongst frames in a short period of time. Experiments demonstrate the superiority of our method.

A. Formulation of (MF)^{2}JMT

We start from the formulation of the basic (MF)^{2}JMT. Given M training frames, we assume the number of candidate patches in the t-th ($t = 1, 2, \cdots, M$) frame is n. Suppose the dimensions for the first and second types of features are p and q respectively[1], we denote $X_t \in \mathbb{R}^{n\times p}$ ($Z_t \in \mathbb{R}^{n\times q}$) the matrix consists of the first (second) type of feature in the t-th frame (each row represents one sample). Also, let $\mathbf{y}_t \in \mathbb{R}^{n\times 1}$ represent sample labels. Then, the objective of (MF)^{2}JMT can thus be formulated as:

$$\begin{aligned}\min_{\mathbf{w}_0,\mathbf{p}_t,\mathbf{v}_0,\mathbf{q}_t} \quad \mathcal{J} \quad = \quad & \sum_{t=1}^{M}\Big(\|\mathbf{y}_t - X_t\mathbf{w}_t\|_2^2 + \lambda_1 \|\mathbf{y}_t - Z_t\mathbf{v}_t\|_2^2 \\ & +\lambda_2\|X_t\mathbf{w}_t - Z_t\mathbf{v}_t\|_2^2\Big) + \frac{\gamma_1}{M}\sum_{t=1}^{M}\|\mathbf{p}_t\|_2^2 \\ & +\gamma_2\|\mathbf{w}_0\|_2^2 + \frac{\eta_1}{M}\sum_{t=1}^{M}\|\mathbf{q}_t\|_2^2 + \eta_2\|\mathbf{v}_0\|_2^2, \\ s.t. \quad & \mathbf{w}_t = \mathbf{w}_0 + \mathbf{p}_t \\ & \mathbf{v}_t = \mathbf{v}_0 + \mathbf{q}_t \end{aligned} \tag{1}$$

where $\lambda_1, \lambda_2, \gamma_1, \gamma_2, \eta_1, \eta_2$ are the non-negative regularization parameters controlling model complexity. $\mathbf{w}_0$ is the regression coefficients shared by M frames for the first type of feature and $\mathbf{p}_t$ denotes the deviation term from $\mathbf{w}_0$ in the t-th frame. The same definition goes for $\mathbf{v}_0$ and $\mathbf{q}_t$ for the second type of feature.

The problem (1) contains three different items with distinct objectives, namely the MTL item, the MVL item and the regularization item. The MTL item, i.e., $\sum_{t=1}^{M}\|\boldsymbol{y}_t - X_t\mathbf{w}_t\|_2^2 + \frac{\gamma_1}{M}\sum_{t=1}^{M}\|\mathbf{p}_t\|_2^2$ (or $\sum_{t=1}^{M}\|\boldsymbol{y}_t - Z_t\mathbf{v}_t\|_2^2 + \frac{\eta_1}{M}\sum_{t=1}^{M}\|\mathbf{q}_t\|_2^2$), is analogous to the formulation of regularized multi-task learning (RMTL) [14], as it encourages $\mathbf{w}_t$ (or $\mathbf{v}_t$) close to the mean value $\mathbf{w}_0$ (or $\mathbf{v}_0$) with a small deviation $\mathbf{p}_t$ (or $\mathbf{q}_t$). The MVL item $\|\mathbf{y}_t - X_t\mathbf{w}_t\|_2^2 + \lambda_1\|\mathbf{y}_t - Z_t\mathbf{v}_t\|_2^2 + \lambda_2\|X_t\mathbf{w}_t - Z_t\mathbf{v}_t\|_2^2$ employs the view consistency principle that widely exists in MVL approaches (e.g., [44]) to constrain the objectives learned from two views agree on most training samples. Finally, the regularization term $\|\mathbf{w}_0\|_2^2$ (or $\|\mathbf{v}_0\|_2^2$) serves to prevent the ill-posed solution and enhance the robustness of selected features to noises or outliers.

B. Solution to (MF)^{2}JMT

Minimization of $\mathcal{J}$ has a closed-form solution by equating the gradients of (1) w.r.t. $\mathbf{w}_0, \mathbf{p}_t, \mathbf{v}_0$ and $\mathbf{q}_t$ to zero. Albeit its simplicity, reorganizing linear equation array into standard form is intractable and computational expensive herein. As an alternative, we present an equivalent yet simpler form of (1), which can be solved with matrix inversion in one step.

[1]This paper only considers two types of features, but the developed objective function (1) and associated solution can be straightforwardly extended to three or more feature cues.

Denote

$$\mu_1 = \frac{M\gamma_2}{\gamma_1}, \quad (2)$$

$$\mu_2 = \frac{M\eta_2}{\eta_1}, \quad (3)$$

$$\overline{X}_t = \Big(\frac{X_t}{\sqrt{\mu_1}}, \underbrace{\mathbf{0}, \cdots, \mathbf{0}}_{t-1}, X_t, \underbrace{\mathbf{0}, \cdots, \mathbf{0}}_{M-t}\Big), \quad (4)$$

$$\overline{Z}_t = \Big(\frac{Z_t}{\sqrt{\mu_2}}, \underbrace{\mathbf{0}, \cdots, \mathbf{0}}_{t-1}, Z_t, \underbrace{\mathbf{0}, \cdots, \mathbf{0}}_{M-t}\Big), \quad (5)$$

$$\mathbf{w} = \begin{pmatrix} \sqrt{\mu_1}\mathbf{w}_0 \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_M \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} \sqrt{\mu_2}\mathbf{v}_0 \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_M \end{pmatrix}, \quad (6)$$

where $\mathbf{0}$ denotes a zero matrix of the same size as X_t in $\overline{X}_t$ (or Z_t in $\overline{Z}_t$), we have:

$$\overline{X}_t\mathbf{w} = X_t\mathbf{w}_0 + X_t\mathbf{p}_t = X_t\mathbf{w}_t, \quad (7)$$

$$\overline{Z}_t\mathbf{v} = Z_t\mathbf{v}_0 + Z_t\mathbf{q}_t = Z_t\mathbf{v}_t, \quad (8)$$

$$\|\mathbf{w}\|_2^2 = \sum_{t=1}^{M}\|\mathbf{p}_t\|_2^2 + \mu_1\|\mathbf{w}_0\|_2^2 = \sum_{t=1}^{M}\|\mathbf{p}_t\|_2^2 + \frac{M\gamma_2}{\gamma_1}\|\mathbf{w}_0\|_2^2, \quad (9)$$

$$\|\mathbf{v}\|_2^2 = \sum_{t=1}^{M}\|\mathbf{q}_t\|_2^2 + \mu_2\|\mathbf{v}_0\|_2^2 = \sum_{t=1}^{M}\|\mathbf{q}_t\|_2^2 + \frac{M\eta_2}{\eta_1}\|\mathbf{v}_0\|_2^2. \quad (10)$$

Substituting (7)-(10) into (1) yields:

$$\min_{\mathbf{w},\mathbf{v}} \mathcal{J} = \sum_{t=1}^{M}\Big(\big\|\mathbf{y}_t - \overline{X}_t\mathbf{w}\big\|_2^2 + \lambda_1\big\|\boldsymbol{y}_t - \overline{Z}_t\mathbf{v}\big\|_2^2 + \lambda_2\|\overline{X}_t\mathbf{w} - \overline{Z}_t\mathbf{v}\|_2^2\Big) + \frac{\gamma_1}{M}\|\mathbf{w}\|_2^2 + \frac{\eta_1}{M}\|\mathbf{v}\|_2^2. \quad (11)$$

Denote

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_M \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \overline{X}_1 \\ \vdots \\ \overline{X}_M \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} \overline{Z}_1 \\ \vdots \\ \overline{Z}_M \end{pmatrix}, \quad (12)$$

then (11) becomes:

$$\min_{\mathbf{w},\mathbf{v}} \mathcal{J} = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1\|\mathbf{y} - \mathbf{Z}\mathbf{v}\|_2^2 + \lambda_2\|\mathbf{X}\mathbf{w} - \mathbf{Z}\mathbf{v}\|_2^2 + \frac{\gamma_1}{M}\|\mathbf{w}\|_2^2 + \frac{\eta_1}{M}\|\mathbf{v}\|_2^2. \quad (13)$$

Equating the gradients of $\mathcal{J}$ w.r.t. $\mathbf{w}$ and $\mathbf{v}$ to zero. With straightforward derivation, we have:

$$\begin{pmatrix} (1+\lambda_2)\mathbf{X}^T\mathbf{X} + \frac{\gamma_1}{M}\mathbf{I} & -\lambda_2\mathbf{X}^T\mathbf{Z} \\ -\lambda_2\mathbf{Z}^T\mathbf{X} & (\lambda_1+\lambda_2)\mathbf{Z}^T\mathbf{Z} + \frac{\eta_1}{M}\mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{w} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{X}^T\mathbf{y} \\ \lambda_1\mathbf{Z}^T\mathbf{y} \end{pmatrix}. \quad (14)$$

Denote $\boldsymbol{\xi} \doteq (\mathbf{w}, \mathbf{v})^T$, the solution of $\mathcal{J}$ is given by [45]:

$$\boldsymbol{\xi} = \Big(\mathbf{U}^T\mathbf{D}\mathbf{U} + \mathbf{R}\Big)^{-1}\mathbf{U}^T\mathbf{Y} = \mathbf{R}^{-1}\mathbf{U}^T\Big(\mathbf{U}\mathbf{R}^{-1}\mathbf{U}^T + \mathbf{D}^{-1}\Big)^{-1}\mathbf{D}^{-1}\mathbf{Y}, \quad (15)$$

where

$$\mathbf{D} = \begin{pmatrix} (1+\lambda_2)\mathbf{I} & -\lambda_2\mathbf{I} \\ -\lambda_2\mathbf{I} & (\lambda_1+\lambda_2)\mathbf{I} \end{pmatrix}, \mathbf{U} = \begin{pmatrix} \mathbf{X} & 0 \\ 0 & \mathbf{Z} \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \frac{\gamma_1}{M}\mathbf{I} & 0 \\ 0 & \frac{\eta_1}{M}\mathbf{I} \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} \mathbf{y} \\ \lambda_1\mathbf{y} \end{pmatrix}. \quad (16)$$

Having computed $\mathbf{w}_t$ and $\mathbf{v}_t$, the responses of candidate samples $\mathbf{z}$ in the next frame for the trained $(\mathrm{MF})^2$JMT model can be computed as:

$$f(\mathbf{z}) = \mathbf{U}^{new}\boldsymbol{\xi} = \mathbf{U}^{new}\mathbf{R}^{-1}\mathbf{U}^T\Big(\mathbf{U}\mathbf{R}^{-1}\mathbf{U}^T + \mathbf{D}^{-1}\Big)^{-1}\mathbf{D}^{-1}\mathbf{Y}, \quad (17)$$

where

$$\mathbf{U}^{new} = \begin{pmatrix} \mathbf{X}^{new} & 0 \\ 0 & \mathbf{Z}^{new} \end{pmatrix}, \quad (18)$$

in which $\mathbf{X}^{new}$ and $\mathbf{Z}^{new}$ are feature matrices constructed from features in the new frame. Specifically, let $\mathbf{X}^{new} = (\overline{X}_1^{new}, ..., \overline{X}_M^{new})^T \in \mathbb{R}^{Mn\times(M+1)p}$ consist of the first type of feature, we construct $\overline{X}_M^{new}$ with feature in the new frame as defined in (4) and set $\overline{X}_1^{new}, ..., \overline{X}_{M-1}^{new}$ as zero matrices to coincide with the size of $\mathbf{X}$ in (12). In this sense, only $\mathbf{w}_M$ and $\mathbf{v}_M$ contribute to $f(\mathbf{z})$. The same goes for $\mathbf{Z}^{new}$ and $\overline{Z}_t^{new}$ $(t = 1, ..., M)$.

C. Kernel extension and fast implementation

Although (15) gives a tractable solution to (1), it contains the inversion of $\mathbf{U}^T\mathbf{D}\mathbf{U} + \mathbf{R}$ with the computational complexity $\mathcal{O}(n^3)$ when using the well acknowledged Gaussian elimination algorithm [46]. In this section, for a more powerful regression function and a fast implementation, we demonstrate how to incorporate the dense sampling strategy in the basic $(\mathrm{MF})^2$JMT under the kernel setting to speed up its tracking and detection. A computational-efficient optimization method is also presented by exploiting the structure of *blockwise diagonal matrix*.

The dense sampling was considered previously to be a drawback for discriminative trackers because of the large number of redundant samples that are required [3]. However, when these samples are collected and organized properly, they form a circulant matrix that can be diagonalized efficiently using the DFT matrix, thereby making the dual rigid regression problem can be solved entirely in the frequency domain [4]. Due to this attractive property, we first show that it is easy to embed the "kernel trick" on $(\mathrm{MF})^2$JMT, we will also show that it is possible to obtain non-linear filters as fast as linear correlation filters using the dense sampling, both to train and evaluate. We term this improvement kernel multi-frame multi-feature joint modeling tracker (K$(\mathrm{MF})^2$JMT).

Denote

$$\begin{aligned}\mathbf{K}^{xx^{new}} &\doteq \mathbf{X}^{new}\mathbf{X}^T,\\ \mathbf{K}^{zz^{new}} &\doteq \mathbf{Z}^{new}\mathbf{Z}^T,\\ \mathbf{K}^{xx} &\doteq \mathbf{X}\mathbf{X}^T,\\ \mathbf{K}^{zz} &\doteq \mathbf{Z}\mathbf{Z}^T,\end{aligned} \tag{19}$$

we have

$$\mathbf{U}^{new}\mathbf{R}^{-1}\mathbf{U}^T = \begin{pmatrix} \frac{M}{\gamma_1}\mathbf{K}^{xx^{new}} & 0 \\ 0 & \frac{M}{\eta_1}\mathbf{K}^{zz^{new}} \end{pmatrix}, \tag{20}$$

and

$$\mathbf{U}\mathbf{R}^{-1}\mathbf{U}^T+\mathbf{D}^{-1} = \begin{pmatrix} \left(\frac{M}{\gamma_1}\mathbf{K}^{xx}+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I}\right) & \left(\tau^{-1}\lambda_2\mathbf{I}\right) \\ \left(\tau^{-1}\lambda_2\mathbf{I}\right) & \left(\frac{M}{\eta_1}\mathbf{K}^{zz}+\tau^{-1}(1+\lambda_2)\mathbf{I}\right) \end{pmatrix}, \tag{21}$$

where $\tau = (1+\lambda_2)(\lambda_1+\lambda_2) - \lambda_2^2 = \lambda_1+\lambda_2+\lambda_1\lambda_2$.

According to (12), $\mathbf{K}^{xx^{new}}$ consists of $M \times M$ block matrices and the (i,j)-th $(i,j=1,...,M)$ block matrix can be represented as:

$$\mathbf{K}_{ij}^{xx^{new}} = \overline{X}_i^{new}\overline{X}_j^T = (\frac{1}{\mu_1}+\delta_{ij})X_i^{new}X_j^T, \tag{22}$$

where $\delta_{ij} = \mathbf{1}_{\{i=j\}}$ with $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function.

If we project $\mathbf{X}$ and $\mathbf{Z}$ onto the Reproducing Kernel Hilbert Space (RKHS), i.e., applying a non-linear transform ϕ to both $\mathbf{X}$ and $\mathbf{Z}$, we can obtain the kernelized version of (1), i.e., K(MF)^{2}JMT. According to [4], (22) can be represented as:

$$\mathbf{K}_{ij}^{xx^{new}} = \left(\frac{1}{\mu_1}+\delta_{ij}\right)C(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i^{new}}), \tag{23}$$

where $C(\mathbf{x})$ denotes a circular matrix generated by $\mathbf{x}$ (see Appendix A for more details). Similarly,

$$\begin{aligned}\mathbf{K}_{ij}^{xx} &= \left(\frac{1}{\mu_1}+\delta_{ij}\right)C(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i}), &(24)\\ \mathbf{K}_{ij}^{zz^{new}} &= \left(\frac{1}{\mu_2}+\delta_{ij}\right)C(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i^{new}}), &(25)\\ \mathbf{K}_{ij}^{zz} &= \left(\frac{1}{\mu_2}+\delta_{ij}\right)C(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i}). &(26)\end{aligned}$$

Note that $C(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i^{new}})$, $C(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i})$, $C(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i^{new}})$ and $C(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i})$ are circular matrices, thus can be made diagonal as expressed below [47]:

$$\begin{aligned}C(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i^{new}}) &= F diag(\hat{\mathbf{k}}^{\mathbf{x}_i\mathbf{x}_j^{new}})F^H,\\ C(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i}) &= F diag(\hat{\mathbf{k}}^{\mathbf{x}_i\mathbf{x}_j})F^H,\\ C(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i^{new}}) &= F diag(\hat{\mathbf{k}}^{\mathbf{z}_j\mathbf{z}_i^{new}})F^H,\\ C(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i}) &= F diag(\hat{\mathbf{k}}^{\mathbf{z}_i\mathbf{z}_j})F^H.\end{aligned} \tag{27}$$

where F is the DFT matrix. Combining (20)-(27), according to Appendix A, the $f(\mathbf{z})$ (defined in (17)) under kernel setting can be computed as:

$$\begin{aligned}f(\mathbf{z}) &= \mathbf{U}^{new}\mathbf{R}^{-1}\mathbf{U}^T\left(\mathbf{U}\mathbf{R}^{-1}\mathbf{U}^T+\mathbf{D}^{-1}\right)^{-1}\mathbf{D}^{-1}\mathbf{Y}\\ &= \mathbf{F}\Omega_1\mathbf{F}^H\left(\mathbf{F}\Omega_2\mathbf{F}^H\right)^{-1}\mathbf{D}^{-1}\mathbf{Y}\\ &= \mathbf{F}\Omega_1\Omega_2^{-1}\mathbf{F}^H\mathbf{D}^{-1}\mathbf{Y},\end{aligned} \tag{28}$$

where Ω_1 and Ω_2 are given in (29) and $\mathbf{F}$ is a *block diagonal matrix* with blocks F on its main diagonal.

Denote

$$\begin{aligned}L^x &= \frac{M}{\gamma_1}\left(\left(\frac{1}{\mu_1}+\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i})\right)_{i,j=1}^{M},\\ L^z &= \frac{M}{\eta_1}\left(\left(\frac{1}{\mu_2}+\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i})\right)_{i,j=1}^{M},\end{aligned} \tag{30}$$

then

$$\Omega_2 = \begin{pmatrix} L^x+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I} & \left(\tau^{-1}\lambda_2\mathbf{I}\right) \\ \left(\tau^{-1}\lambda_2\mathbf{I}\right) & L^z+\tau^{-1}(1+\lambda_2)\mathbf{I} \end{pmatrix}. \tag{31}$$

According to the Equation (2.76) of [45], we have:

$$\Omega_2^{-1} = \begin{pmatrix} \left(L^z+\tau^{-1}(1+\lambda_2)\mathbf{I}\right)A^{-1} & \left(-\tau^{-1}\lambda_2\mathbf{I}\right)B^{-1} \\ \left(-\tau^{-1}\lambda_2\mathbf{I}\right)A^{-1} & \left(L^x+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I}\right)B^{-1} \end{pmatrix}, \tag{32}$$

where

$$\begin{aligned}A &= \left(L^x+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I}\right)\left(L^z+\tau^{-1}(1+\lambda_2)\mathbf{I}\right)-\tau^{-2}\lambda_2^2\mathbf{I},\\ B &= \left(L^z+\tau^{-1}(1+\lambda_2)\mathbf{I}\right)\left(L^x+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I}\right)-\tau^{-2}\lambda_2^2\mathbf{I}\\ &= A^T.\end{aligned} \tag{33}$$

It is obvious that A and B are $Mn \times Mn$ *blockwise diagonal matrix* that can be partitioned into $M \times M$ diagonal matrices of size $n \times n$. According to *Theorem 1*, the computational cost for A^{-1} or B^{-1} is $\mathcal{O}(nM^3)$. Besides, it takes nM^3 product operations to compute $L^x \times L^z$, $\left(L^z+\tau^{-1}(1+\lambda_2)\mathbf{I}\right) \times A^{-1}$ and $\left(L^x+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I}\right) \times B^{-1}$. In this sense, the computational cost for Ω_2^{-1} is still $\mathcal{O}(nM^3)$. On the other hand, the DFT bounds the cost at nearly $\mathcal{O}(n\log n)$ by exploiting the circulant structure [4]. Therefore, the overall cost for computing $f(\mathbf{z})$ is $\mathcal{O}(Mn\log n+nM^3)$ given that there are $2M$ inverse DFTs in (28).

Theorem 1: Given an invertible *blockwise diagonal matrix* $\mathbf{S}$ that can be partitioned into $M \times M$ diagonal matrices of size $n \times n$, the computational complexity of $\mathbf{S}^{-1}$ is $\mathcal{O}(nM^3)$.

Proof: Denote S_{ij} $(i,j=1,...,M)$ the (i,j)-th block of $\mathbf{S}$, where S_{ij} is a diagonal matrix of size $n \times n$. After a series of elementary matrix operations, e.g., pre-multiply or post-multiply the matrix $\mathbf{S}$ by different elementary matrices, we can interchange the rows and columns of $\mathbf{S}$ arbitrarily. Therefore, there exists an invertible matrix $\mathbf{P}$ with $\mathbf{P}\mathbf{P}^T = \mathbf{I}$, such that the matrix $\tilde{\mathbf{S}} = \mathbf{P}\mathbf{S}\mathbf{P}^T \triangleq (\tilde{S}_{\tilde{i}\tilde{j}})_{\tilde{i},\tilde{j}=1,\cdots,n}$ satisfies: the elements of $\tilde{S}_{\tilde{i}\tilde{j}}$ come from $\mathbf{S}$ with row indices $(\tilde{i}, n+\tilde{i}, 2n+\tilde{i}, \cdots, (M-1)n+\tilde{i})$ and column indices $(\tilde{j}, n+\tilde{j}, 2n+\tilde{j}, \cdots, (M-1)n+\tilde{j})$. Obviously, $\tilde{S}_{\tilde{i}\tilde{j}} = \mathbf{0}$ for $\tilde{i} \neq \tilde{j}$. Thus, $\tilde{\mathbf{S}}^{-1}$ can be represented as:

$$\tilde{\mathbf{S}}^{-1} = \begin{pmatrix} \tilde{S}_{11}^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \tilde{S}_{22}^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \tilde{S}_{nn}^{-1} \end{pmatrix}. \tag{34}$$

$$
\begin{aligned}
\Omega_1 &= \begin{pmatrix} \frac{M}{\gamma_1}\left(\left(\frac{1}{\mu_1}+\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i^{new}})\right)_{i,j=1}^{M} & 0 \\ 0 & \frac{M}{\eta_1}\left(\left(\frac{1}{\mu_2}+\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i^{new}})\right)_{i,j=1}^{M} \end{pmatrix} \\
&= \begin{pmatrix} \left(\left(\frac{1}{\gamma_2}+\frac{M}{\gamma_1}\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i^{new}})\right)_{i,j=1}^{M} & 0 \\ 0 & \left(\left(\frac{1}{\eta_2}+\frac{M}{\eta_1}\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i^{new}})\right)_{i,j=1}^{M} \end{pmatrix}, \\
\Omega_2 &= \begin{pmatrix} \frac{M}{\gamma_1}\left(\left(\frac{1}{\mu_1}+\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i})\right)_{i,j=1}^{M}+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I} & \left(\tau^{-1}\lambda_2\mathbf{I}\right) \\ \left(\tau^{-1}\lambda_2\mathbf{I}\right) & \frac{M}{\eta_1}\left(\left(\frac{1}{\mu_2}+\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i})\right)_{i,j=1}^{M}+\tau^{-1}(1+\lambda_2)\mathbf{I} \end{pmatrix} \\
&= \begin{pmatrix} \left(\left(\frac{1}{\gamma_2}+\frac{M}{\gamma_1}\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{x}_j\mathbf{x}_i})\right)_{i,j=1}^{M}+\tau^{-1}(\lambda_1+\lambda_2)\mathbf{I} & \left(\tau^{-1}\lambda_2\mathbf{I}\right) \\ \left(\tau^{-1}\lambda_2\mathbf{I}\right) & \left(\left(\frac{1}{\eta_2}+\frac{M}{\eta_1}\delta_{ij}\right)diag(\mathbf{k}^{\mathbf{z}_j\mathbf{z}_i})\right)_{i,j=1}^{M}+\tau^{-1}(1+\lambda_2)\mathbf{I} \end{pmatrix}.
\end{aligned}
\tag{29}
$$

Given that $\mathbf{S}^{-1} = \mathbf{P}^T\tilde{\mathbf{S}}^{-1}\mathbf{P}$, which means $\mathbf{S}^{-1}$ can be obtained by allocating the elements of $\tilde{S}_{\tilde{i}\tilde{i}}^{-1}$ to locations with row indices $(\tilde{i}, n+\tilde{i}, 2n+\tilde{i}, \cdots, (M-1)n+\tilde{i})$ and column indices $(\tilde{j}, n+\tilde{j}, 2n+\tilde{j}, \cdots, (M-1)n+\tilde{j})$. The main computational cost of $\mathbf{S}^{-1}$ comes from the calculation of $\tilde{S}_{\tilde{i}\tilde{i}}^{-1}, i = 1, \cdots, n$. The size of $\tilde{S}_{\tilde{i}\tilde{i}}$ is $M \times M$, thus the computational complexity of $\tilde{S}_{\tilde{i}\tilde{i}}$ is $\mathcal{O}(M^3)$. As a result, the computational complexity of $\mathbf{S}^{-1}$ is $\mathcal{O}(nM^3)$. ■

IV. Scale Adaptive K(MF)^{2}JMT

To further improve the overall performance of K(MF)^{2}JMT, we follow the Integrated Correlation Filters (ICFs) framework in [16] to cope with scale variations. The ICF is a cascading-stage filtering process that performs translation estimation and scale estimation, respectively (same as the pipeline adopted in [15] and [9]). Unless otherwise specified, the K(MF)^{2}JMT mentioned in the following experimental parts refers to the scale adaptive one.

Specifically, in scale adaptive K(MF)^{2}JMT, the training of basic K(MF)^{2}JMT is accompanied by the training of another 1D Discriminative Scale Space Correlation Filter (DSSCF) [15] and this new trained filter is performed for scale estimation. To evaluate the trained DSSCF, S image patches centered around the location found by the K(MF)^{2}JMT are cropped from the image, each of size $a^sL \times a^sN$, where $L \times N$ is the target size in the current frame, a is the scale factor, and $s \in \{-\frac{S-1}{2}, ...\frac{S-1}{2}\}$. All S image patches are then resized to the template size for the feature extraction. Finally, the final output from the scale estimation is given as the image patch with the highest filtering response. Similar to K(MF)^{2}JMT, the model parameters are also updated in an interpolating manner with learning rate η. We refer readers to [15] for more details and the implementation of DSSCF.

V. Experiments

We conduct four groups of experiments to demonstrate the effectiveness and superiority of our proposed K(MF)^{2}JMT. First, we implement K(MF)^{2}JMT and several of its baseline variants, including multi-feature-only tracker (MFT), multi-frame-only tracker (MFT-2), scale-adaptive-only tracker (SAT), scale-adaptive multi-feature tracker (SAMFT) and scale-adaptive multi-frame tracker (SAMFT-2), to analyze and evaluate the component-wise contributions to the performance gain. We then evaluate our tracker against 33 state-of-the-art trackers on Object Tracking Benchmark (OTB) 2015 [5]. Following this, we present results on Visual Object Tracking (VOT) 2015 challenge [48]. Finally, we compare K(MF)^{2}JMT with several other representative CFTs, including MOSSE [49], SAMF [9], MUSTer [16] and the recently published C-COT [36], to reveal the properties of our proposed tracker among CFTs family and also illustrate future research directions for modern discriminative tracker design.

A. Experimental setup

To make a comprehensive evaluation, we use all the color video sequences in OTB 2015 dataset (77 in total). These sequences are captured in various conditions, including occlusion, deformation, illumination variation, scale variation, out-of-plane rotation, fast motion, background clutter, etc. We use the success plot to evaluate all trackers on OTB dataset. The success rate counts the percentage of the successfully tracked frames by measuring the overlap score S for trackers on each frame. The average overlap measure is the most appropriate for tracker comparison, which accounts for both position and size. Let B_T denote the tracking bounding box and B_G denote the ground truth bounding box, the overlap score is defined as $S = \frac{|B_T \cap B_G|}{|B_T \cup B_G|}$, where $\cap$ and $\cup$ represent the intersection and union of two regions, and $|\cdot|$ denotes the number of pixels in the region. In success plot, S is varied from 0 to 1, and the ranking of trackers is based on the Area Under Curve (AUC) score. We also report the speed of trackers in terms of average frames per second (FPS) over all testing sequences.

We also test tracker performance on VOT 2015 dataset containing 60 video sequences. The VOT challenge is a popular competition for single object tracking. Different from OTB 2015 dataset, a tracker is restarted in the case of a failure (i.e., there is no overlap between the detected bounding box and ground truth) in the VOT 2015 data set. For the VOT 2015 dataset, tracking performance is evaluated in terms of accuracy (overlap with the ground-truth), robustness (failure rate) and the expected average overlap (EAO). EAO is obtained by combining the raw values of accuracy and failures. Its score represents the average overlap a tracker would obtain on a

typical short-term sequence without reset. For a full treatment of these metrics, interested readers can refer to [48].

B. Implementation details of the proposed tracker

We set the regularization parameters to $\lambda_1 = 0.5$, $\lambda_2 = 1.32$, $\gamma_1 = 0.0006$, $\gamma_2 = 0.005$, $\eta_1 = 0.001$, $\eta_2 = 0.005$. These parameters are tuned with a coarse-to-fine procedure. In the coarse module, we roughly determine a satisfactory range for each parameter (for example, the range of γ_1 is $[0.0001\ 0.001]$). Here, the word "satisfactory" means that the value in the range can achieve higher mean success rate at the threshold 0.5 in the OTB 2015 dataset. Then, in the fine-tuning module, we divide these parameters into three groups based on their correlations: (1) $\{\lambda_1, \lambda_2\}$; (2) $\{\gamma_1, \gamma_2\}$; and (3) $\{\eta_1, \eta_2\}$. When we test the value of one group of parameters, other groups are set to default values, i.e., the mean value of the optimal range given by the coarse module. In each group, the parameters are tuned with grid search (for example, γ_1 in the first group is tuned at the range $[0.0001\ 0.001]$ with an interval 0.0001). We finally pinpointed the specific value as the one that can achieve the highest mean success rate (for example, the final value of γ_1 is 0.0006). Besides, we set the learning rate to $\eta = 0.025$ as previous work [4], [9], [15] and model three consecutive frames in the MTL module, i.e., $M = 3$. We select HOGs [27] and color names [50] for image representation in the MVL module. The HOGs and color names are complementary to each other, as the former puts emphasis on the image gradient which is robust to illumination and deformation while the latter focuses on the color information which is robust to motion blur. For HOGs, we employ a variant in [51], with the implementation provided by [52]. More particular, the cell size is 4×4 and number of orientations is set to 9. For color names, we map the RGB values to a probabilistic 11 dimensional color representation which sums up to 1. All the experiments mentioned in this work are conducted using MATLAB on an Intel i7-4790 3.6GHz Quad-Core PC with 8GB RAM. MATLAB code is available from our project homepage http://bmal.hust.edu.cn/project/KMF2JMTtracking.html.

C. Evaluation on component-wise contributions

Before systematically evaluating the performance of our proposed tracker, we first compare K(MF)^{2}JMT with its baseline variants to demonstrate the component-wise contributions to the performance gain. To this end, we implement seven trackers with various degraded settings, including multi-feature-only tracker (MFT) which just uses multiple feature cues, multi-frame-only tracker (MFT-2) which just uses the temporal relatedness across consecutive frames, scale-adaptive-only tracker (SAT) which just concerns scale variations, scale-adaptive multi-feature tracker (SAMFT) and scale-adaptive multi-frame tracker (SAMFT-2). Besides, to validate the efficiency of modeling multiple feature cues under a MVL framework rather than simply concatenating them, we also implement Multiple Feature tracker with feature concatenation (MFT-C). Note that, the KCF [4], which only uses the HOG feature without temporal modeling and scale searching (i.e., $\lambda_1 = \lambda_2 = 0$, $\eta_1 = \gamma_1 = +\infty$, $M = 1$), serves as a baseline in this section.

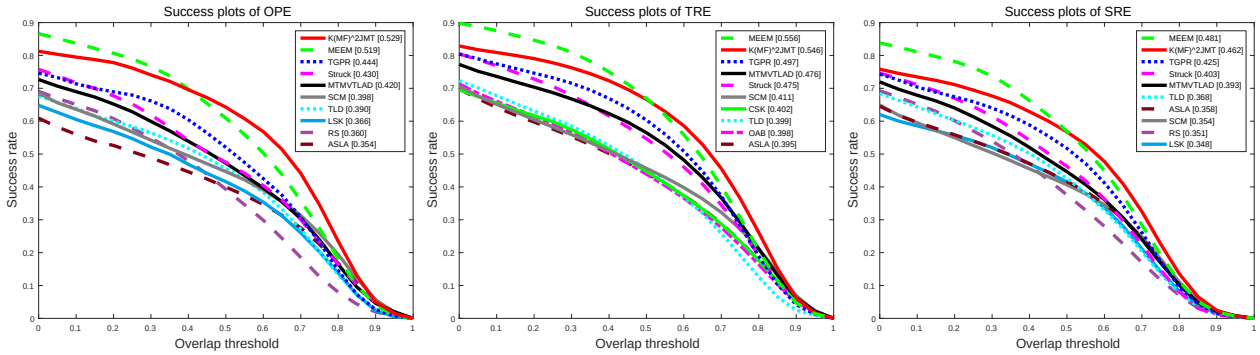
TABLE I
A COMPARISON OF OUR K(MF)^{2}JMT WITH DIFFERENT BASELINE VARIANTS. THE MEAN OVERLAP PRECISION (OP) SCORE (%) AT THRESHOLD 0.5 OVER ALL THE 77 COLOR VIDEOS IN THE OTB DATASET ARE PRESENTED. THE BEST TWO RESULTS ARE MARKED WITH RED AND BLUE RESPECTIVELY. MF, TM AND SA ARE THE ABBREVIATION OF MULTIPLE FEATURES, TEMPORAL MODELING AND SCALE ADAPTIVE, RESPECTIVELY.

	MF	TM	SA	mean success rate
KCF	no	no	no	51.0
MFT	yes	no	no	54.2
MFT-C	yes	no	no	53.7
MFT-2	no	yes	no	54.3
SAT	no	no	yes	56.4
SAMFT	yes	no	yes	60.1
SAMFT-2	no	yes	yes	61.7
K(MF)^{2}JMT-2	yes	yes	no	58.1
K(MF)^{2}JMT	yes	yes	yes	64.3

Table I summarized the differences between these trackers, where K(MF)^{2}JMT-2 denotes the basic kernel multi-frame multi-feature joint modeling tracker without scale estimation. In this table, we report the mean success rate at the threshold of 0.5, which corresponds to the PASCAL evaluation criterion [53]. Although these trackers share one or more common components, their tracking performances differ significantly. This indicates that the visual features, temporal relatedness and scale searching strategy all are essentially important to the visual tracking tasks. KCF ranks the lowest among the compared trackers as expected. MFT (or MFT-C), MFT-2 and SAT extend KCF by augmenting the feature space with color information, taking advantage of temporal information with frame relatedness and introducing scale adaptive searching strategy respectively, thus achieving a few improvements. MFT outperforms MFT-C with a more "intelligent" feature fusion strategy. Besides, it is obvious that the scale adaptive searching can effectively handle scale variations, thus obtaining a large improvement in success rate (see SAT, SAMFT, SAMFT-2 as against KCF, MFT and MFT-2 in Table I). Moreover, by comparing MFT-2 with MFT and comparing SAMFT-2 with SAMFT, one can see that the integration of multiple frames plays a more positive role to improve tracking performance than the integration of multiple features. Finally, it is interesting to find that the success rate gains of MFT, MFT-2 and SAT are 3.2%, 3.3% and 5.4% respectively compared with KCF, while K(MF)^{2}JMT gets a 13.3% improvement. This indicates that the K(MF)^{2}JMT is not just the simple combination of the MFT, MFT-2 and SAT.

D. Comparison with state-of-the-art trackers

In this section, we provide a comprehensive comparison of our proposed K(MF)^{2}JMT with 33 state-of-the-art trackers on OTB 2015 dataset: 1) 4 state-of-the-art trackers that do not follow CFT framework yet achieved remarkable performance on OTB 2015, that is MEEM [54], TGPR [55], LSST [56] and MTMVTLAD [26]; 2) the 29 popular trackers provided in [5], such as Struck [57], ASLA [58], SCM [59], TLD [60], MTT [61], VTD [62], VTS [63] and MIL [64]. Note that, as



(a) One pass evaluation (OPE) (b) Temporal robustness evaluation (TRE) (c) Spatial robustness evaluation (SRE)

Fig. 1. Success plot showing the performance of our K(MF)^{2}JMT compared to 33 state-of-the-art methods on the OTB dataset. The AUC score for each tracker is reported in the legend. Only the top ten trackers are displayed in the legend for clarity in (a) OPE, (b) TRE and (c) SRE. Our approach provides the best performance in OPE and the second best performance in TRE and SRE.

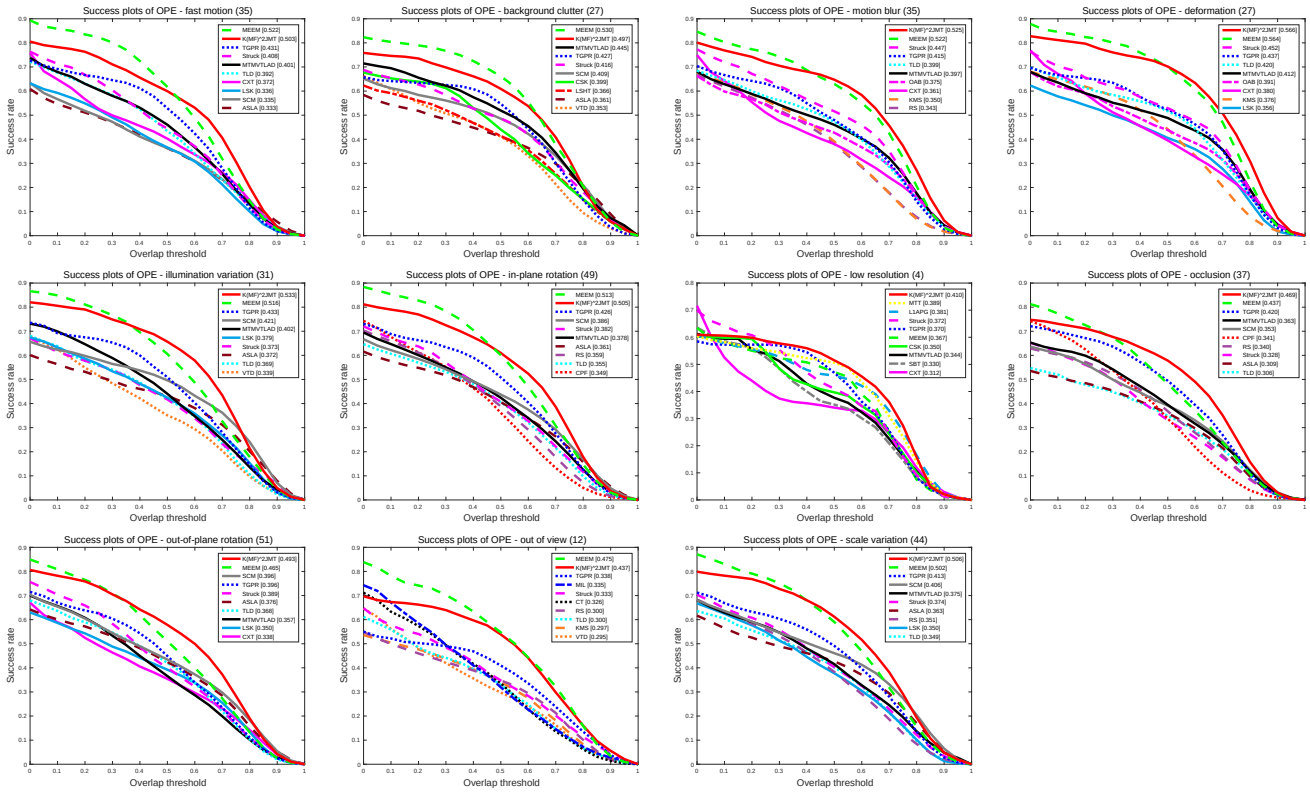


Fig. 2. Success plots for each attribute on OTB dataset. The value presented in the title represents the number of videos corresponding to the attributes. The success score is shown in the legend for each tracker. Our approach provides the best performance on 7 out of 11 attributes, namely motion blur, deformation, illumination variation, low resolution, occlusion, out-of-plane rotation, and scale variation.

a generative tracker, MTMVTLAD has similar motivations as our approach, as it also attempts to integrate multi-task multi-view learning to improve tracking performance. However, different from K(MF)^{2}JMT, MTMVTLAD casts tracking as a sparse representation problem in a particle filter framework. Moreover, MTMVTLAD does not reconcile the temporal coherence in consecutive frames explicitly.

The success plot of all the 34 competing trackers using One Pass Evaluation (OPE) is shown in Fig. 1(a). For clarity, we only show the top 10 trackers in this comparison. As can be seen, our proposed K(MF)^{2}JMT achieves overall the best performance, which persistently outperforms the overall second and third best trackers, i.e., MEEM [54] and TGPR [55]. If we look deeper (see a detailed summarization in Table II), the MEEM tracker, which uses an entropy-minimization-based ensemble learning strategy to avert bad model update, obtains a mean success rate of 60.9%. The transfer learning based TGPR tracker achieves a mean success rate of 52.0%. By contrast, our approach follows a CFT framework, while using an explicit temporal consistency regularization to enhance robustness. The MTMVTLAD tracker provides a mean success rate of 47.2%. Our tracker outperforms MTMVTLAD by 17.1% in mean

TABLE II
A DETAILED COMPARISON OF OUR K(MF)^{2}JMT WITH MEEM [54], TGPR [55] AND MTMVTLAD [26]. THE MEAN OVERLAP PRECISION (OP) SCORE (%) AT THRESHOLD 0.5 OVER ALL THE 77 COLOR VIDEOS IN THE OTB DATASET ARE PRESENTED.

	mean success rate	FPS
MEEM	60.9	13.53
TGPR	52.0	0.70
MTMVTLAD	47.2	0.30
K(MF)^{2}JMT	64.3	30.46

success rate. Finally, it is worth noting that our approach achieves superior performance while operating at real time, while the mean FPS for MEEM, TGPR and MTMVTLAD are approximately 13.53, 0.70 and 0.30, respectively.

We also report the tracking results using Temporal Robustness Evaluation (TRE) and Spatial Robustness Evaluation (SRE). For TRE, it runs trackers on 20 sub-sequences segmented from the original sequence with different lengths, and SRE evaluates trackers by initializing them with slightly shifted or scaled ground truth bounding boxes. With TRE and SRE, the robustness of each evaluated trackers can be comprehensively interpreted. The SRE and TRE evaluations are shown in Fig. 1(b) and Fig. 1(c) respectively. In both evaluations, our approach provides a constant performance gain over the majority of existing methods. The MEEM achieves better robustness than our approach. A possible reason is that MEEM combines the estimates of an ensemble of experts to mitigate inaccurate predictions or sudden drifts, so that the weaknesses of the trackers are reciprocally compensated. We also perform an attribute based analysis on our tracker. In OTB, each sequence is annotated with 11 different attributes, namely: fast motion, background clutter, motion blur, deformation, illumination variation, in-plane rotation, low resolution, occlusion, out-of-plane rotation, out-of-view and scale variation. It is interesting to find that K(MF)^{2}JMT ranks the first on 7 out of 11 attributes (see Fig. 2), especially on illumination variation, occlusion, out-of-plane rotation and scale variation. This verifies the superiority of K(MF)^{2}JMT on target appearance representation and its capability on discovering reliable coherence from short-term memory. However, the overwhelming advantage no longer exists for fast motion and out-of-view. This is because the temporal consistency among consecutive frames becomes weaker in these two scenarios. The qualitative comparison shown in Fig. 3 corroborates quantitative evaluation results. It is worth noting that, we strictly follow the protocol provided in [5] and use the same parameters for all sequences.

E. VOT 2015 Challenge

In this section, we present results on the VOT 2015 challenge. We compare our proposed K(MF)^{2}JMT with 62 participating trackers in this challenge. For a fair comparison, the DSST [15] is substituted with its fast version (i.e., fDSST [65]) raised by the same authors, as fDSST demonstrates superior performance than DSST as shown in [65].

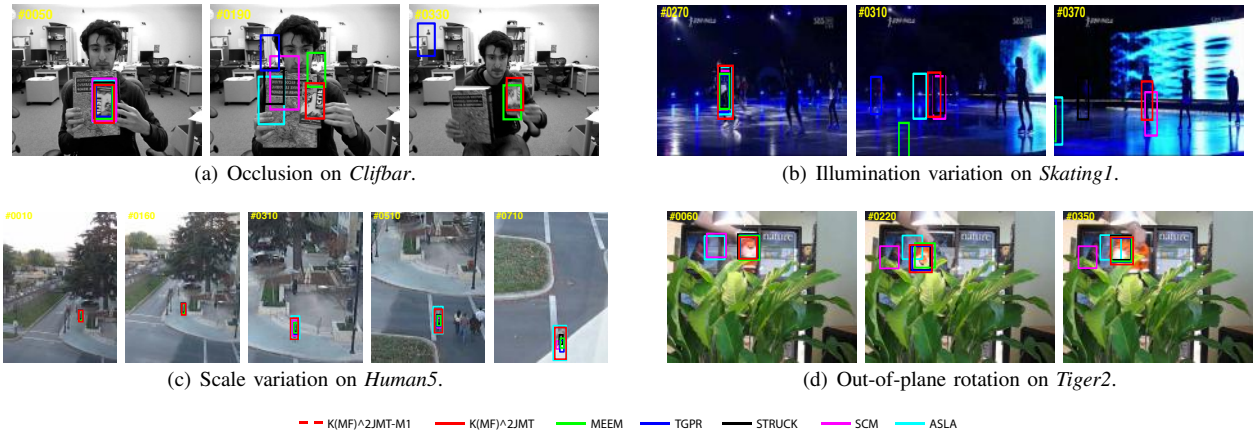
In the VOT 2015 benchmark, each video is annotated by five different attributes: camera motion, illumination change, occlusion, size change and motion change. Different from OTB 2015 dataset, the attributes in VOT 2015 are annotated per-frame in a video. Fig. 4 shows the accuracy and robustness (AR) rank plots generated by sequence pooling and attribute normalization. The pooled AR plots are generated by concatenating the experimental results on all sequences to directly obtain a rank list, while the attribute normalized AR rank plots are obtained based on the average of ranks achieved on these individual attributes. Fig. 5 shows the EAO ranks. Only results for the top-20 trackers in the VOT challenge are reported for clarity.

It is easy to summarize some key observations from these figures:
1) The CFTs (including our tracker K(MF)^{2}JMT, DeepSRDCF [35], SRDCF [66], RAJSSC [67], NSAMF [48], SAMF [9]) account for majority of the top-performing trackers. By fully exploiting the representation power of CNNs or rotating candidate samples to augment candidate set, MDNet [68] and sPST [69] also demonstrate superior (or even the best) performance. This result indicates that a well-designed discriminative tracker with conventional tracking-by-detection scheme and random sampling in the detection phase can achieve almost the same tracking accuracy with state-of-the-art CFTs coupled with dense sampling, at the cost of high computational burden (as officially reported in [48]).
2) Among top-performing CFTs, K(MF)^{2}JMT tied for the first place with DeepSRDCF, SRDCF, RAJSSC and NSAMF in terms of tracking accuracy. However, the robustness of K(MF)^{2}JMT is inferior to DeepSRDCF and SRDCF. Both DeepSRDCF and SRDCF introduce a spatial regularization penalty term to circumvent boundary effects caused by conventional circular correlation, thus significantly mitigating inaccurate training samples and restricted searching regions. However, one should note that these two trackers can hardly run in real time. A thorough investigation between K(MF)^{2}JMT and DeepSRDCF is demonstrated in Section V-F.
3) Our tracker outperforms RAJSSC, NSAMF and SAMF in terms of robustness and EAO values. All these trackers use HOGs and color information to describe target appearance under a scale-adaptive framework. The performance difference indicates that our K(MF)^{2}JMT enjoys an advanced feature fusion scheme and the integration of multiple frames is beneficial to performance gain.

Apart from these three observations, there are other interesting points. For example, the NSAMF achieves a large performance gain compared with SAMF. The main difference is that NSAMF substitutes the color name with color probability. On the other hand, the EBT [70] reaches a fairly high robustness and EAO value. However, its tracking accuracy is desperately poor. One possible reason is that the adopted contour information does not have desirable adaptability to target scale and aspect ratio changes.

F. Comparison among correlation filter-based trackers (CFTs)

In the last section, we investigate the performance of 15 representative CFTs. Our goal is to demonstrate the effectiveness of K(MF)^{2}JMT and reveal its properties when compared



(a) Occlusion on *Clifbar*.

(b) Illumination variation on *Skating1*.

(c) Scale variation on *Human5*.

(d) Out-of-plane rotation on *Tiger2*.

(e) Tracker legend

Fig. 3. A qualitative comparison of our method with five state-of-the-art trackers. Tracking results are shown on four example videos from the OTB 2015 dataset. The videos show challenging situations, such as (a) occlusion, (b) illumination variation, (c) scale variations, and (d) out-of-plane rotation. Our approach offers superior performance compared to the existing trackers in these challenging situations. (e) shows tracker legend.

with other CFT counterparts. We also attempt to illustrate the future research directions of CFTs through a comprehensive evaluation. Table III summarized the basic information of the selected CFTs as well as their corresponding FPS values (some descriptions are adapted from [71]).

Fig. 6 shows the AUC scores of success plots vs. FPS for all competing CFTs in OPE, TRE and SRE, respectively. The dashed vertical line (with a FPS value of 25 [72]) separates the trackers into real-time trackers and those cannot run in real time. Meanwhile, the solid horizontal line (mean of AUC values for all competing CFTs) separates trackers into well-performed trackers and those perform poorly in terms of tracking accuracy. As can be seen, our method performs the best in terms of accuracy among all real-time trackers, thus achieving a good trade-off in speed and accuracy. This suggests that our modifications on the basic KCF tracker are effective and efficient. By contrast, although introducing long-term tracking strategy (e.g., MUSTer, LCT) or imposing spatial regularization penalty term (e.g., DeepSRDCF) can augment tracking performance as well, most of these modifications cannot be directly applied in real applications where the real-time condition is a prerequisite. This unfortunate fact also applies to C-COT, in which the hand-crafted features are substituted with powerful CNN features. Therefore, a promising research direction is to investigate computational-efficient long term tracking strategy or spatial regularization with little or no sacrifice in speed.

Finally, it is worth mentioning that our approach provides a consistent gain in performance compared to MvCFT, although both methods employ the same feature under a MVL framework. The MvCFT only fuses tracking results from different view to provide a more precise prediction. By contrast, our approach enjoys a more reliable and computational-efficient training model. On the other hand, it is surprising to find that SAMF can achieve desirable performance on both TRE and SRE experiments. One possible reason is that the scale estimation method in SAMF, i.e., exhaustively searching a scaling pool, is more robust to scale variations (although time-consuming).

VI. Conclusions and Future Work

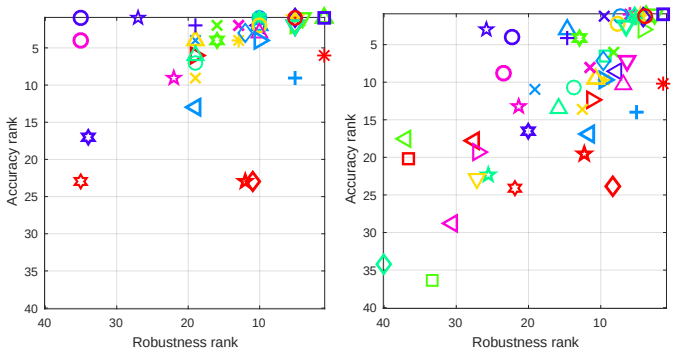
In this paper, we proposed kernel multi-frame multi-feature joint modeling tracker (K(MF)^{2}JMT) to promote the original correlation filter-based trackers (CFTs) by exploiting multiple feature cues and the temporal consistency in a unified framework. A fast *blockwise diagonal matrix* inversion algorithm has been developed to speed up learning and detection. An adaptive scale estimation mechanism was incorporated to handle scale variations. Experiments on OTB 2015 and VOT 2015 datasets show that K(MF)^{2}JMT improves tracking performance in contrast with most state-of-the-art trackers. Our tracker performs well in terms of overlap success in the context of large appearance variations caused by occlusion, illumination, scale variation, etc. Our tracker also demonstrates favorable tracking accuracy and robustness compared with prevalent trackers from different categories (not limited to CFTs). We finally show that K(MF)^{2}JMT can achieve the best tracking accuracy among state-of-the-art real-time correlation filter-based trackers (CFTs).

In future work, we will study how to effectively handle severe drifts and shot changes. For the problem of drifts, possible solutions include closed-loop system design [75] or tracking-and-verifying framework [72]. On the other hand, to circumvent the existence of shot (or scene) changes, possible modifications include assigning different weights to different frames in the overall objective Eq. (1) or explicitly incorporating a short change detector (e.g., [76]), such that the tracker can automatically detect the shot changes. Once a shot change is confirmed, the tracker needs to re-identify the location of the target (see Supplementary Material for initial results). At the same time, we are also interested in investigating computational efficient long term tracking strategy or spatial regularization to further augment tracking accuracy.

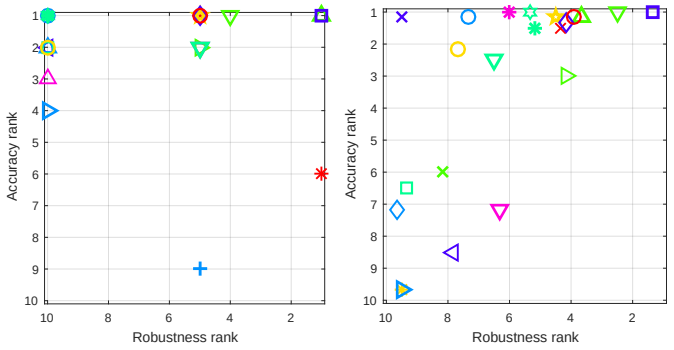
Appendix A

TABLE III
SELECTED COMPETING CFTS (INCLUDING SUMMARIZED MAJOR CONTRIBUTIONS) AND THEIR CORRESPONDING FPS VALUES (ADAPTED FROM [71]).

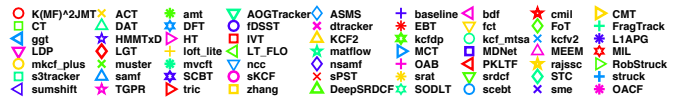
	Published year	Major contribution	FPS
MOSSE [49]	2010	Pioneering work of introducing correlation filters for visual tracking.	367.31
CSK [73]	2012	Introduced Ridge Regression problem with circulant matrix to apply kernel methods.	336.00
STC [10]	2014	Introduced spatio-temporal context information.	572.03
CN [74]	2014	Introduced color attributes as effective features.	171.69
DSST [15]	2014	Relieved the scaling issue using feature pyramid and 3-dimensional correlation filter.	39.54
SAMF [9]	2014	Integrated both color feature and HOG feature; Applied a scaling pool to handle scale variations.	17.23
KCF [4]	2015	Formulated the work of CSK and introduced multi-channel HOG feature.	241.70
LCT [40]	2015	Introduced online random fern classifier as re-detection component for long-term tracking.	21.63
MUSTer [16]	2015	Proposed a biology-inspired framework to integrate short-term processing and long-term processing.	4.54
RPT [39]	2015	Introduced reliable local patches to facilitate tracking.	4.76
CF2 [34]	2015	Introduced features extracted from convolutional neural networks (CNN) for visual tracking.	10.76
DeepSRDCF [35]	2015	Introduced CNN features for visual tracking and a spatial regularization term to handle bound effect.	0.21
MvCFT [12]	2016	Introduced Kullback-Leibler (KL) divergence to fuse multiple feature cues.	7.07
C-COT [36]	2016	Employed an implicit interpolation model to train convolutional filters in continuous spatial domain.	0.22
K(MF)^{2}JMT	2018	Integrated multiple feature cues and temporal consistency in a unified model.	30.46



(a) AR rank (sequence pooling) (b) AR rank (attribute normalization)



(c) Zoomed-in AR rank (sequence pooling) (d) Zoomed-in AR rank (attribute normalization)



(e) Tracker legend

Fig. 4. The accuracy and robustness (AR) rank plots generated by (a) sequence pooling and by (b) attribute normalization in the VOT 2015 dataset. The accuracy and robustness rank are plotted along the vertical and horizontal axis respectively. (c) and (d) demonstrate the zoomed-in figure of (a) and (b) respectively, in which only top-10 accuracy and robustness ranks are plotted. (e) shows the tracker legend. Our proposed K(MF)^{2}JMT (denoted by the red circle) achieves top-6 performance in terms of both accuracy and robustness among 63 competitors in both experiments.

CIRCULAR MATRIX AND BLOCKWISE CIRCULAR MATRIX

Given a vector $\mathbf{x} = [x_0, x_1, ..., x_{n-1}]^T$ of length n and its Discrete Fourier Transform (DFT) $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x})$, the *circular*

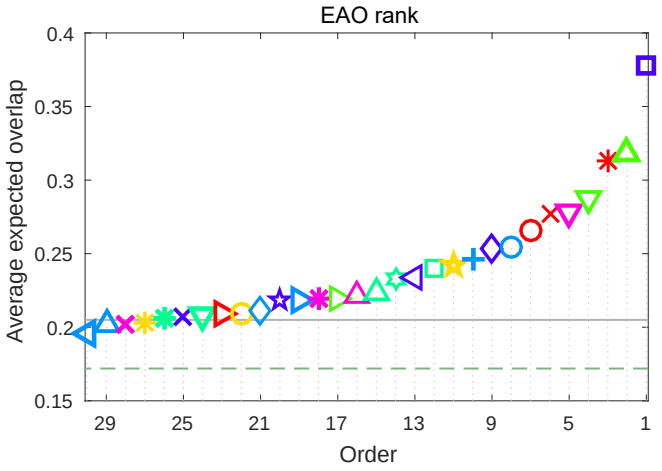


Fig. 5. The expected average overlap (EAO) graphs with trackers ranked from right to left. The right-most tracker is the top-performing in terms of EAO values. The grey solid line denotes the average performance of trackers published at ICCV, ECCV, CVPR, ICML or BMVC in 2014/2015 (nine papers from 2015 and six from 2014), beyond which a tracker can be thought of as state of the art [48]. The green dashed line denotes the performance of VOT 2014 winner (i.e., fDSST [65]). See Fig. 4 for legend.

matrix $X = C(\mathbf{x})$ generated by $\mathbf{x}$ has the following form:

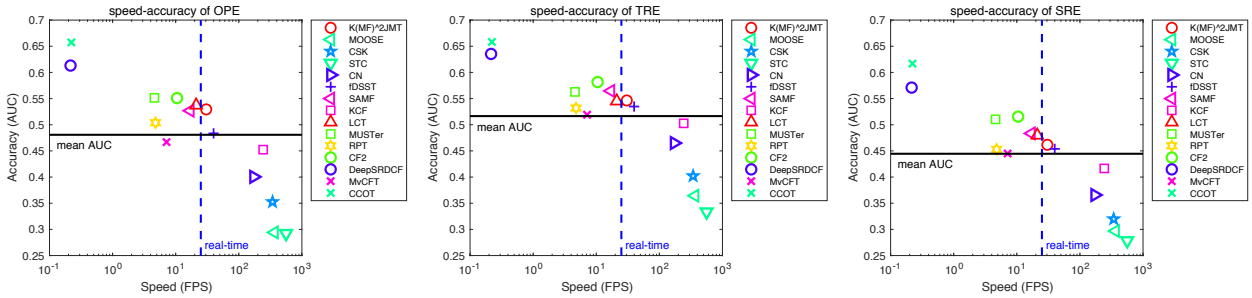
$$X = C(\mathbf{x}) = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_0 & x_1 & \cdots & x_{n-2} \\ x_{n-2} & x_{n-1} & x_0 & \cdots & x_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_0 \end{pmatrix}. \quad (35)$$

[47] proved that X can be diagonalized as:

$$X = F\text{diag}(\hat{\mathbf{x}})F^H, \quad (36)$$

where F is known as the *DFT matrix* ($\hat{\mathbf{x}} = \sqrt{n}F\mathbf{x}$), and $\text{diag}(\hat{\mathbf{x}})$ denotes a square diagonal matrix with elements of $\hat{\mathbf{x}}$ on the diagonal.

We term $\mathbf{X}$ *blockwise circular matrix* if it consists of $M \times M$ blocks $\mathbf{X}_{ij}$ $(i, j = 1, ..., M)$ and each block $\mathbf{X}_{ij}$ is a circular matrix generated by $\mathbf{x}_{ij}$ of length n. We then denote $\mathbf{F}$ a *block*



(a) One pass evaluation (OPE) (b) Temporal robustness evaluation (TRE) (c) Spatial robustness evaluation (SRE)

Fig. 6. Speed and accuracy plot of state-of-the-art CFTs on OTB 2015 dataset. We use the AUC score of success plots to measure tracker accuracy or robustness. The dashed vertical line (with a FPS value of 25 [72]) separates the trackers into real-time trackers and those cannot run in real time. Meanwhile, the solid horizontal line (mean of AUC values for all competing CFTs) separates trackers into well-performed trackers and those perform poorly in terms of tracking accuracy. The proposed K(MF)^{2}JMT achieves the best accuracy among all real-time trackers in terms of (a) OPE, (b) TRE and (c) SRE.

diagonal matrix with blocks F on the main diagonal:

$$\mathbf{F} = \begin{pmatrix} F & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & F \end{pmatrix} \tag{37}$$

and $\mathbf{\Sigma}$ a *blockwise diagonal matrix*:

$$\mathbf{\Sigma} = \begin{pmatrix} \mathrm{diag}(\hat{\mathbf{x}}_{11}) & \cdots & \mathrm{diag}(\hat{\mathbf{x}}_{1M}) \\ \vdots & \ddots & \vdots \\ \mathrm{diag}(\hat{\mathbf{x}}_{M1}) & \cdots & \mathrm{diag}(\hat{\mathbf{x}}_{MM}) \end{pmatrix}, \tag{38}$$

then $\mathbf{X}$ can be decomposed as:

$$\mathbf{X} = \mathbf{F}\mathbf{\Sigma}\mathbf{F}^{H}. \tag{39}$$

Acknowledgment

This work was supported in part by the Key Science and Technology of Shenzhen (No. CXZZ20150814155434903), the Key Program for International S&T Cooperation Projects of China (No. 2016YFE0121200), the Key Science and Technology Innovation Program of Hubei Province (No. 2017AAA017), the Special Projects for Technology Innovation of Hubei Province (2018ACA135), the National Natural Science Foundation of China (No. 61571205 and No. 61772220).

References

[1] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[2] J. Ding, Y. Huang, W. Liu, and K. Huang, "Severely blurred object tracking by learning deep image representations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 2, pp. 319–331, 2016.

[3] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, p. 58, 2013.

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

[5] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[6] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, 2016.

[7] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Robust tracking with weighted online structured learning," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 158–172.

[8] S. Wang, S. Zhang, W. Liu, and D. N. Metaxas, "Visual tracking with reliable memories." in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 3491–3497.

[9] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 254–265.

[10] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 127–141.

[11] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3038–3046.

[12] X. Li, Q. Liu, Z. He, H. Wang, C. Zhang, and W.-S. Chen, "A multi-view model for visual tracking via correlation filters," *Knowledge-Based Systems*, vol. 113, pp. 88–99, 2016.

[13] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *the 11th Annual Conference on Computational Learning Theory (COLT)*, 1998, pp. 92–100.

[14] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004, pp. 109–117.

[15] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference (BMVC)*, 2014.

[16] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 749–758.

[17] X. Zhou, X. Liu, C. Yang, A. Jiang, and B. Yan, "Multi-channel features spatio-temporal context learning for visual tracking," *IEEE Access*, vol. 5, pp. 12 856–12 864, 2017.

[18] H. Fan and J. Xiang, "Robust visual tracking with multitask joint dictionary learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 5, pp. 1018–1030, 2017.

[19] T. Zhou, Y. Lu, and H. Di, "Locality-constrained collaborative model for robust visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 2, pp. 313–325, 2017.

[20] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.

[21] S. He, R. W. Lau, Q. Yang, J. Wang, and M.-H. Yang, "Robust object tracking via locality sensitive histograms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 5, pp. 1006–1017, 2017.

[22] H. T. Nguyen and A. Smeulders, "Tracking aspects of the foreground against the background," in *European Conference on Computer Vision (ECCV)*, 2004, pp. 446–456.

[23] H. Grabner and H. Bischof, "On-line boosting and vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 260–267.

[24] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, 2007.

[25] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[26] X. Mei, Z. Hong, D. Prokhorov, and D. Tao, "Robust multitask multiview tracking in videos," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2874–2890, 2015.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[28] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[29] J. H. Yoon, M.-H. Yang, and K.-J. Yoon, "Interacting multiview tracker," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 903–917, 2016.

[30] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6309–6318.

[31] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Learning a temporally invariant representation for visual tracking," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 857–861.

[32] H. Hu, B. Ma, J. Shen, and L. Shao, "Manifold regularized correlation object tracking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1786–1795, 2018.

[33] Q. Hu, Y. Guo, Z. Lin, W. An, and H. Cheng, "Object tracking using multiple features and adaptive model updating," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 11, pp. 2882–2897, 2017.

[34] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082.

[35] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 58–66.

[36] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 472–488.

[37] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6638–6646.

[38] T. Liu, G. Wang, Q. Yang, and L. Wang, "Part-based tracking via discriminative correlation filters," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[39] Y. Li, J. Zhu, and S. C. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 353–361.

[40] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5388–5396.

[41] E. Gundogdu, H. Ozkan, H. Seckin Demir, H. Ergezer, E. Akagunduz, and S. Kubilay Pakin, "Comparison of infrared and visible imagery for object tracking: Toward trackers with superior ir performance," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 1–9.

[42] Q. Liu, X. Lu, Z. He, C. Zhang, and W.-S. Chen, "Deep convolutional neural networks for thermal infrared object tracking," *Knowledge-Based Systems*, vol. 134, pp. 189–198, 2017.

[43] D. K. Yadav and K. Singh, "A combined approach of kullback–leibler divergence and background subtraction for moving object detection in thermal video," *Infrared Physics & Technology*, vol. 76, pp. 21–31, 2016.

[44] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *SIAM International Conference on Data Mining (SDM)*, vol. 13, 2013, pp. 252–260.

[45] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[46] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[47] R. M. Gray, "Toeplitz and circulant matrices: A review," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2006.

[48] M. Kristan *et al.*, "The visual object tracking vot2015 challenge results," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 1–23.

[49] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2544–2550.

[50] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

[51] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[52] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," https://github.com/pdollar/toolbox.

[53] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[54] J. Zhang, S. Ma, and S. Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 188–203.

[55] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 188–203.

[56] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2371–2378.

[57] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 263–270.

[58] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1822–1829.

[59] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1838–1845.

[60] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[61] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2042–2049.

[62] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1269–1276.

[63] ——, "Tracking by sampling trackers," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1195–1202.

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

[65] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.

[66] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4310–4318.

[67] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 32–40.

[68] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4293–4302.

[69] Y. Hua, K. Alahari, and C. Schmid, "Online object tracking with proposal selection," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3092–3100.

[70] G. Zhu, F. Porikli, and H. Li, "Tracking randomly moving objects on edge box proposals," *arXiv preprint arXiv:1507.08085*, 2015.

[71] Z. Chen, Z. Hong, and D. Tao, "An experimental survey on correlation filter-based tracking," *arXiv preprint arXiv:1509.05520*, 2015.

[72] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5486–5494.
[73] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 702–715.
[74] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1090–1097.
[75] Q. Hu, Y. Guo, Y. Chen, and J. Xiao, "Correlation filter tracking: Beyond an open-loop system," in *British Machine Vision Conference (BMVC)*, 2017.
[76] P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1030–1044, 1999.
[77] M. Birinci and S. Kiranyaz, "A perceptual scheme for fully automatic video shot boundary detection," *Signal Processing: Image Communication*, vol. 29, no. 3, pp. 410–423, 2014.
[78] I. K. Sethi and N. V. Patel, "Statistical approach to scene change detection," in *Storage and Retrieval for Image and Video Databases III*, vol. 2420. International Society for Optics and Photonics, 1995, pp. 329–339.

SUPPLEMENTARY MATERIAL TO ROBUST VISUAL TRACKING USING MULTI-FRAME MULTI-FEATURE JOINT MODELING

A. K(MF)^{2}JMT on video sequences with shot changes and possible modifications

In this section, we provide tracking results of K(MF)^{2}JMT and five state-of-the-art trackers (i.e., MEEM [54], TGPR [55], Struck [57], SCM [59], ASLA [58]) on five video sequences (three are from OTB 2015, the remaining two are from VOT2015 benchmark) with shot changes or scene cuts. We also suggest two modifications to our current K(MF)^{2}JMT to alleviate the negative effects incurred by the these changes.

The first modification is to give different weights to different frames in the overall objective of K(MF)^{2}JMT (i.e., Eq. (1) in the main text). The motivation is intuitive: in the scenarios of shot changes or scene cuts, the temporal coherence (from previous frame) becomes weaker and the tracker needs to assign more weight to the most adjacent (or neighboring) frame to better capture the instantaneous information. The second modification is to incorporate a shot change detector (e.g., [76], [77]) into our K(MF)^{2}JMT, such that the system can automatically detect the shot changes. Once a shot change is confirmed, the system needs to re-detect or re-identify the location of the target. However, one should note that, there is no guarantee that the selected shot detector can reconcile with the given tracker. Moreover, the integration of shot detector will introduce more hyper-parameters.

The selected videos are *DragonBaby*, *BlurOwl*, *Soccer*, *Singer1* and *Singer3*. In the video *DragonBaby*, the shot change is caused by varying camera-subject distances, i.e., there is shot change from full shot to medium shot[2]. In the video *BlurOwl*, the shot change is caused by the sudden changes of camera point-of-view or angle. In the video *Soccer*, the shot change is caused by either the gradual changes of camera point-of-view or the varying camera-subject distances. In the videos *Singer1* and *Singer3*, the shot change is caused by (rapid) changes of both camera point-of-view and camera-subject distances.

We implement the first modification to validate its effectiveness due to its simplicity. Specifically, given M training frames in the overall objective, the weight in the current frame is A_0, then the weights in previous frames are decayed inversely proportional to the square of the distance from the current frame (i.e., the weight in the most adjacent frame is $A_0/4$, the weight in the second most adjacent frame is $A_0/9$, and the weight in the farthest frame is A_0/M^2). We term this modification K(MF)^{2}JMT-M1 and set $A_0 = 5$ in the following proof-of-concept experiment[3]. Fig. 7 plots the tracking results of our K(MF)^{2}JMT and K(MF)^{2}JMT-M1 as well as their five competitors. Table IV summarizes the overlap precision (%) at threshold 0.5 for all competing trackers.

As can be seen, our basic K(MF)^{2}JMT performs favorably in these videos, but it may miss the target or overestimate the target size due to unconstrained shot changes. The simple modification can effectively alleviate the negative effects incurred by these changes, thus further improving the performance of K(MF)^{2}JMT. This result suggests that the precise utilization of temporal information (coupled with a careful weighting strategy) is preferred in (unconstrained) videos containing shot changes or scene cuts. At the same time, it also suggests the (possible) existence of the room for performance improvement with an advanced strategy to address shot changes. We leave the implementation of the second modification as future work.

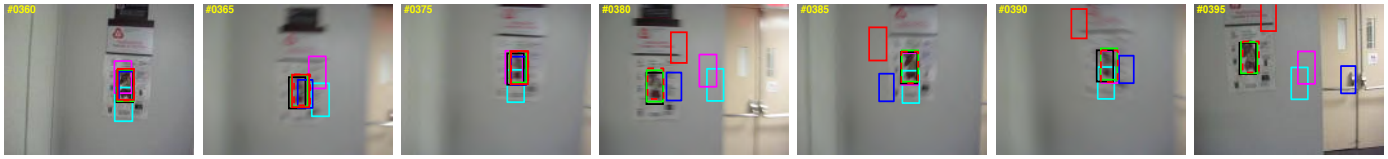
TABLE IV
A COMPARISON OF K(MF)^{2}JMT AND K(MF)^{2}JMT-M1 WITH FIVE STATE-OF-THE-ART TRACKERS. FOR EACH TRACKER, THE OVERLAP PRECISION (%) AT THRESHOLD 0.5 IS PRESENTED. THE BEST TWO RESULTS ARE MARKED WITH RED AND BLUE RESPECTIVELY.

	MEEM	TGPR	STRUCK	SCM	ASLA	K(MF)^{2}JMT	K(MF)^{2}JMT-M1
DragonBaby	65.5	73.5	8.8	23.0	15.0	46.0	66.4
BlurOwl	98.6	51.2	98.6	21.6	17.6	55.9	90.2
Soccer	36.0	13.0	15.6	23.7	12.5	56.6	78.6
Singer1	25.1	22.8	29.9	100	100	93.7	98.6
Singer3	15.3	15.3	24.4	15.3	16.0	17.6	37.4
Mean	48.1	35.2	35.5	36.7	32.2	54.0	74.2

[2]Please refer to [78] for definitions of full shot, medium shot, etc.

[3]The parameter A_0 is selected, from the range $[1, 10]$ with interval 1, as the one that achieves the highest mean success rate among all selected video sequences with scene cuts.

(a) Shot changes in *DragonBaby*: there are abrupt shot changes from full shot to medium shot (see frame 78 to frame 84) and from medium shot to full shot (see frame 88 to frame 94).



(b) Shot changes in *BlurOwl*: there are abrupt shot changes due to the sudden change of camera point-of-view (see the transition between frame 375 and frame 380 or the transition between frame 390 and frame 395).

(c) Shot changes in *Soccer*: there are gradual shot changes due to the changes of camera point-of-view (see frame 45 to frame 90 and frame 335 to frame 375) or incurred by varying camera-subject distances (see frame 200 to frame 230 and finally to frame 305).

(d) Shot changes in *Singer1*: there are shot changes due to the gradual changes of both camera point-of-view and camera-subject distances (see frame 2 to frame 102 and finally to frame 302).

(e) Shot changes in *Singer3*: there are shot changes due to the gradual (and rapid) changes of both camera point-of-view and camera-subject distances (e.g., frame 26 to frame 36). Our modification K(MF)^{2}JMT-M1 may underestimate the target size due to the rapid changes, but it still provides the most accurate estimation among others.

(f) Tracker legend

Fig. 7. A qualitative comparison of our method and its modification with five state-of-the-art trackers. Tracking results are shown on five videos contain scene cuts or shot transitions. *DragonBaby*, *BlurOwl* and *Soccer* are from OTB 2015, whereas *Singer1* and *Singer3* are from VOT2015 benchmark. The basic K(MF)^{2}JMT performs favorably in these videos. Our modification K(MF)^{2}JMT-M1 offers the best performance. (f) shows tracker legend.