Visual Object Tracking

Mengmeng Wang
mengmewang@gmail.com
知乎专栏： 目标跟踪算法

4/25/2017

Outline

Overview about visual tracking

Typical Trackers

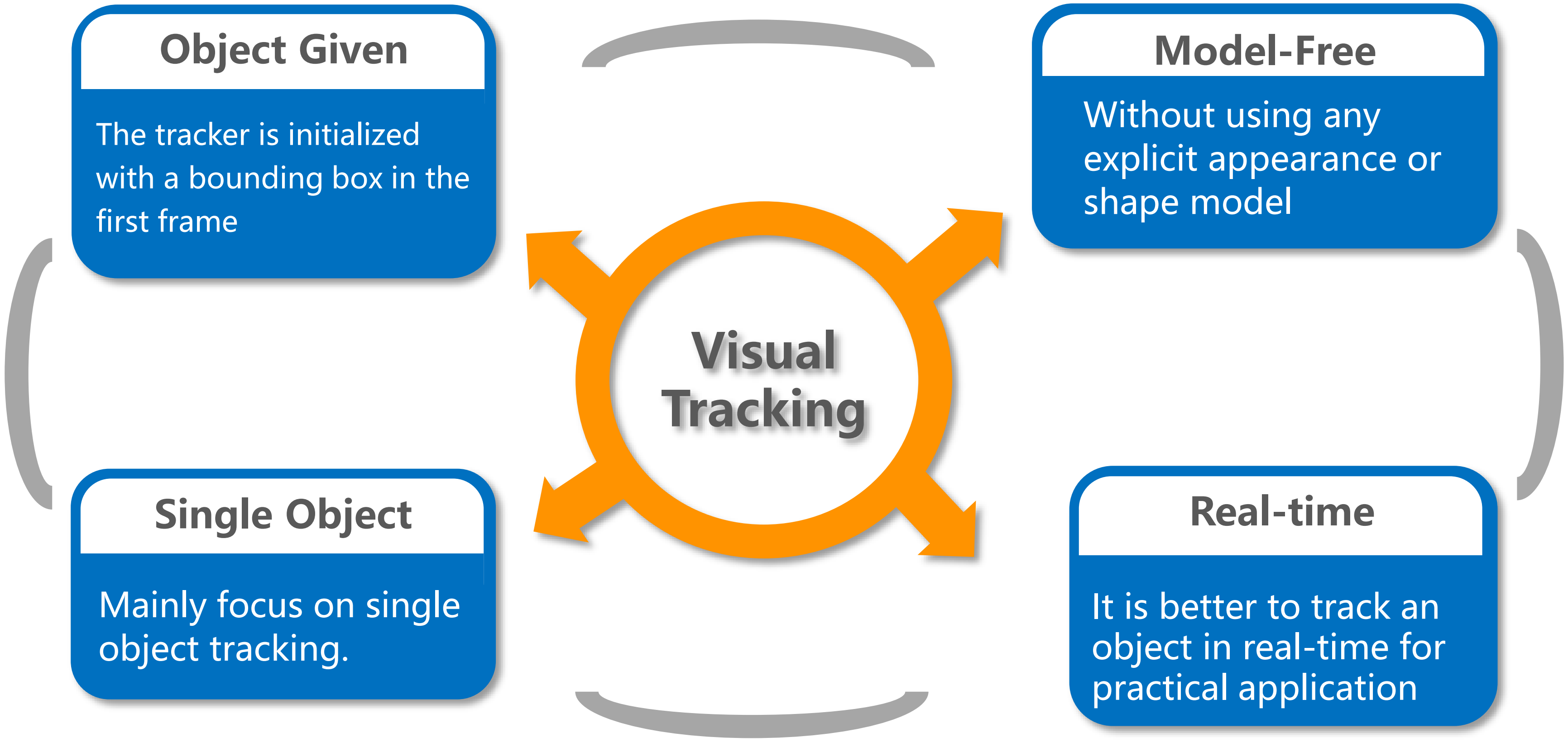
- CF based trackers
  - KCF (João F. Henriques) High-Speed Tracking with **K**ernelized **C**orrelation **F**ilters
  - LMCF (Our work) **L**arge **M**argin Object Tracking with **C**irculant Feature **M**aps
- CNN based trackers
  - SiamFC (Hyeonseob Nam) **F**ully-**C**onvolutional Siamese Networks for Object Tracking
  - ECO (Martin Danelljan) **E**fficient **C**onvolution **O**perators for Tracking

Summary &Tips

PART ONE 01

Overview

What is visual tracking?



01 PART ONE

Overview

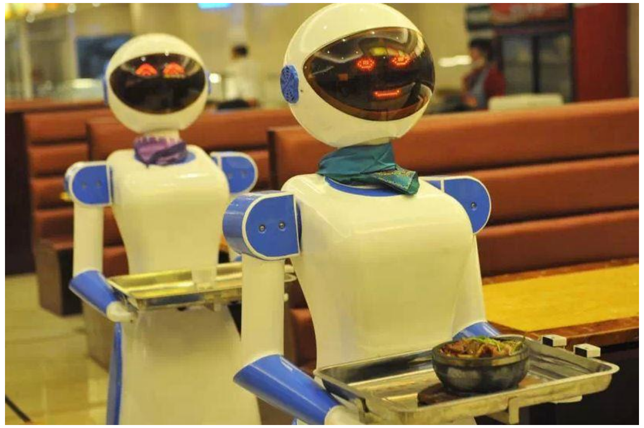
What is visual tracking?

Where is the baby?

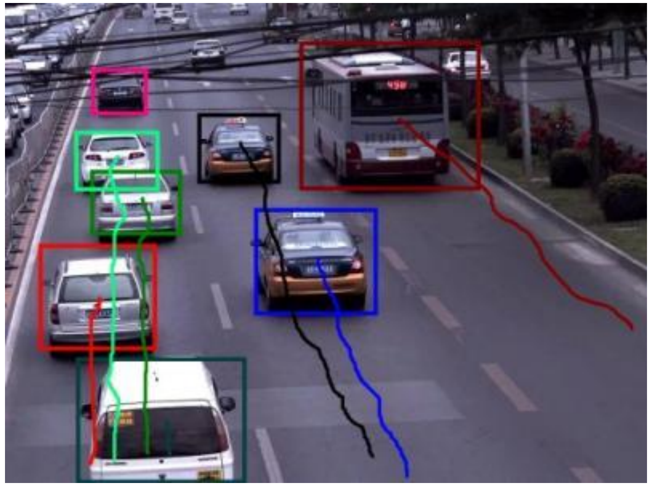
01 PART ONE

Overview

What is visual tracking?

1


2


3


4

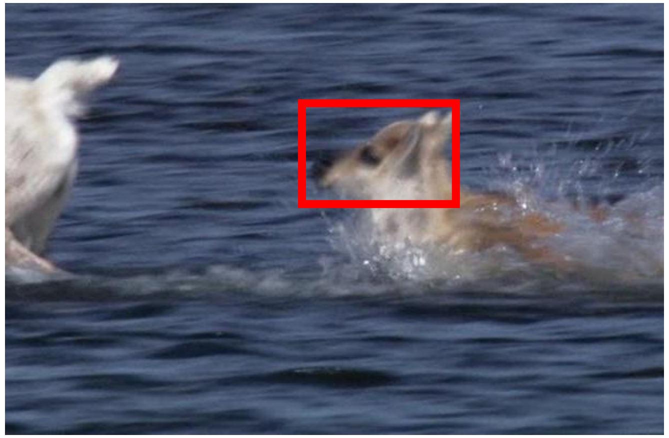

1. Service robot
2. Intelligent monitoring
3. Intelligent Transportation
4. Human-computer interaction

01
PART ONE

Overview

What is visual tracking?

1



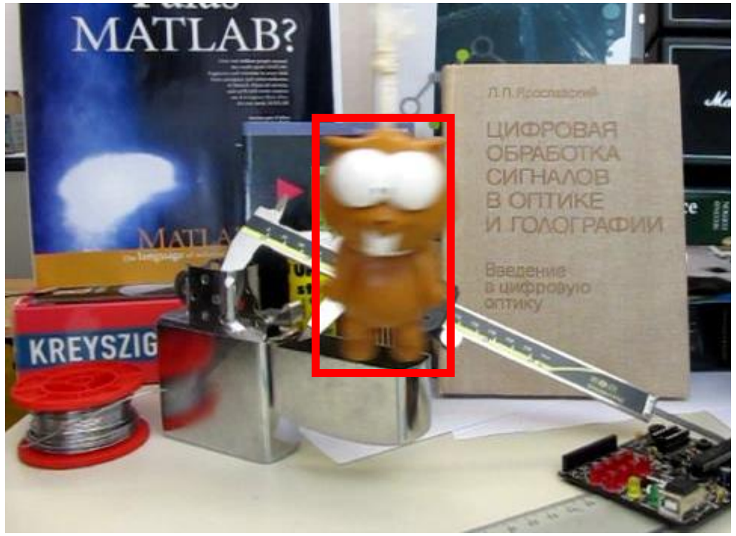
2

3

1 Motion Blur
2 Occlusion
3 Deformation
4 Scale Variation

4

→



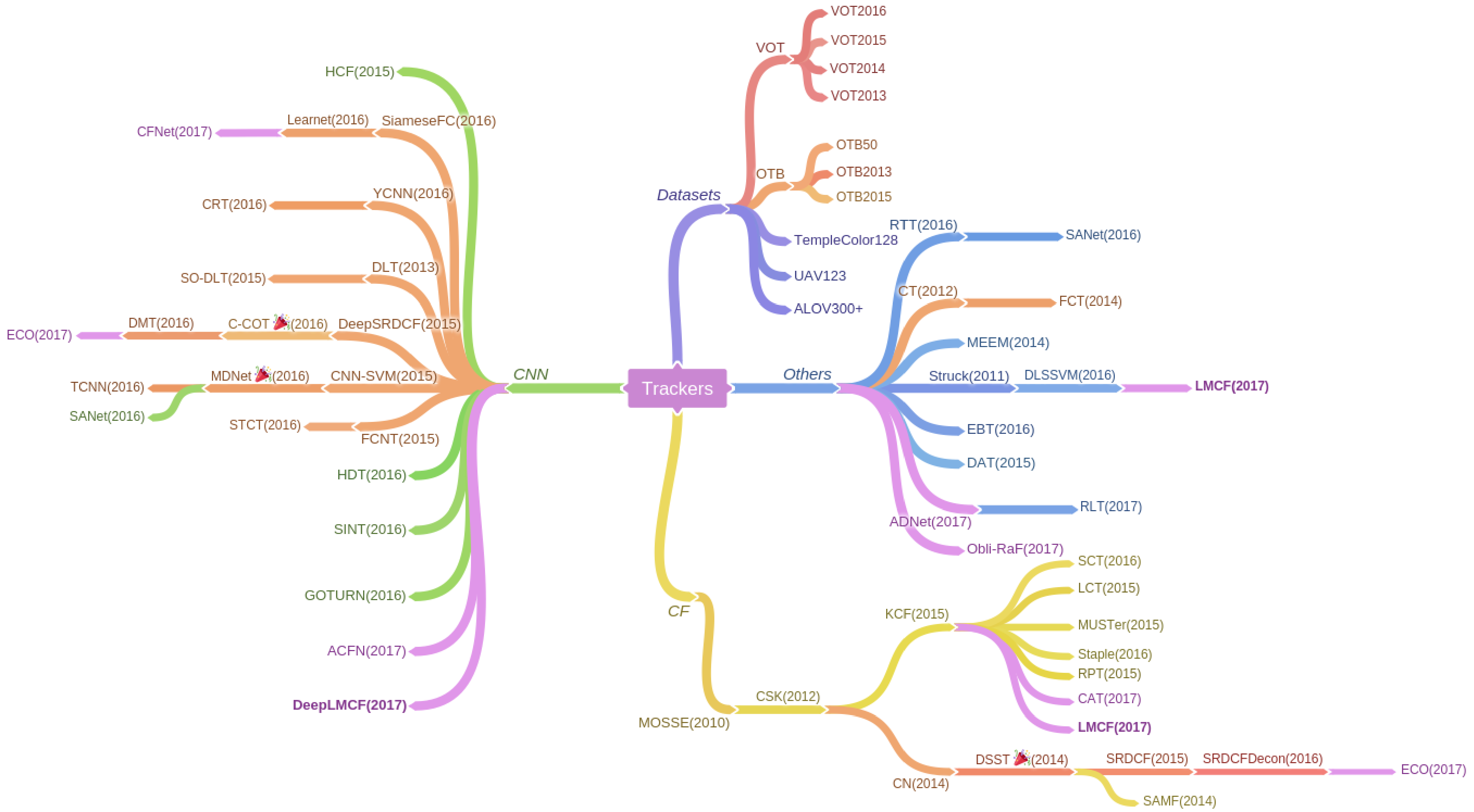
Difficulty

A lot of challenging
Lack of training samples

01 PART ONE

Overview

Existing Trackers



Typical Trackers

KCF: High-Speed Tracking with Kernelized Correlation Filters

- Cross-Correlation

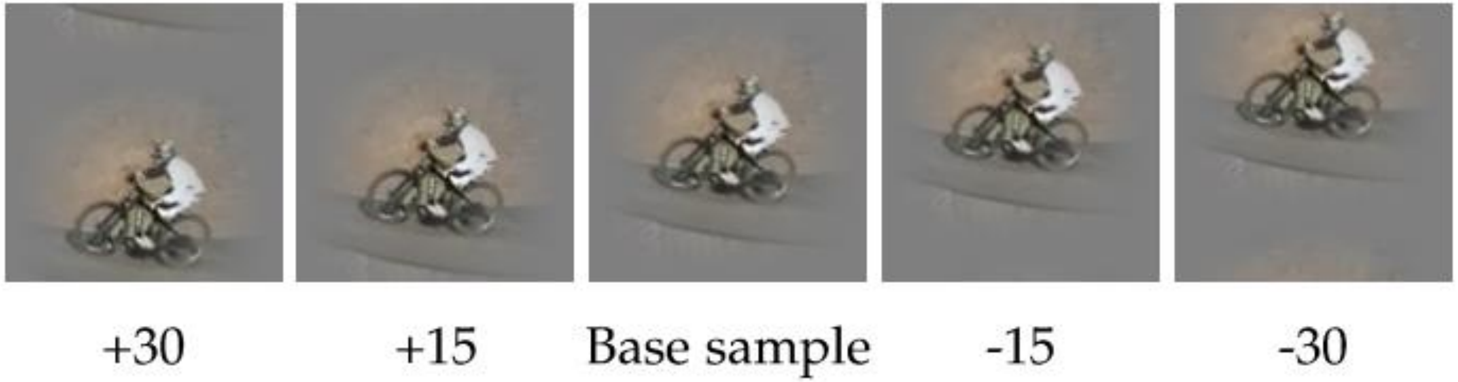
$$(f \star g)(\tau) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} f^*(t)\, g(t+\tau)\, dt$$

A measure of similarity of two series as a function of the displacement of one relative to the other.

- Circulant Matrices

All circulant matrices are made diagonal by the Discrete Fourier Transform (DFT).

$$X = C(x) = F \cdot diag(\hat{x}) \cdot F^H$$



$$\mathcal{F}(Xy) = \mathcal{F}(C(x)y) = \mathcal{F}(\bar{x} * y) = \mathcal{F}^*(x) \odot \mathcal{F}(y)$$

$$X^H X = F \cdot diag(\hat{x} \odot \hat{x}^*) \cdot F^H = C\left(\mathcal{F}^{-1}(\hat{x} \odot \hat{x}^*)\right)$$

$O(K^3) \rightarrow O(K\log K)$

Typical Trackers

KCF: High-Speed Tracking with Kernelized Correlation Filters

- Training →Ridge regression

Linear: $f(\mathbf{z}) = \mathbf{w}^T\mathbf{z}$ $\qquad \min_{\mathbf{w}} \sum_i (f(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{w}\|^2$

Solution (closed): $\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$ $\Rightarrow$ $\boxed{\mathcal{F}(w) = \frac{\hat{x}}{\hat{x} \odot \hat{x}^* + \lambda\delta} \odot \mathcal{F}(y) = \frac{\hat{x} \odot \hat{y}}{\hat{x} \odot \hat{x}^* + \lambda\delta}}$

Non-linear: $\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i)$ $\qquad f(\mathbf{z}) = \mathbf{w}^T\mathbf{z} = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{z}, \mathbf{x}_i)$

Solution (closed):

$\boldsymbol{\alpha} = (K + \lambda I)^{-1} \mathbf{y}$ $\qquad K_{ij} = \kappa(x_i, x_j)$ $\Rightarrow$ $\boxed{\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda}}$

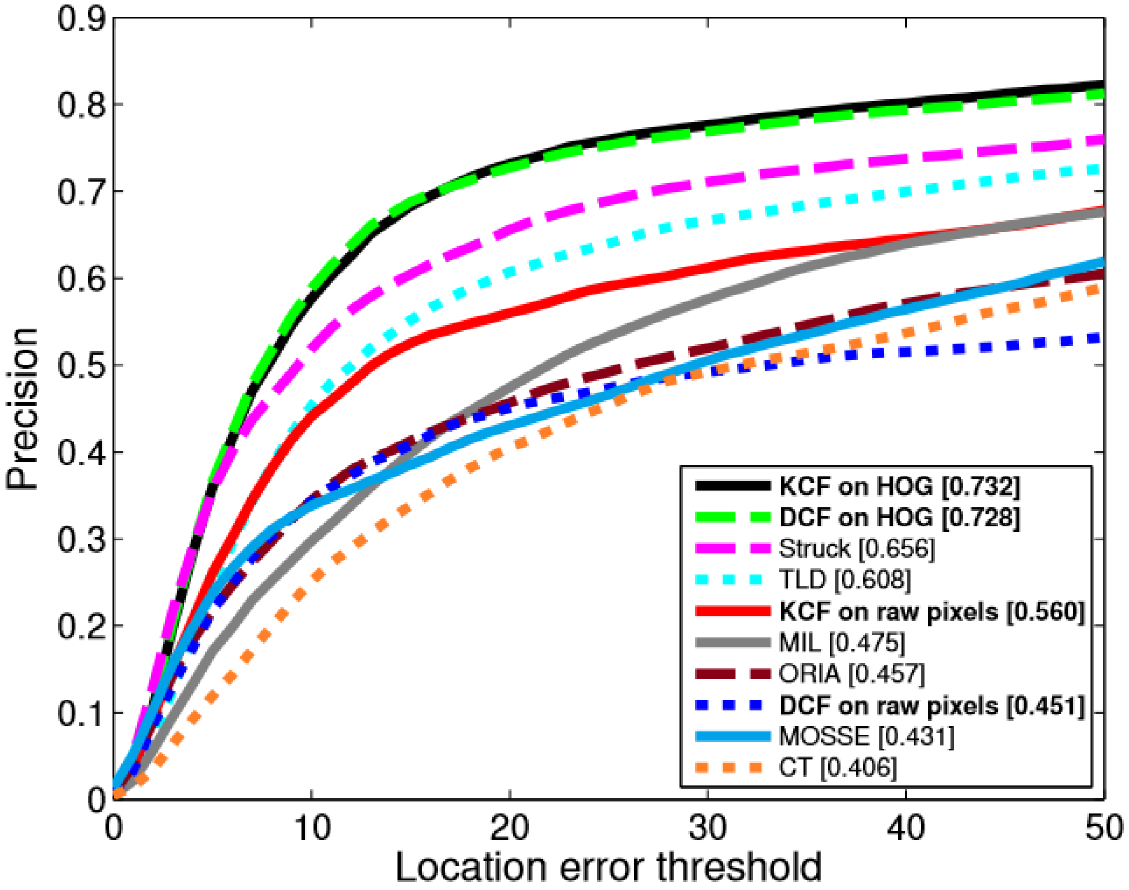
- Detection: $\mathbf{f}(\mathbf{z}) = (K^{\mathbf{z}})^T \boldsymbol{\alpha}$ $\Rightarrow$ $\boxed{\hat{\mathbf{f}}(\mathbf{z}) = \hat{\mathbf{k}}^{\mathbf{xz}} \odot \hat{\boldsymbol{\alpha}}}$

02 PART TWO

Typical Trackers

KCF: High-Speed Tracking with Kernelized Correlation Filters

- Experiment: OTB2013



	Algorithm	Feature	Mean precision (20 px)	Mean FPS
Proposed	KCF	HOG	**73.2%**	172
	DCF		**72.8%**	**292**
	KCF	Raw pixels	56.0%	154
	DCF		45.1%	278
Other algorithms	Struck [7]		65.6%	20
	TLD [4]		60.8%	28
	MOSSE [9]		43.1%	**615**
	MIL [5]		47.5%	38
	ORIA [14]		45.7%	9
	CT [3]		40.6%	64

02 PART TWO

Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

Motivation

1. Framework: Structured output SVM based tracking algorithms have shown favorable performance while limited by the time-consuming candidate sampling and complex optimization.
2. Forward Tracking: uncontrolled while decisive
3. Model update: significant while time-consuming

Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

1 Framework Structured output SVM

Structured output SVM is a kind of classification algorithm which can deal with complex outputs like trees, sequences, or sets rather than class labels.

Input: $\mathbf{x} \in X$

Output: $Y = \{(w,h) | w \in \{0,...,W-1\}, h \in \{0,...,H-1\}\}$

- *All the cyclic shifts of the image patch centered around the target are considered as the training samples* $(\mathbf{x}, \mathbf{y}_{w,h})$

02 PART TWO

Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

1 Framework Structured output SVM

- Objective function:

$$f\left(\mathbf{x};\mathbf{w}\right) = \underset{\mathbf{y}\in Y}{\arg\max}\, F\left(\mathbf{x},\mathbf{y};\mathbf{w}\right)$$

$$F\left(\mathbf{x},\mathbf{y};\mathbf{w}\right) = \left\langle \mathbf{w}, \Psi\left(\mathbf{x},\mathbf{y}\right)\right\rangle$$

- Optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{w=1}^{W-1}\sum_{h=1}^{H-1}\xi_{w,h}^2$$

$$\text{s.t.}\forall w, \forall h, \forall \mathbf{y}_{w,h} \in Y\backslash \mathbf{y}_{0,0}:$$

$$F\left(\mathbf{x},\mathbf{y}_{0,0};\mathbf{w}\right) - F\left(\mathbf{x},\mathbf{y}_{w,h};\mathbf{w}\right) \geqslant \sqrt{\Delta\left(\mathbf{y}_{0,0},\mathbf{y}_{w,h}\right)} - \xi_{w,h}$$

$$\hat{\mathbf{w}} = \frac{\hat{\Psi}^*\left(\mathbf{x},\mathbf{y}_0\right)\circ\hat{\mathbf{u}}^T}{\hat{\Psi}^*\left(\mathbf{x},\mathbf{y}_0\right)\circ\hat{\Psi}\left(\mathbf{x},\mathbf{y}_0\right)+\frac{1}{2C}}$$

$$\hat{\alpha} = \frac{\hat{\mathbf{u}}^T}{\hat{\mathbf{k}}^{\Psi_0\Psi_0}+\frac{1}{2C}}$$

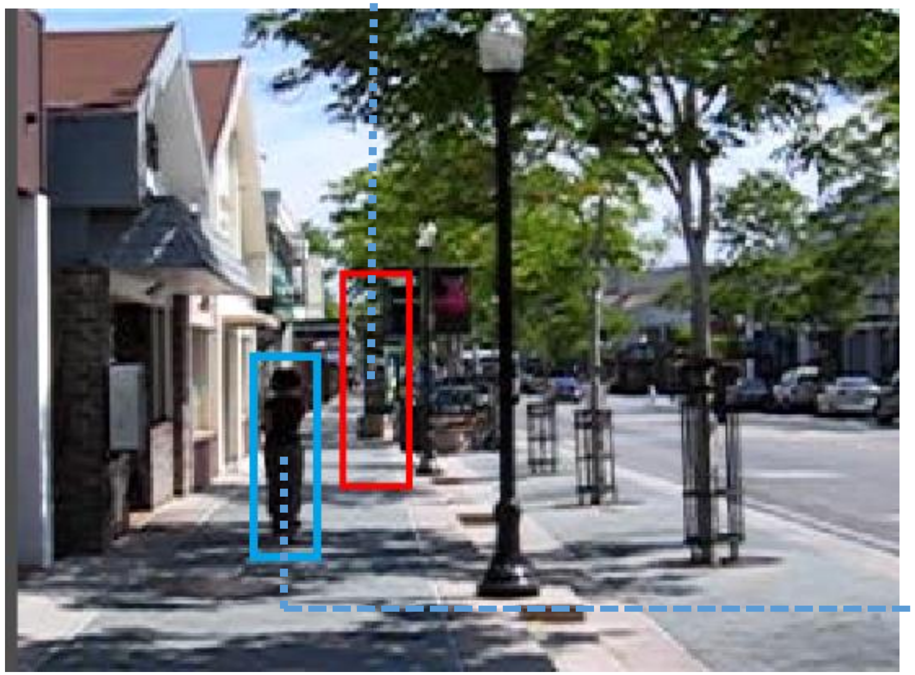
02 PART TWO

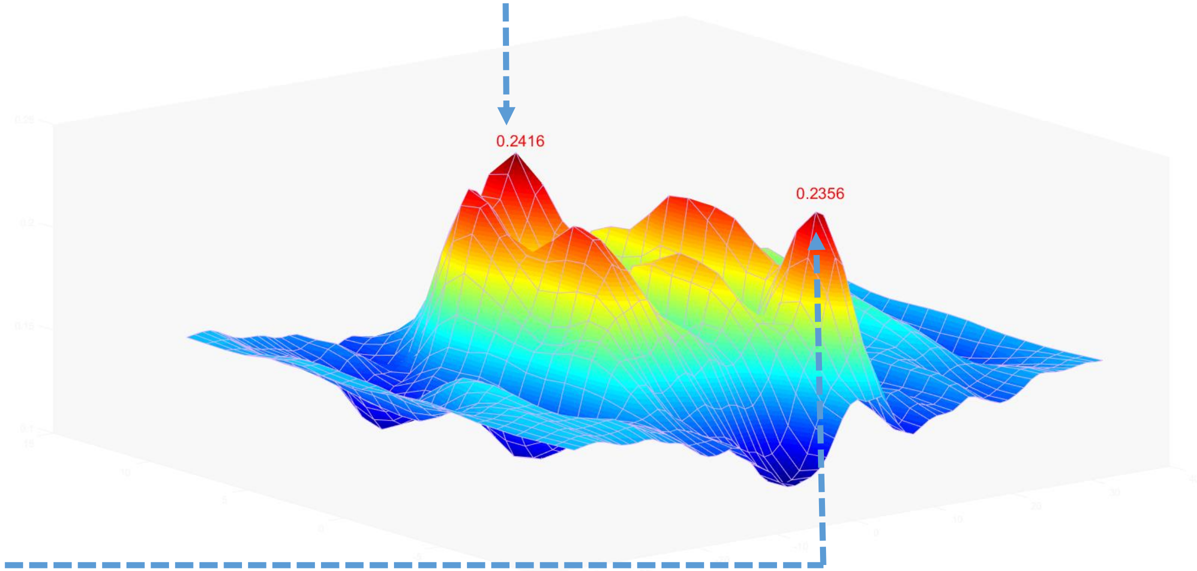
Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

② Forward Tracking

Unimodal: $F(\mathbf{s}, \mathbf{y}; \mathbf{w}) = \mathcal{F}^{-1}\left(\hat{\Psi}^{*}_{\mathbf{s}0} \circ \hat{\mathbf{w}}\right) = \mathcal{F}^{-1}\left(\hat{\mathbf{k}}^{\Psi_{\mathbf{x}0}\Psi_{\mathbf{s}0}} \circ \hat{\alpha}\right)$





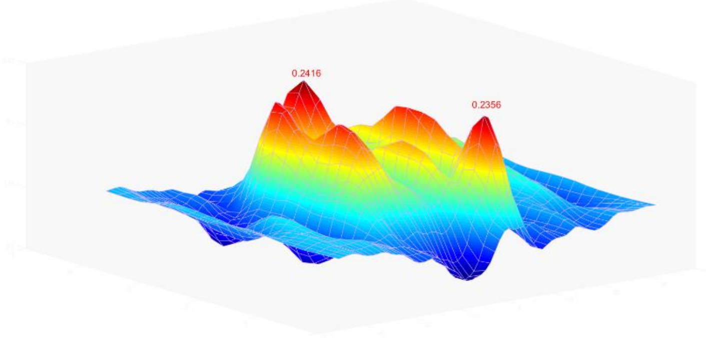
02 PART TWO

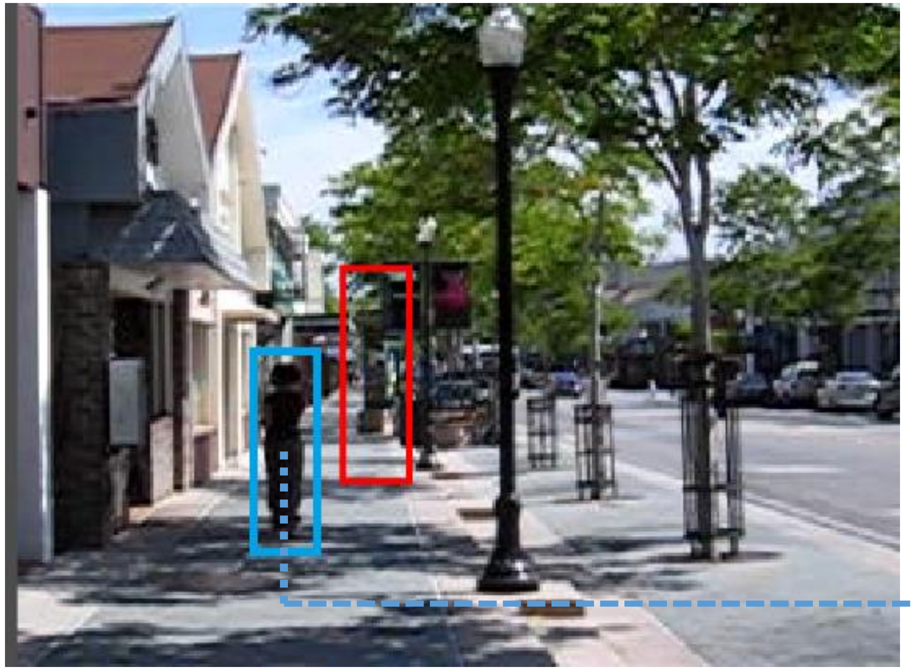
Typical Trackers

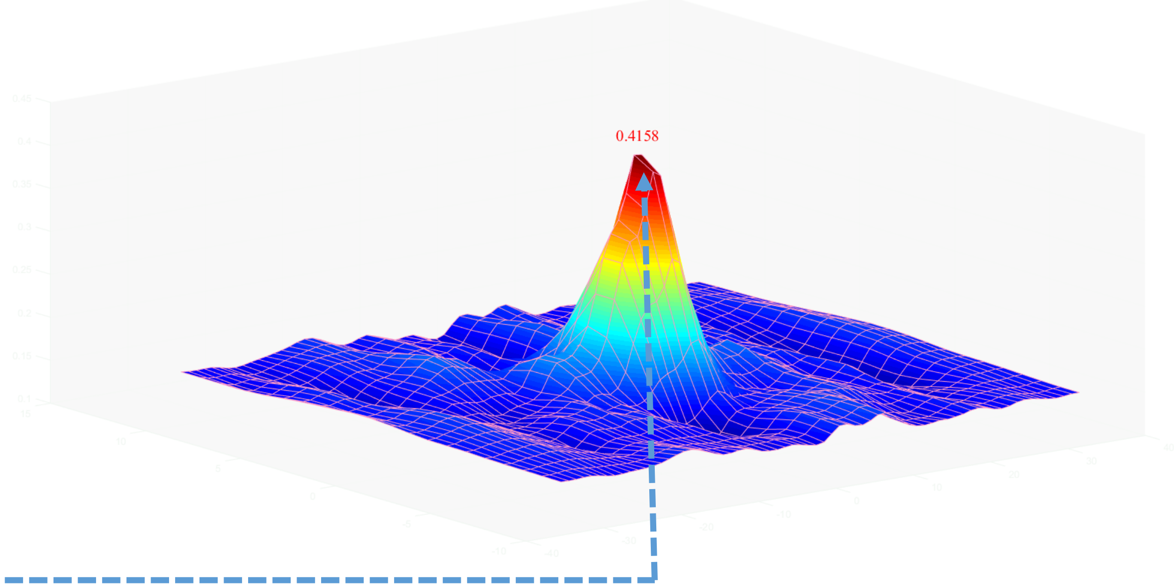
Large Margin Object Tracking with Circulant Feature Maps

② **Forward Tracking** Multimodal target tracking

Multimodal: $P(\mathbf{s}) = F(\mathbf{s}, \mathbf{y}; \mathbf{w}) \circ \mathbf{B}$



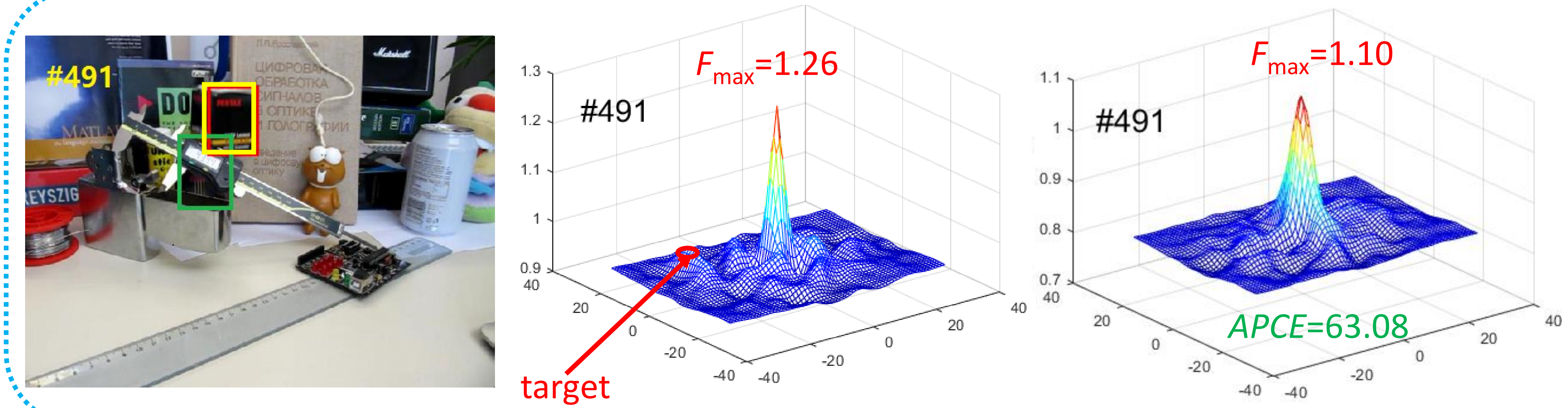




Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

③ Model update High-confidence Update



$$F_{\max} = \max F(\mathbf{s}, \mathbf{y}; \mathbf{w})$$

+

$$APCE = \frac{|F_{\max} - F_{\min}|^2}{mean\left(\sum_{w,h} (F_{w,h} - F_{\min})^2\right)}$$

02 PART TWO

Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

Experiments

- Analyses of LMCF:

Trackers	multimodal detection	high-confidence update	feature representations	OPE		TRE		SRE		mean FPS
				precision	success	precision	success	precision	success	
LMCF-N2	No	No	conventional	0.799	0.586	0.813	0.612	0.740	0.540	60.74
LMCF-Uni	No	Yes	conventional	0.809	0.606	0.815	0.616	0.757	0.549	61.38
LMCF-NU	Yes	No	conventional	0.813	0.605	0.820	0.619	0.750	0.545	46.45
LMCF	Yes	Yes	conventional	0.839	0.624	0.829	0.625	0.760	0.552	85.23
DeepLMCF	Yes	Yes	deep CNNs	0.892	0.643	0.877	0.649	0.850	0.596	8.11

LMCF-Uni: Without multimodal detection
LMCF-NU: Without high confidence update strategy
LMCF-N2: With neither of these two
LMCF: With both of these two
DeepLMCF: With CNN features

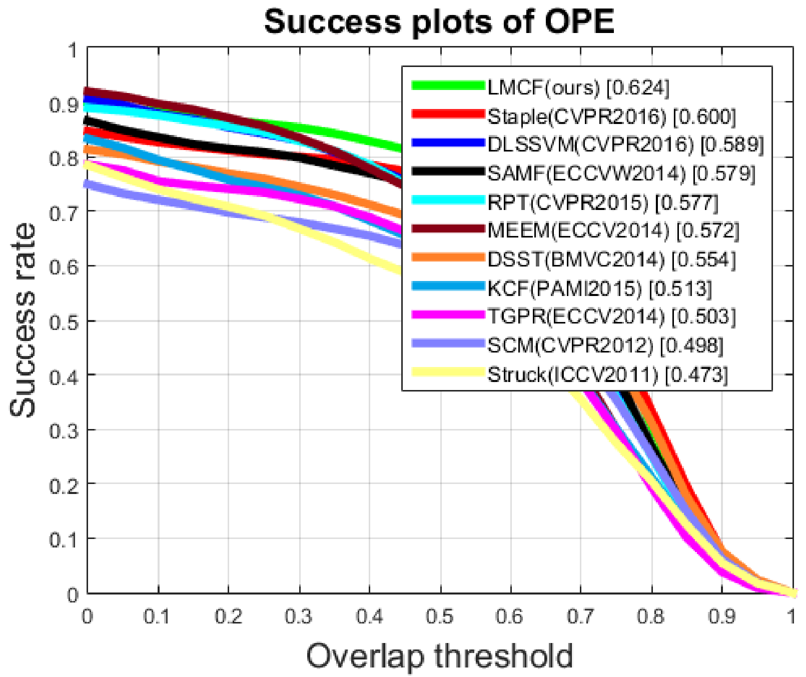
02 PART TWO

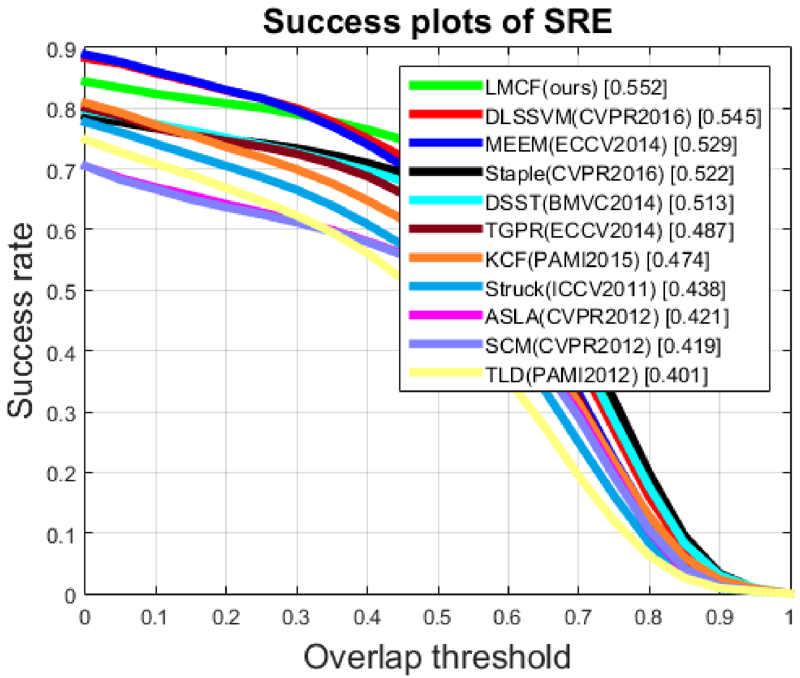
Typical Trackers

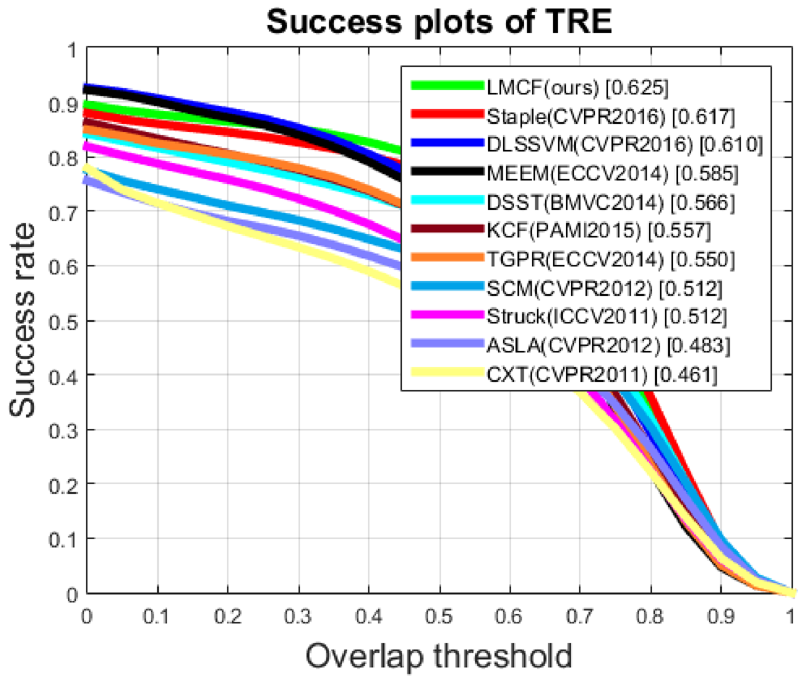
Large Margin Object Tracking with Circulant Feature Maps

Experiments

- LMCF-OTB2013 (80FPS)







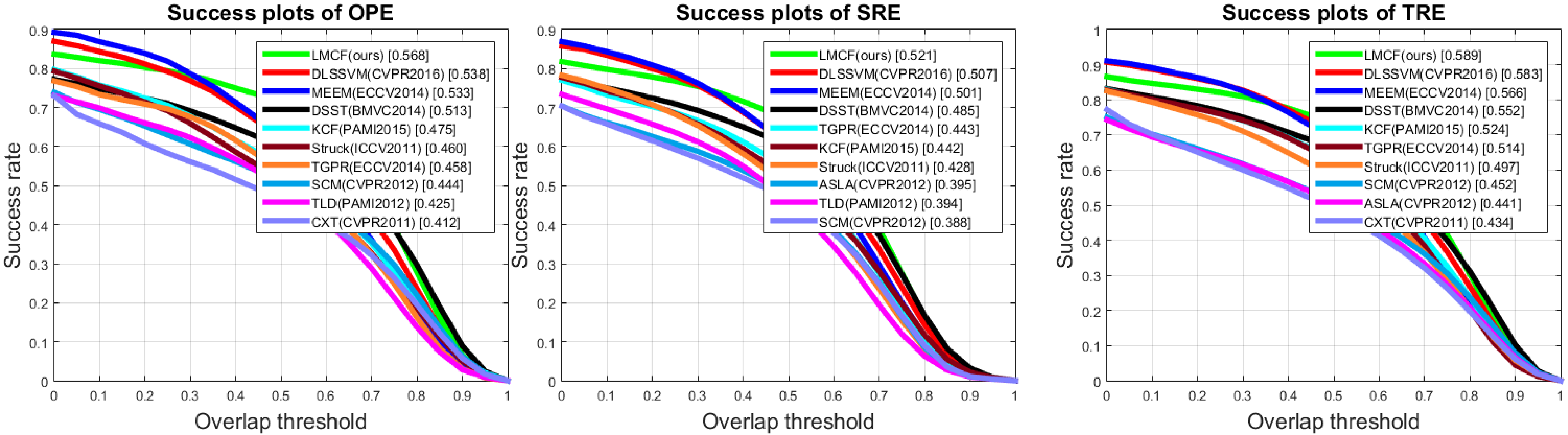
02 PART TWO

Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

Experiments

- LMCF-OTB2015 (80FPS)



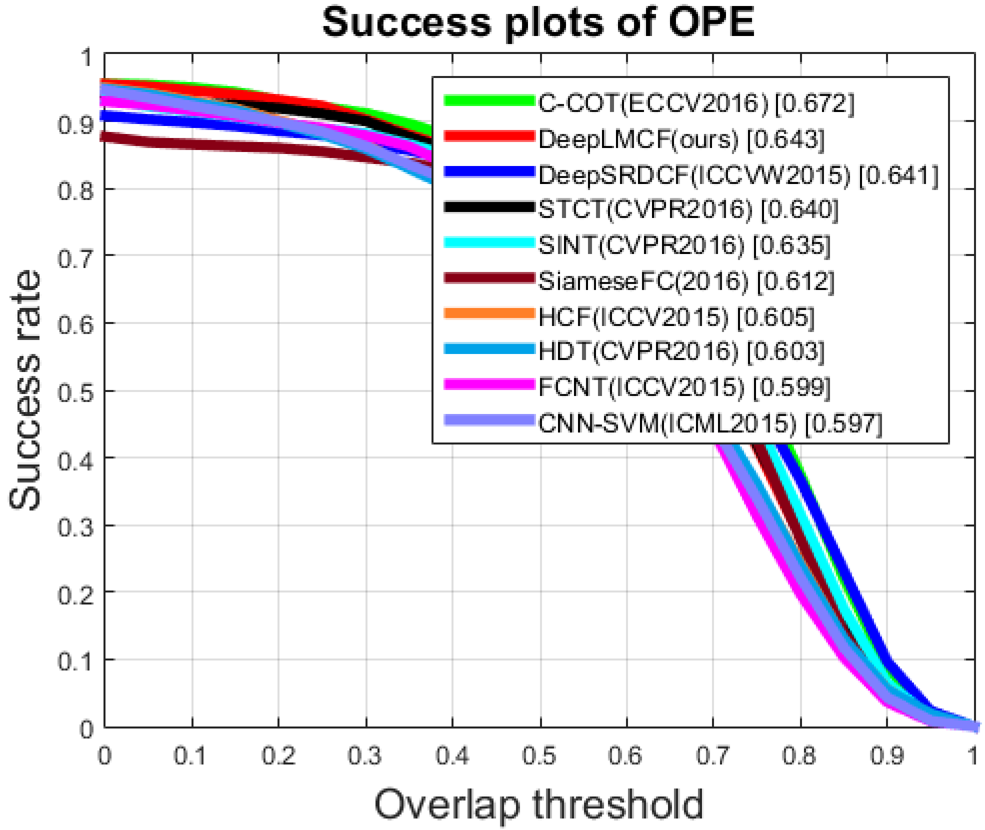
Typical Trackers

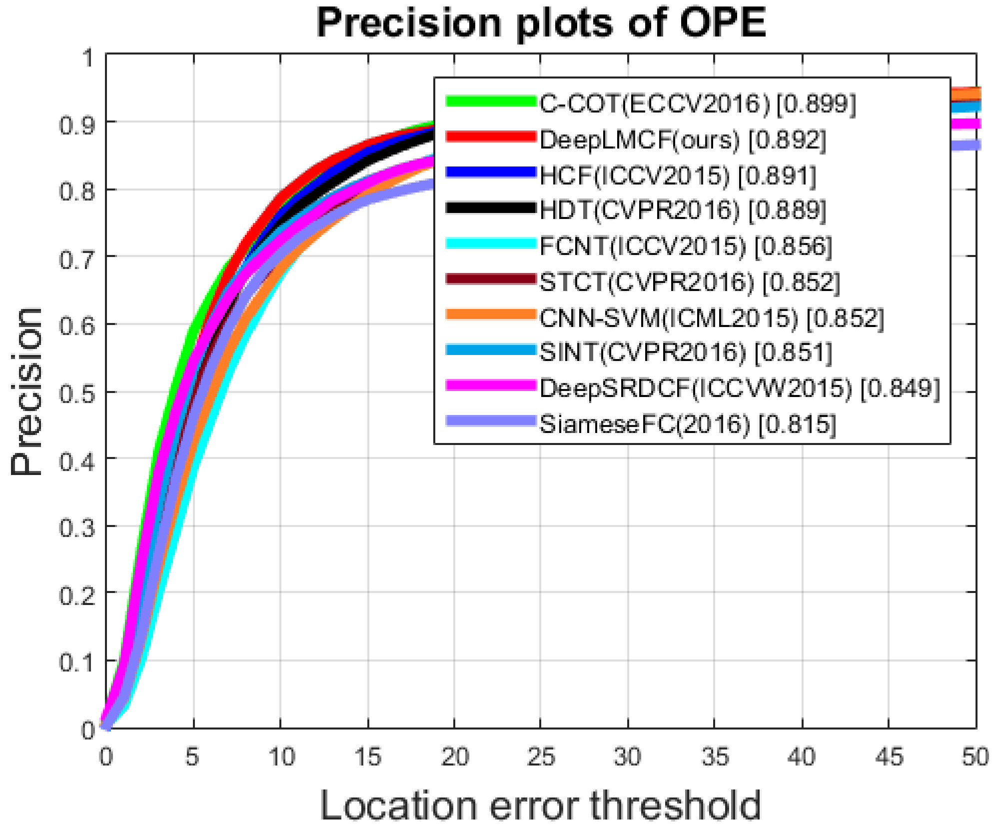
Large Margin Object Tracking with Circulant Feature Maps

PART TWO 02

Experiments

- DeepLMCF-OTB2013 (10FPS)





Typical Trackers

Large Margin Object Tracking with Circulant Feature Maps

Conclusion

- Powerful classifier
- Multimodal target tracking
- High-confidence model update

Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Motivation

For Deep Neural Network based methods, it is necessary to perform SGD online to adapt the weights of the network, severely compromising the speed of the system.

Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Contributions

1. Train a Siamese network to locate an exemplar image within a larger search image (similarity learning problem).

   Demonstrate that this approach achieves very competitive performance at speeds that far exceed the frame-rate requirement.

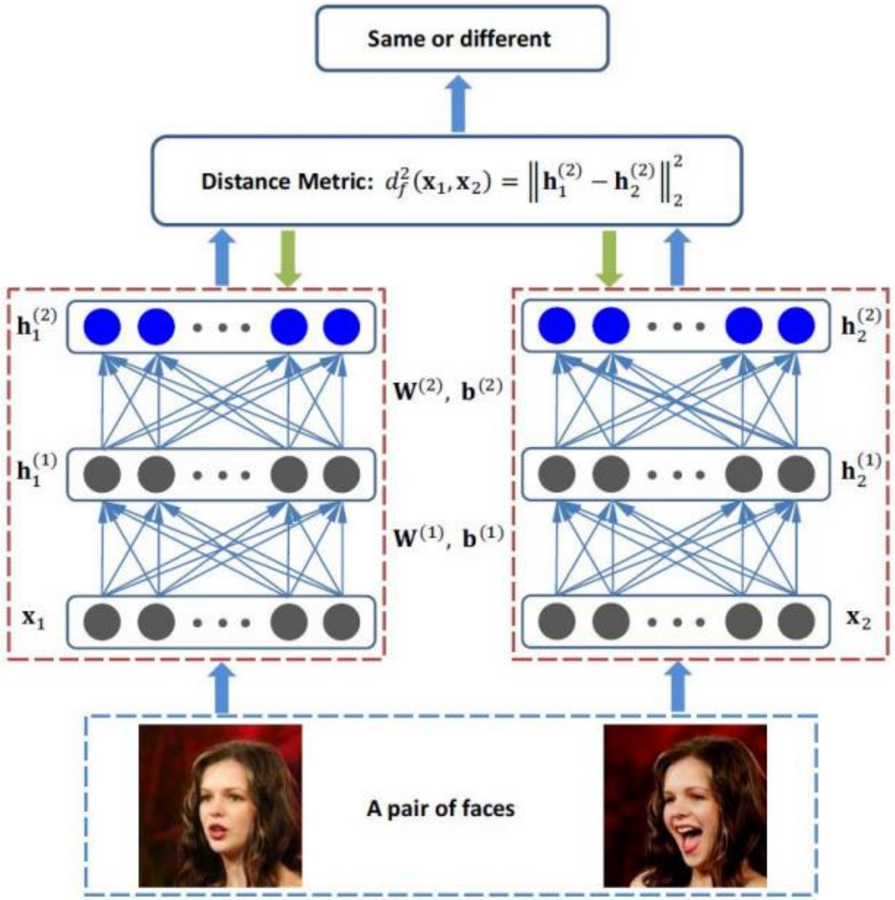
2. A novel architecture:

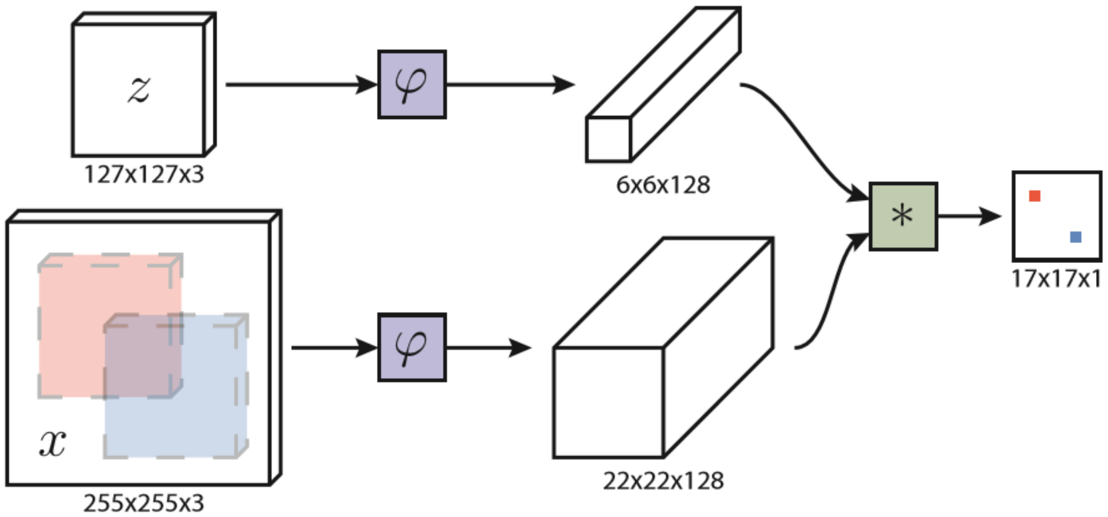
   Fully-convolutional with respect to the search image: dense and efficient sliding-window evaluation is achieved with a bilinear layer that computes the cross-correlation of its two inputs.

Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Fully-convolutional Siamese architecture





Embedding function resembles the conv of AlexNet.

02 PART TWO

Typical Trackers

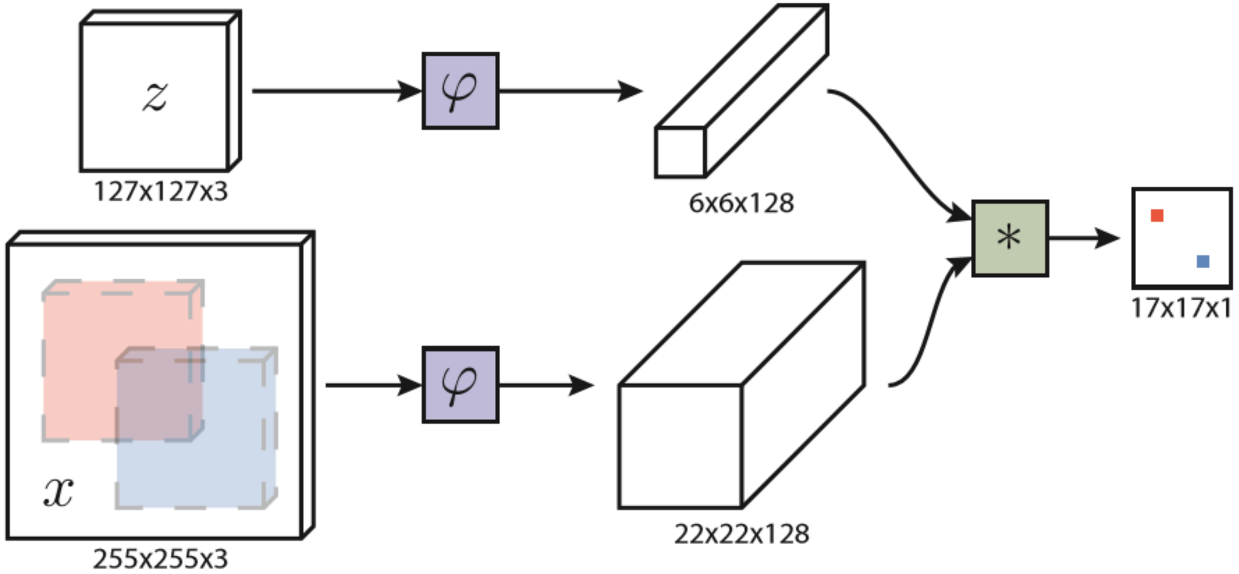
Fully-Convolutional Siamese Networks for Object Tracking

Fully-convolutional Siamese architecture

- All translated sub-windows
- A cross-correlation layer
- The position of the target (maximum score)
- Multiple scales (mini-batch, a single forward-pass)

Input: A single exemplar-candidate pair (z, x)

Output: Score map v



$$f(z, x) = \varphi(z) * \varphi(x) + b\,\mathbb{1}$$

02 PART TWO

Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Training

- Ground-truth label y:

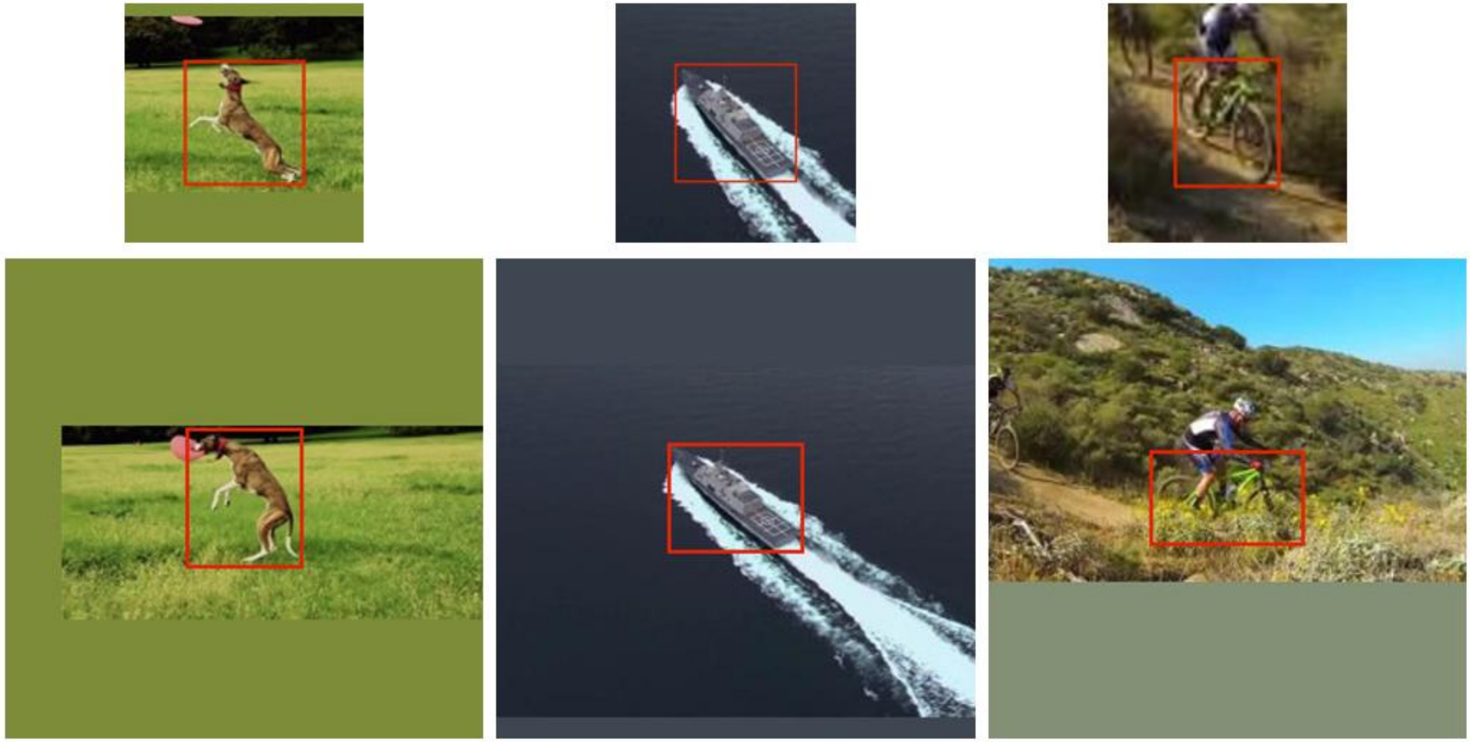
$$y[u] = \begin{cases} +1 & \text{if } k\|u - c\| \le R \\ -1 & \text{otherwise} \end{cases} .$$

- Loss of a score map v:

$$L(y, v) = \frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{D}} \ell(y[u], v[u])$$

- logistic loss:

$$\ell(y, v) = \log(1 + \exp(-yv))$$



ILSVRC 2015 object detection from video challenge
30 different classes of animals and vehicles, ~4500 videos

02 PART TWO

Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Tracking

Do not

- Update a model
- Maintain a memory of past appearances
- Incorporate additional cues
- Refine the prediction with bounding box regression

Do

- Search for the object within a region of approximately four times its previous size
- A cosine window is added to the score map to penalize large displacements
- Multiple scales are searched in a single forward-pass by assembling a mini-batch of scaled images.
- The embedding φ(z) of the initial object appearance is computed once in the first frame.
- Upsample the score map using bicubic interpolation, from 17×17 to 272×272

02 PART TWO

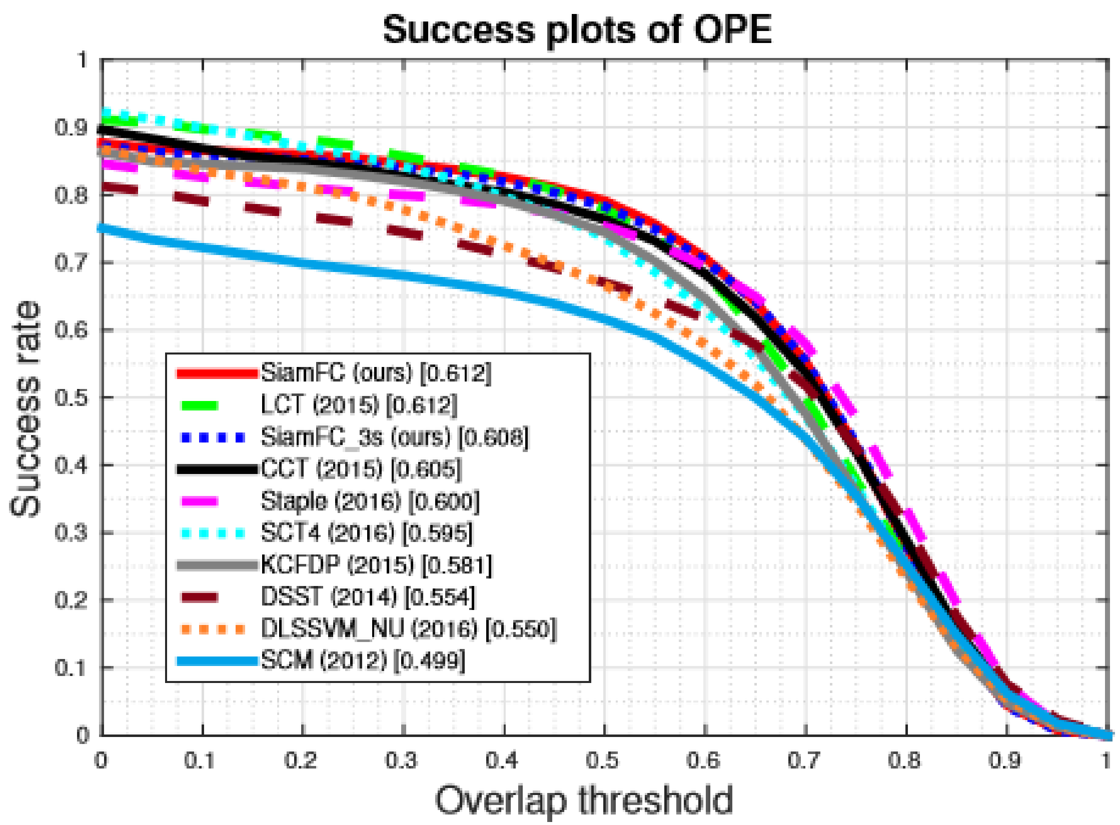
Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Experiments

- OTB2013

58FPS for 5 scales, 86FPS fps 3 scales.



02 PART TWO

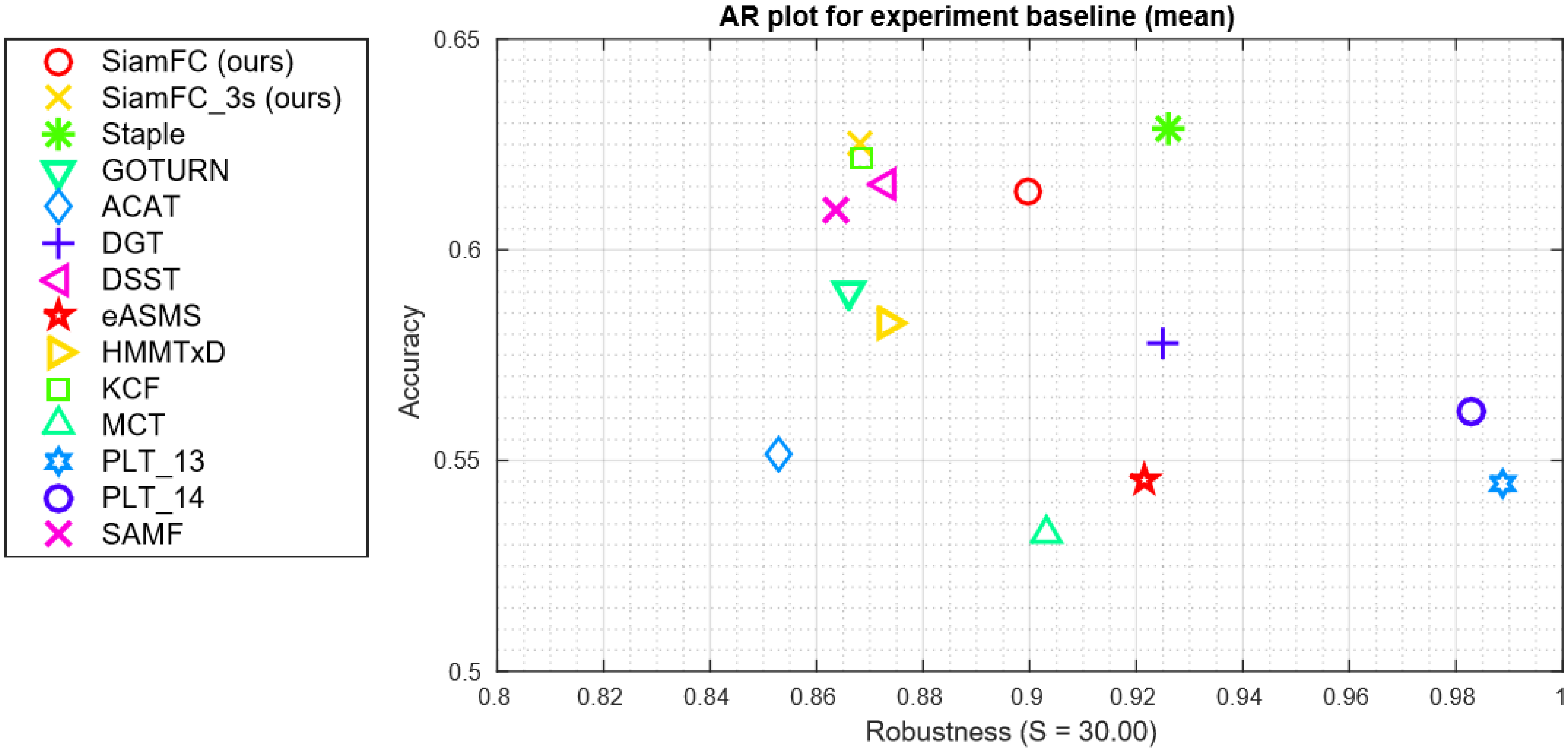
Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Experiments

- VOT2014

58FPS for 5 scales, 86FPS fps 3 scales.



Experiments

- Dataset size

This finding suggests that using a larger video dataset could increase the performance even further.

Table 3: Effects of using increasing portions of the ImageNet Video dataset on tracker's performance.

Dataset (%)	# videos	# objects	accuracy	# failures	expected avg. overlap
2	88	60k	0.484	183	0.168
4	177	110k	0.501	160	0.192
8	353	190k	0.484	142	0.193
16	707	330k	0.522	132	0.219
32	1413	650k	0.521	117	0.234
100	4417	2m	**0.524**	**87**	**0.274**

Typical Trackers

Fully-Convolutional Siamese Networks for Object Tracking

Conclusion

- ILSVRC 2015 (~4500 videos)
- Deep representation
- Large search regions

02 PART TWO

Typical Trackers

ECO: Efficient Convolution Operators for Tracking

Motivations

Focus on three key factors that contribute to both increased computational complexity and over-fitting in state-of-the-art DCF trackers.

1. Model size: a radical increase of the number of parameters in the appearance model, often beyond the dimensionality of the input.
2. Training set size: State-of-the-art DCF trackers, require a large training sample set to be stored due to their reliance on iterative optimization algorithms.
3. Model update strategy: Most DCF-based trackers apply a continuous learning strategy, where the model is updated rigorously in every frame.

02 PART TWO

Typical Trackers

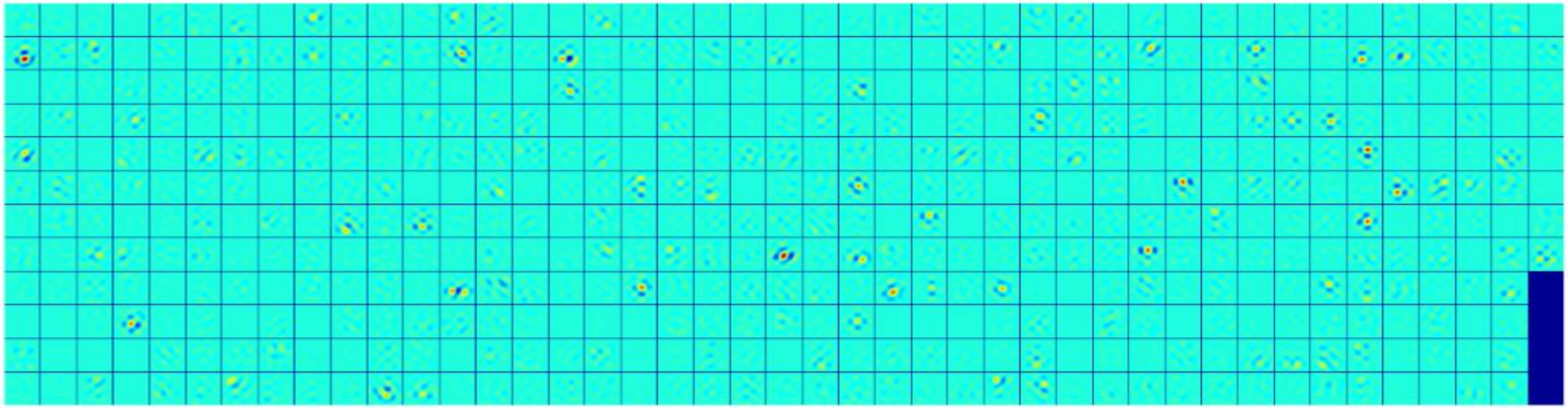
ECO: Efficient Convolution Operators for Tracking

① **Model size:** Factorized Convolution Operator

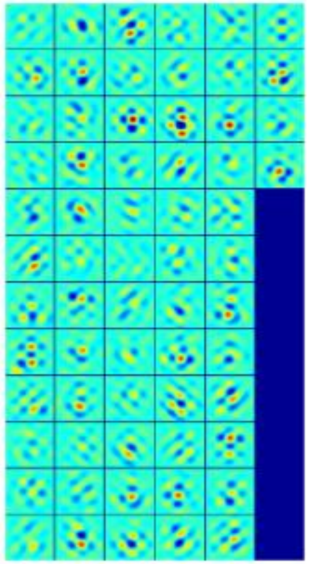
Baseline: C-COT

The feature map is transfered to the continuous spatial domain $t \in [0, T)$

$$J_d\{x^d\}(t) = \sum_{n=0}^{N_d-1} x^d[n] b_d\left(t - \frac{T}{N_d} n\right)$$



(a) C-COT



(b) Ours

02 PART TWO

Typical Trackers

ECO: Efficient Convolution Operators for Tracking

1 **Model size:** Factorized Convolution Operator D→C

Instead of learning one separate filter for each feature channel d, ECO use a smaller set of basis filters $f^1,\ldots,f^C$, C<D.

The filter for feature layer d is then constructed as a linear combination:

$$\sum_{c=1}^{C} p_{d,c} f^c$$

The factorized convolution formulation learns a compact set of discriminative basis filters with significant energy, achieving a radical reduction of parameters.

$$S_{Pf}\{x\} = Pf * J\{x\} = \sum_{c,d} p_{d,c} f^c * J_d\{x^d\} = f * P^{\mathrm{T}} J\{x\}$$

$$E(f,P) = \left\| \hat{z}^{\mathrm{T}} P \hat{f} - \hat{y} \right\|_{\ell^2}^2 + \sum_{c=1}^{C} \left\| \hat{w} * \hat{f}^c \right\|_{\ell^2}^2 + \lambda \|P\|_F^2.$$

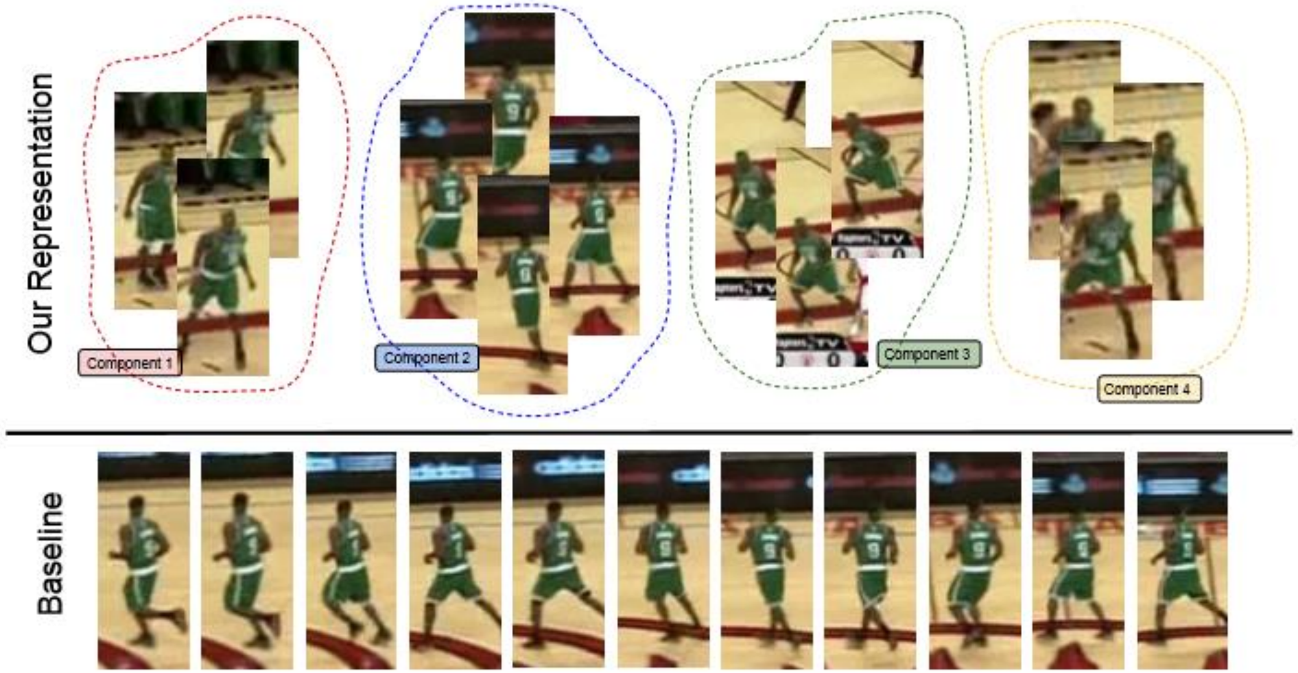
Gauss-Newton, Conjugate Gradient

02 PART TWO

Typical Trackers

ECO: Efficient Convolution Operators for Tracking

② **Training set size:** Generative Sample Space Model M→L



Model the training data as a mixture of Gaussian components, where each component represent a different aspect of the appearance.

$$E(f) = \sum_{l=1}^{L} \pi_l \left\| S_f\{\mu_l\} - y_0 \right\|_{L^2}^2 + \sum_{d=1}^{D} \left\| w f^d \right\|_{L^2}^2$$

PART TWO

Typical Trackers

ECO: Efficient Convolution Operators for Tracking

③ Model Update Strategy Ns = 6

Update the filter in every Ns frames.

Note that Ns does not affect the updating of the sample space model which is updated every frame.

VOT2016:

	Baseline C-COT (Sec. 2)	$\Longrightarrow$ Factorized Convolution (Sec. 3.1)	$\Longrightarrow$ Sample Space Model (Sec. 3.2)	$\Longrightarrow$ Model Update (Sec. 3.3)
EAO	0.331	0.335	0.351	**0.375**
FPS	0.3	1.1	2.6	6.0
Compl. change	-	$D \rightarrow C$	$M \rightarrow L$	$N_{CG} \rightarrow \frac{N_{CG}}{N_S}$
Compl. red.	-	$6\times$	$8\times$	$6\times$

CSC

02 PART TWO

Typical Trackers

ECO: Efficient Convolution Operators for Tracking

Experiments (8 FPS)

- VOT2016

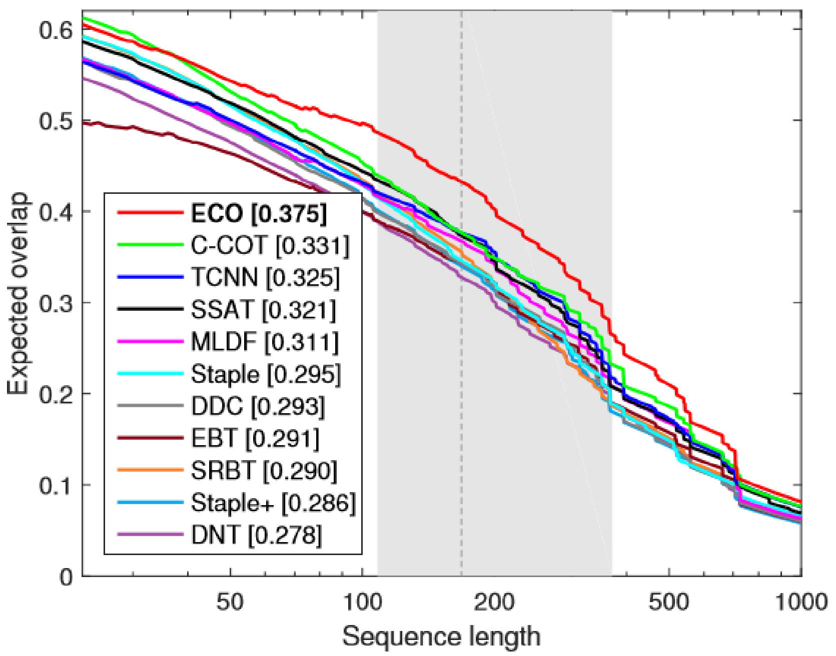


Figure 4. Expected Average Overlap (EAO) curve on VOT2016.

	DNT	Staple+	SRBT	EBT	DDC	Staple	MLDF	SSAT	TCNN	C-COT	ECO
EAO	0.278	0.286	0.290	0.291	0.293	0.295	0.311	0.321	0.325	*0.331*	**0.375**
Failure rate	1.18	1.32	1.25	0.90	1.23	1.35	*0.83*	1.04	0.96	0.85	**0.73**
Accuracy	0.50	*0.55*	0.50	0.44	0.53	0.54	0.48	**0.57**	0.54	0.52	0.53
EFO	1.127	**44.765**	3.688	3.011	0.198	*11.144*	1.483	0.475	1.049	0.507	4.530

Expected average overlap: how accurate the estimated bounding box is after a certain number of frames are processed since initialization.

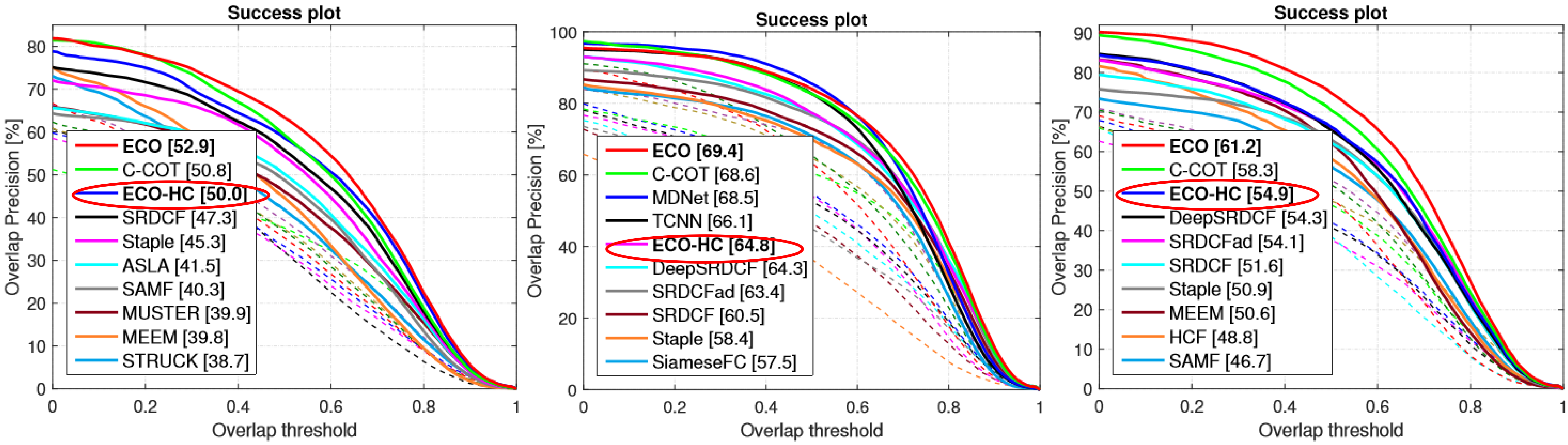
PART TWO 02

Typical Trackers

ECO: Efficient Convolution Operators for Tracking

Experiments

- UAV123, OTB100, Temple-Color
- ECO-HC (60 FPS)



(a) UAV123 (b) OTB-2015 (c) Temple-Color

Typical Trackers

ECO: Efficient Convolution Operators for Tracking

Conclusion：

- CNN+HOG+CN
- Powerful filters
- Training components (diversity)
- Update the filters every 6 frames

03 PART THREE

Summary &Tips

The most important factors

1. Feature representation: HOG, CN, CNN…
2. Classifier: Structured SVM, Ridge Regression, Deep Neural Networks…
3. Model update strategy: Fixed Interval, High-confidence Strategy…

PART THREE

Summary &Tips

Some tips about visual tracking

Tips 1

- CVPR, ICCV, ECCV, NIPS, ICML, BMVC
- João F. Henriques

Visual Geometry Group, University of Oxford

- Martin Danelljan

Computer Vision Laboratory, Linköping, Sweden

- Huchuan Lu

IIAU-Lab, Dalian University of Technology

- Bohyung Han

Computer Vision Laboratory, POSTECH, Korea

- Ming-Hsuan Yang

University of California at Merced

Tips 2

- Dataset: OTB, VOT, Temple-Color
- Foundations:

Naiyan Wang: Understanding and Diagnosing Visual Tracking Systems

João F. Henriques: High-Speed Tracking with Kernelized Correlation Filters

Hyeonseob Nam: Learning Multi-Domain Convolutional Neural Networks for Visual Tracking

- Matlab, python, C++
- Object detection
- https://github.com/foolwood/benchmark_results
- 知乎专栏： 目标跟踪算法

Thank you