

DeepDCF: The best feature for the DCF based  
tracker  
DCFNet: Discriminant Correlation Filter Network  
on Visual Tracking  
Best Convolutional Features for Correlation Filter  
based Visual Tracking

QiangWang

December 20, 2016

## 1 Abstract

Discriminative Correlation Filters(DCF) have been widely used recent years in visual tracking due to the high precision and fast training and detection.

Convolutional Networks can enjoy millions labeled images for the offline representation learning. CNN based object classification and detection make a revolutionary progress due to the rich representation abilities. Classification even loss the spatial information.

Tracking is different with detection and classification. Classification and DET both eliminate the differences in the same class.

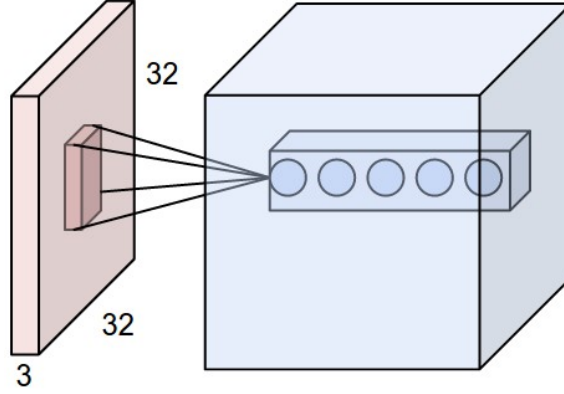
To put the tracking problem simple to the similarity metrics for total offline learning is a unreasonable hypothesis. The surrounding and the target motion fusing in temporal jointly determine the trajectory of the target.

## 2 Introduction

### 2.1 related work

### 2.2 CNN

CNN can be treated as a encode function  $\Phi(x)$  of receptive field.



$$\Phi(x) = f^l$$

$$x \in \mathbb{R}^{m \times n} \longrightarrow f \in \mathbb{R}^d$$

It is sparse (only a few input units contribute to a given output unit) and reuses parameters (the same weights are applied to multiple locations in the input)

### 2.3 contributions

## 3 Discriminative Correlation Filters

### 3.1 DSST Derivation

#### (PCA-HOG+Gray)+DCF+Scale Estimation

key word: circular correlation,Parseval's identity, dense feature

##### 3.1.1 Single Feature(gray)

patch:  $x_1, \dots, x_t$

label:  $y_1, \dots, y_t$

filter:  $w_t$

test patch:  $z$

$$\epsilon = \sum_{j=1}^t \|w_t \star x_j - y_j\|^2 = \sum_{j=1}^t \|\overline{W}_t \odot X_j - Y_j\|^2$$

$$W_t = \frac{\sum_{j=1}^t \bar{Y}_j \odot X_j}{\sum_{j=1}^t X_j \odot \bar{X}_j}$$

$$y = w_t \star z = \mathfrak{F}^{-1}(\bar{W}_t \odot Z) = \mathfrak{F}^{-1}\left(\frac{\sum_{j=1}^t \bar{X}_j \odot Y_j \odot Z}{\sum_{j=1}^t \bar{X}_j \odot X_j}\right)$$

This is a little different between MOSSE[1] and this Derivation. No regularization term.

### 3.1.2 Multidimensional Features

patch:  $x_t^l, l \in \{1, \dots, d\}$

label:  $y$

filter:  $w_t^l$

test patch:  $z^l$

$$\epsilon = \left\| \sum_{l=1}^d w_t^l \star x_t^l - y \right\|^2 + \lambda \sum_{l=1}^d \|w^l\|^2$$

$$W_t^l = \frac{\bar{Y} X_t^l}{\sum_{k=1}^d \bar{X}_t^k X_t^k + \lambda}$$

To obtain a robust approximation, here we update the numerator  $A_t^l$  and denominator  $B_t$

$$A_t^l = (1 - \eta) A_{t-1}^l + \eta \bar{G}_t F_t^l$$

$$B_t^l = (1 - \eta) B_{t-1}^l + \eta \sum_{k=1}^d \bar{F}_t F_t^l$$

$$y = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^d \bar{A}_t^l Z^l}{B_t + \lambda} \right\}$$

### 3.2 CN

### 3.3 SRDCF

### 3.4 CF2

## 4 DCFNet

The network in this filed utilize pooling to reduce the computation complexity and eliminate the spatial information. So we rebuild the network with 1 stride and total small size kernel convnet. We think the resolution in the spatial must be reserved. For the shortcomings of small stride and no pooling, we have a small receptive field. So we anlynsis use the DilatedNet to increase the receptive field for each position.

target patch:  $x^l, l \in \{1, \dots, d\}$

idea label:  $y$

filter:  $w^l$

test patch:  $z^l$

test output:  $g$

For the learing part.

$$\epsilon = \left\| \sum_{l=1}^d w^l \star x^l - y \right\|^2 + \lambda \sum_{l=1}^d \|w^l\|^2$$

$$W^l = \frac{\bar{Y} \odot X^l}{\sum_{k=1}^d \bar{X}^k \odot X^k + \lambda}$$

$$g = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^d \bar{A}^l \odot Z^l}{B + \lambda} \right\}$$

$$g = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^d \bar{A}^l \odot Z^l}{B + \lambda} \right\} = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^d \bar{Y} \odot \bar{X}^l \odot Z^l}{\sum_{k=1}^d \bar{X}^k \odot X^k + \lambda} \right\} = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^d (Z^l \odot \bar{X}^l) \odot Y}{\sum_{k=1}^d \bar{X}^k \odot X^k + \lambda} \right\}$$

$$g = \text{ifft2} \left( \left( \text{sum}(\text{fft2}(z) .* \text{conj}(\text{fft2}(x))), 3 \right) / \text{numel}(x) .* \text{fft2}(y) \right) ./ \dots$$

$$\left( \text{sum}(\text{fft2}(x) .* \text{conj}(\text{fft2}(x))), 3 \right) / \text{numel}(x) + \text{lambda} \right);$$

The forward pass derivation should be familiar to us by now. There need some patience for the backward pass.

First, let's begin with some fundamental theorems of DFT and Complex-Valued Derivatives.

Because the DFT is a linear operator, its gradient is simply the transformation matrix itself. During the back-propagation, then, the gradient is conjugated.[3]

$$Y = \mathfrak{F}\{y\}, \frac{\partial l}{\partial Y} = \mathfrak{F}\left\{\frac{\partial l}{\partial y}\right\}$$

The second theorem we will use is Complex-Valued Derivatives[2]. Because we need pass  $\delta$  through the frequency domain. We will be more familiar with the derivatives operator about complex. Two basic formulas we should revisit.

Let  $z = x + yi \in \mathbb{C}$  represent a complex scalar variable, where  $Re(z) = x$  and  $Im(z) = y$ . The following four relations hold between the real and imaginary parts of  $z$  and its complex conjugate  $z^*$ :

$$\begin{aligned} z &= x + iy \\ z^* &= x - iy \\ x &= \frac{z + z^*}{2} \\ y &= \frac{z - z^*}{2i} \end{aligned}$$

For complex differentials, these four relations can be formulated as follows:

$$\begin{aligned} dz &= dx + i dy \\ dz^* &= dx - i dy \\ dx &= \frac{dz + dz^*}{2} \\ dy &= \frac{dz - dz^*}{2i} \end{aligned}$$

$$\begin{aligned} \frac{\partial f(z)}{\partial z} \\ \frac{\partial f(z, z^*)}{\partial z} = \overline{\frac{\partial f(z, z^*)}{\partial z^*}} \end{aligned}$$

Now, Let's derive the formulas about the backward. For simplify, We start with  $\frac{\partial l}{\partial g}$  and what we need is to results of  $\frac{\partial l}{\partial x}$  and  $\frac{\partial l}{\partial z}$ .

$$G_{uv} = \frac{\sum_{l=1}^d (Z_{uv}^l \overline{X_{uv}^l}) Y_{uv}}{\sum_{k=1}^d X_{uv}^k \overline{X_{uv}^k} + \lambda}$$

$$\frac{\partial l}{\partial Z_{uv}^l} = \frac{\partial l}{\partial G_{u,v}} \frac{\partial G_{u,v}}{\partial Z_{uv}^l} = \mathfrak{F} \left\{ \frac{\partial l}{\partial g} \right\}_{uv} \frac{\overline{X_{uv}^l} Y_{uv}}{\sum_{k=1}^d X_{uv}^k \overline{X_{uv}^k} + \lambda}$$

$$\frac{\partial l}{\partial z_{uv}^l} = \mathfrak{F}^{-1} \left\{ \frac{\partial l}{\partial Z^l} \right\}_{uv}$$

$$\frac{\partial l}{\partial X_{uv}^l} = \frac{\partial l}{\partial G_{u,v}} \frac{\partial G_{u,v}}{\partial X_{uv}^l} = \mathfrak{F} \left\{ \frac{\partial l}{\partial g} \right\}_{uv} \frac{\overline{Z_{uv}^l} Y_{uv} (\sum_{k=1}^d X_{uv}^k \overline{X_{uv}^k} + \lambda) - \overline{X_{uv}^l} (\sum_{l=1}^d (Z_{uv}^l \overline{X_{uv}^l}) Y_{uv})}{(\sum_{k=1}^d X_{uv}^k \overline{X_{uv}^k} + \lambda)^2}$$

$$\frac{\partial l}{\partial x_{uv}^l} = \mathfrak{F}^{-1} \left\{ \frac{\partial l}{\partial X^l} \right\}_{uv}$$

---

**Algorithm 1** Forward pass: Calculate  $y$

---

**Require:** target  $x$ , search  $z$ , initialize position  $p$ , ConvNet  $\Phi(\cdot)$

**return**  $y$   
**for**  $i := 1$  **to**  $n$  **do**  
    crop and feature  
    detection and update  
**end for**

---



---

**Algorithm 2** Backward propagation: Calculate  $\Phi(\cdot)$

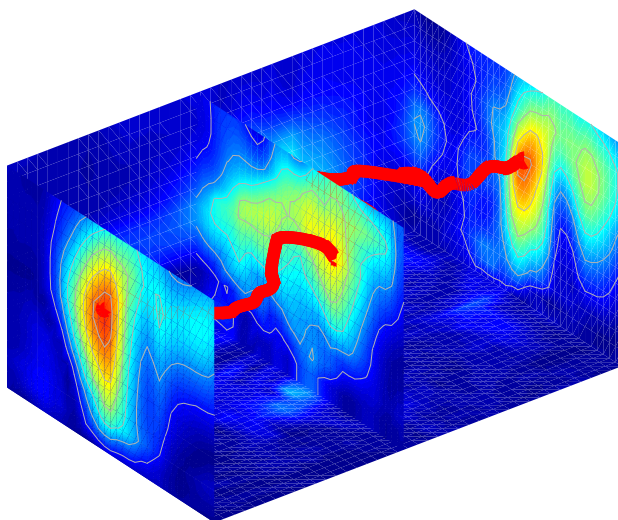
---

**Require:** Gradient output  $\frac{\partial l}{\partial y}$   
**return** Gradient input  $\frac{\partial l}{\partial x}, \frac{\partial l}{\partial z}$

---

## 5 Experiments

I think that is a very perfect visualization of the training process.



### 5.1 Details and Paramemters

### 5.2 Baseline Comparison

Compare with handcraft feature. Compare different vgg(from shallow to deep) Compare different receptive field. Compare different Network(Alex,VGG,ResNet,tingNet)  
Compare different memory cost for the net struct.

### 5.3 OTB

In this benchmark,we should pay attention to raw results. Even We can training a very deep network like ResNet.

### 5.4 VOT

In this part, we should pay more attention in speed.

AR rank (EAO vs speed) should be presented.

## 6 Conclusions

We are the first to achieve end-to-end learning in DCF.

## References

- [1] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE, 2010.
- [2] Are Hjørungnes. *Complex-valued matrix derivatives: with applications in signal processing and communications*. Cambridge University Press, 2011.
- [3] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 2449–2457, 2015.