Online Multi-Object Tracking with Dual Matching Attention Networks

Ji Zhu[1,2], Hua Yang[1⋆], Nian Liu[3], Minyoung Kim[4],
Wenjun Zhang[1], and Ming-Hsuan Yang[5,6]

[1]Shanghai Jiao Tong University [2]Visbody Inc
[3]Northwestern Polytechnical University [4]Massachusetts Institute of Technology
[5]University of California, Merced [6]Google Inc
{jizhu1023, liunian228}@gmail.com minykim@mit.edu
{hyang,zhangwenjun}@sjtu.edu.cn mhyang@ucmerced.edu

1 Overview

This supplementary material contains the concrete derivation of the proposed cost-sensitive loss and more experimental results on the MOT benchmark datasets.

2 Optimization of the Cost-Sensitive Tracking Loss

We propose a cost-sensitive tracking loss defined as follows (the same as (3) in the paper):

$$E(f) = \sum_{j=1}^{M} \alpha_j \left\| q(t)(S_f\{\mathbf{x}_j\}(t) - y_j(t)) \right\|_{L^2} + \sum_{d=1}^{D} \left\| w(t) f^d(t) \right\|_{L^2}. \tag{1}$$

This objective function aims to learn a continuous T-periodic multi-channel convolution filter $f = \{f^1, \cdots, f^D\}$ from a batch of M training samples. In (1), $\mathbf{x}_j = \{\mathbf{x}_j^1, \cdots, \mathbf{x}_j^D\}$ is the feature map with D feature channels extracted from the j^{th} training sample. Each feature channel $\mathbf{x}_j^d \in \mathbb{R}^{N_d}$ has a resolution N_d. Note that in [1], the ECO tracker introduces a factorized convolution operator $\mathbf{P}$ to reduce the dimensionality of the feature map. Here in our formulation, $\mathbf{x}_j$ denotes the feature map obtained after dimensionality reduction. For clarity, we first consider the optimization on single training sample $\mathbf{x}_j$:

$$E_j(f) = \left\| q_j(t)(S_f\{\mathbf{x}_j\}(t) - y_j(t)) \right\|_{L^2} + \sum_{d=1}^{D} \left\| w(t) f^d(t) \right\|_{L^2}. \tag{2}$$

To enable the target localization with sub-pixel precision, the discrete feature map $\mathbf{x}_j$ is transformed to the continuous spatial domain $t \in [0, T)$ by the interpolation operator J_d:

$$J_d\{\mathbf{x}_j^d\}(t) = \sum_{n=0}^{N_d - 1} x_j^d[n] b_d(t - \frac{T}{N_d} n). \tag{3}$$

⋆ Corresponding author.

Here, $x_j^d[n]$ is the feature value of $\mathbf{x}_j^d$ indexed by the discrete spatial variable $n \in \{0, \cdots, N_d - 1\}$ and $b_d(t)$ is a cubic spline kernel with period T. Hence, in (2), the convolution response map of the filter f on the feature map $\mathbf{x}_j$ is defined as:

$$S_f\{\mathbf{x}_j\}(t) = f * J\{\mathbf{x}_j\}(t) = \sum_{d=1}^{D} f^d * J_d\{\mathbf{x}_j^d\}(t). \tag{4}$$

In (4), the circular convolution operation $*$ in the continuous spatial domain is defined as $g * h(t) := \frac{1}{T}\int_0^T g(t-s)h(s)ds$.

By applying the Parseval's theorem, the object function (2) can be transformed to the equivalent objective function in the Fourier domain:

$$E_j(f) = \left\| \hat{q}_j * \left(\widehat{S_f\{\mathbf{x}_j\}}[k] - \hat{y}_j[k] \right) \right\|_{L^2} + \sum_{d=1}^{D} \left\| \hat{w} * \hat{f}^d[k] \right\|_{L^2}. \tag{5}$$

In (5), the hat $\hat{h}$ denotes the Fourier coefficients of the T-periodic function h. The circular convolution operation $*$ in the Fourier domain is defined as $\hat{g} * \hat{h}[k] := \sum_{l=-\infty}^{+\infty} \hat{g}[k-l]\hat{h}[l]$. Using the linearity and the convolution property of Fourier series, the Fourier coefficients of the confidence score $S_f\{\mathbf{x}_j\}$ in (4) can be obtained as

$$\widehat{S_f\{\mathbf{x}_j\}}[k] = \sum_{d=1}^{D} \hat{f}^d[k] X_j^d[k] \hat{b}_d[k], \quad k \in \mathbb{Z}, \tag{6}$$

where $X_j^d[k]$ is the Discrete Fourier Transform (DFT) of the feature channel $\mathbf{x}_j^d$. Thus, the loss function (2) can be optimized with respect to $\hat{f}^d$ in the Fourier domain as:

$$E_j(f) = \left\| \hat{q}_j * \left(\sum_{d=1}^{D} \hat{f}^d[k] X_j^d[k] \hat{b}_d[k] - \hat{y}_j[k] \right) \right\|_{L^2} + \sum_{d=1}^{D} \left\| \hat{w} * \hat{f}^d[k] \right\|_{L^2}. \tag{7}$$

Assume $\hat{f}^d[k] = 0$ for $|k| > K_d$ where $K_d = \lfloor \frac{N_d}{2} \rfloor$, the filter f^d is then parameterized by N_d non-zero Fourier coefficients $\hat{\mathbf{f}}^d = (\hat{f}^d[-K_d], \cdots, \hat{f}^d[K_d])^\top \in \mathbb{C}^{2K_d+1}$. We define $\hat{\mathbf{y}}_j = [\hat{y}_j[-K], \cdots, \hat{y}_j[K]]^\top$ as the $K := \max_d K_d$ first Fourier coefficients of y_j. Like [1], to simplify the convolution operations in (7), we let L_w denote the number of non-zero coefficients $\hat{w}[k]$, such that $\hat{w}[k] = 0$ for all $|k| > L_w$. Then we define $\mathbf{W}^d$ to be the $(2K_d + 2L_w + 1) \times (2K_d + 1)$ Toeplitz matrix corresponding to the convolution operator $\mathbf{W}^d\hat{\mathbf{f}}^d = \operatorname{vec} \hat{w} * \hat{f}^d$. Similarly, assume $\hat{q}_j[k] = 0$ for $|k| > L_q$, we further define $\mathbf{Q}_j$ to be the $(2K + 2L_q + 1) \times (2K + 1)$ Toeplitz matrix corresponding to the convolution operator $\mathbf{Q}_j(\sum_{d=1}^{D} \hat{\mathbf{f}}^d X_j^d \hat{b}_d - \hat{\mathbf{y}}_j) = \operatorname{vec} \hat{q}_j * (\sum_{d=1}^{D} \hat{\mathbf{f}}^d X_j^d \hat{b}_d - \hat{\mathbf{y}}_j)$. Thus, (7) can be rewritten as:

$$E_j(f) = \left\| \mathbf{Q}_j \left(\sum_{d=1}^{D} \mathbf{A}_j^d \hat{\mathbf{f}}^d - \hat{\mathbf{y}}_j \right) \right\|_{L^2} + \sum_{d=1}^{D} \left\| \mathbf{W}^d \hat{\mathbf{f}}^d \right\|_{L^2}, \tag{8}$$

where $\mathbf{A}_j^d$ is the diagonal matrix containing the elements $\{X_j^d[k]\hat{b}_d[k]\}_{-K_d}^{K_d}$.

Finally, let $\hat{\mathbf{f}} = [(\hat{\mathbf{f}}^1)^\top, \cdots, (\hat{\mathbf{f}}^D)^\top]^\top$ be the vectorization of Fourier coefficients $\hat{\mathbf{f}}^d$. We define $\mathbf{A}_j = [\mathbf{A}_j^1, \cdots, \mathbf{A}_j^D]$ ($\mathbf{A}_j^d$ is padded to have $2K+1$ rows), and let $\mathbf{W}$ be the block-diagonal matrix $\mathbf{W} := \mathbf{W}^1 \bigoplus \cdots \bigoplus \mathbf{W}^D$. The minimization of (8) is equivalent to the following least squares problem:

$$E_j(\hat{\mathbf{f}}) = \left\| \mathbf{Q}_j \left(\mathbf{A}_j \hat{\mathbf{f}} - \hat{\mathbf{y}}_j \right) \right\|_{L^2} + \left\| W \hat{\mathbf{f}} \right\|_{L^2}, \tag{9}$$

which can be optimized by solving the normal equations:

$$\left((\mathbf{Q}_j \mathbf{A}_j)^{\mathrm{H}} (\mathbf{Q}_j \mathbf{A}_j) + \mathbf{W}^{\mathrm{H}} \mathbf{W} \right) \hat{\mathbf{f}} = (\mathbf{Q}_j \mathbf{A}_j)^{\mathrm{H}} \mathbf{Q}_j \hat{\mathbf{y}}_j. \tag{10}$$

Here, H denotes the conjugate transpose of a matrix. Like in [1], we use the Conjugate Gradient (CG) method to iteratively solve (10). The dominating computation in CG is the evaluation of the left term in (10). Because CG does not require explicit evaluations of the matrix product $(\mathbf{Q}_j \mathbf{A}_j)^{\mathrm{H}} (\mathbf{Q}_j \mathbf{A}_j)$, we can compute the left term efficiently by switching operations between the spatial domain and the Fourier domain. More specifically, the left term in (10) can be rewritten as $\mathbf{A}_j^{\mathrm{H}} (\mathbf{Q}_j^{\mathrm{H}} (\mathbf{Q}_j (\mathbf{A}_j \hat{\mathbf{f}}))) + \mathbf{W}^{\mathrm{H}} \mathbf{W} \hat{\mathbf{f}}$, where the parentheses are used to indicate the order in which the operations are performed. As defined in (8) and (9), $\mathbf{A}_j \hat{\mathbf{f}} = \widehat{S_f\{\mathbf{x}_j\}}$. Hence, $\mathbf{Q}_j^{\mathrm{H}} (\mathbf{Q}_j (\mathbf{A}_j \hat{\mathbf{f}})) = \hat{q}_j^{\mathrm{H}} * (\hat{q}_j * \widehat{S_f\{\mathbf{x}_j\}})$. Because $\hat{q}_j$ is conjugate symmetric (as $q_j(t)$ is a real-valued function), we obtain:

$$\mathbf{Q}_j^{\mathrm{H}} (\mathbf{Q}_j (\mathbf{A}_j \hat{\mathbf{f}})) = \hat{q}_j * (\hat{q}_j * \widehat{S_f\{\mathbf{x}_j\}}) = \widehat{q_j^2 S_f\{\mathbf{x}_j\}}. \tag{11}$$

Thus, the value of $\mathbf{Q}_j^{\mathrm{H}} (\mathbf{Q}_j (\mathbf{A}_j \hat{\mathbf{f}}))$ can be obtained by first performing element-wise multiplication in the spatial domain and then applying the Fast Fourier Transform (FFT) to transform it to the Fourier domain, which is more efficient than convolution operation in the Fourier domain when the kernel $\hat{q}_j[k]$ of the convolution operator $\mathbf{Q}_j$ is large. Finally, another element-wise multiplication in the Fourier domain is performed to compute the left term in (10). The right term in (10) can be computed using the same strategy. Compared to the normal equations $(\mathbf{A}_j^{\mathrm{H}} \mathbf{A}_j + \mathbf{W}^{\mathrm{H}} \mathbf{W}) \hat{\mathbf{f}} = \mathbf{A}_j^{\mathrm{H}} \hat{\mathbf{y}}$ in [1], the additional computations mainly come from the element-wise multiplication in the spatial domain, which only have a marginal impact on the overall processing time.

The generalization of (10) to M training samples can be obtained by applying a weighted summation on both sides of the equation as follows:

$$\left((\mathbf{Q}\mathbf{A})^{\mathrm{H}} \mathbf{\Gamma} (\mathbf{Q}\mathbf{A}) + \mathbf{W}^{\mathrm{H}} \mathbf{W} \right) \hat{\mathbf{f}} = (\mathbf{Q}\mathbf{A})^{\mathrm{H}} \mathbf{\Gamma} \mathbf{Q} \hat{\mathbf{y}}, \tag{12}$$

where $\mathbf{\Gamma} = \alpha_1 \mathbf{I} \oplus \cdots \alpha_M \mathbf{I}$ is a diagonal matrix containing the weight α_j for each sample $\mathbf{x}_j$.

3 Visualization of Dual Matching Attention Maps

Figure 1 shows more visualization results of the attention maps generated by the proposed Dual Matching Attention Networks (DMAN). As we can see, the

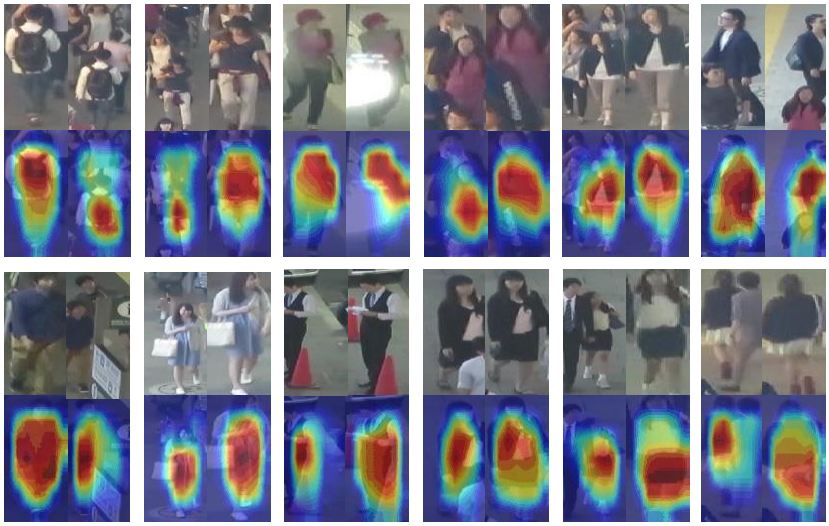

Fig. 1. More visualization results of dual matching attention maps. The red color indicates a high attention value while the blue color refers to a low attention value.

Table 1. Density and tracking speed of each squence in the MOT16 training set

Sequence	16-02	16-04	16-05	16-09	16-10	16-11	16-13
Density	29.7	45.3	8.1	10.0	18.8	10.2	15.3
Runtime	2.1	4.0	1.4	1.7	3.7	1.2	2.1

generated attention maps enable the model to focus on the common patterns between the hard positive pair of images, which enhance the model robustness to noisy detections undergoing misalignment, scale change and occlusion.

4 Runtime Analysis

The overall tracking speed of the proposed approach on the MOT16 training dataset is 0.42 frame per second (FPS) using 2.4GHz CPU and GeForce GTX 1080 Ti GPU. Table 1 presents how the runtime scales with an increasing number of targets. The density (provided by the MOT16 benchmark) indicates the average number of pedestrians per frame. The runtime indicates the average running time (second) for each frame.

References

1. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: Efficient convolution operators for tracking. In: CVPR (2017)