


ECO: Efficient Convolution Operators for Tracking

Martin Danelljan   Goutam Bhat

Computer Vision Laboratory, Linköping University, Sweden

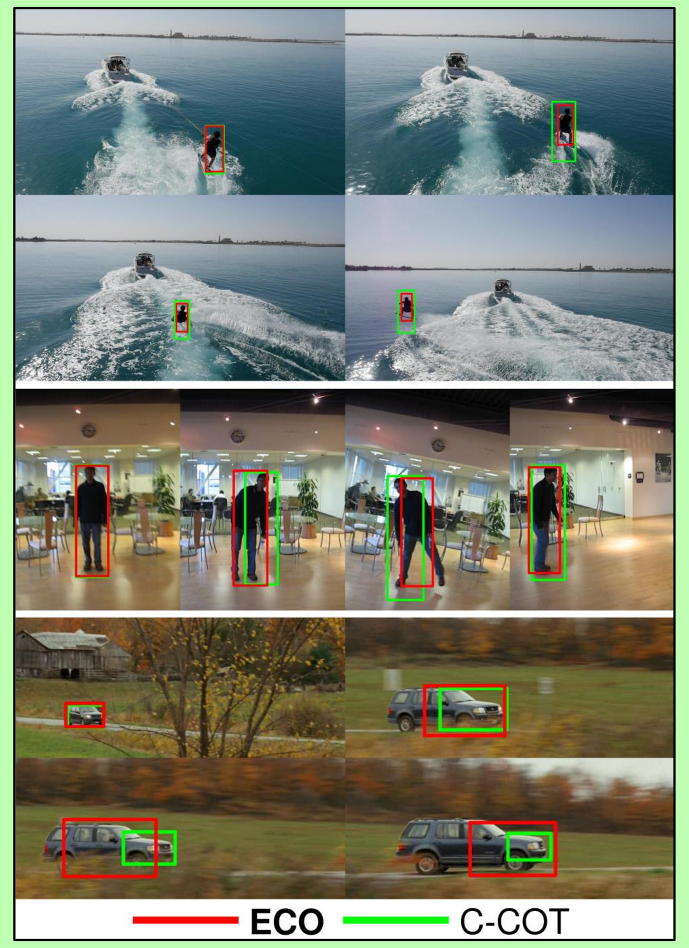
Fahad Khan   Michael Felsberg

LINKÖPING UNIVERSITY


Introduction

Discriminative Correlation Filter (DCF) Trackers: A historical comparison

	MOSSE [CVPR 2010]	**CCOT** [ECCV 2016]
Status	Pioneering work, but obsolete	State-of-the-art, winner of VOT2016
Image Features	Raw grayscale values	Conv layers from a CNN (and other)
Parameters	$\sim 10^3$	$\sim 10^6$
Speed	~1000 FPS	~1 FPS



Problem: Improved tracking performance at the cost of increased model size and complexity.

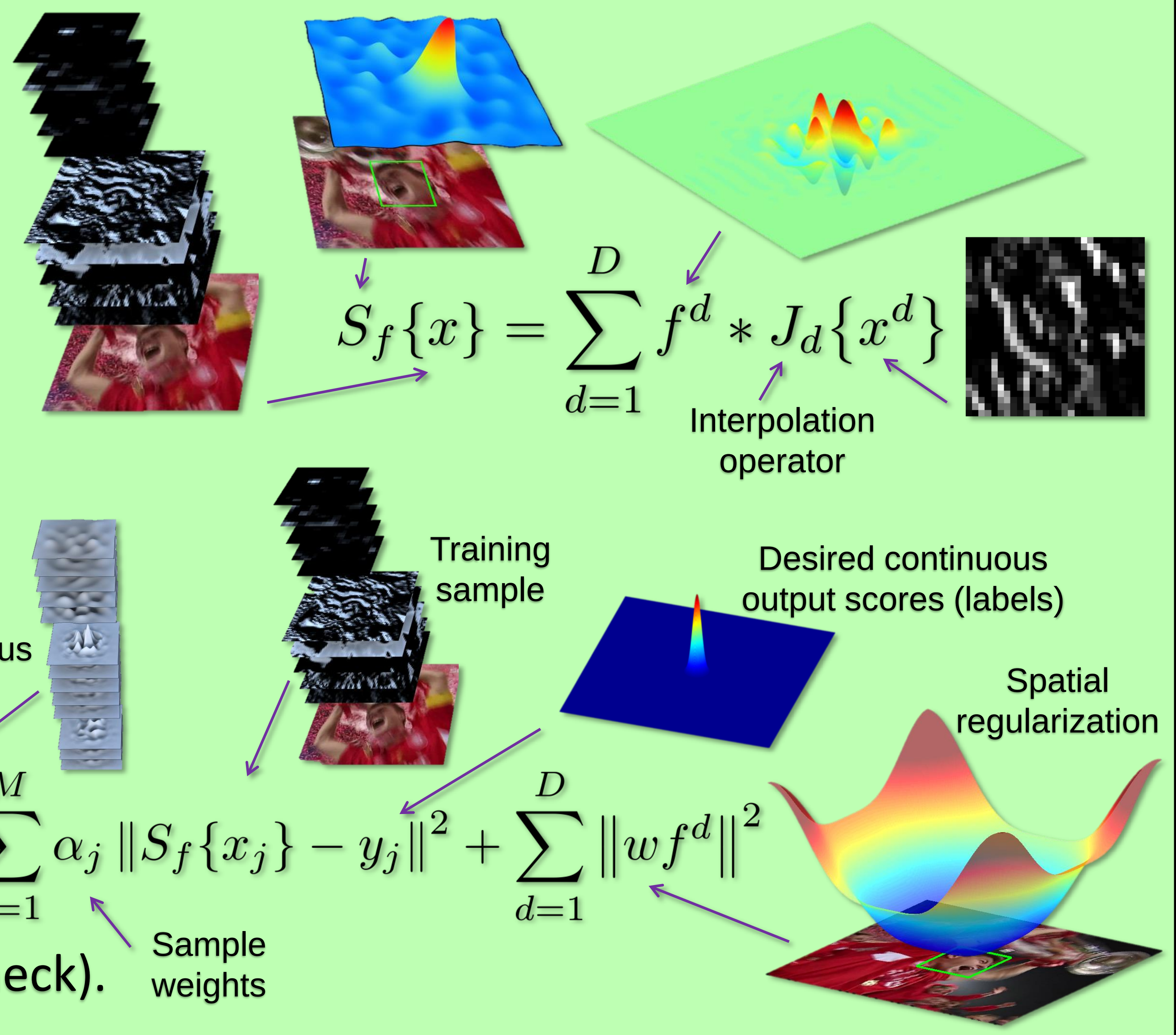
Consequences: (1) Slow tracking, **(2)** Overfitting

We address (1) computational complexity and **(2)** overfitting in state-of-the-art DCF trackers by
- Reducing the model size using factorized convolution
- Introducing a training set model that reduces its size and increases diversity
- Investigating the model update scheme, for better speed and robustness

Continuous Convolution Operator Tracker (CCOT) [1]

Convolution operator: Predicts the continuous detection scores of the target given a feature map x.

Training loss:
- Least squares regression.
- Optimized in the Fourier domain.
- Conjugate Gradient solver (computational bottleneck).



$$S_f\{x\} = \sum_{d=1}^{D} f^d * J_d\{x^d\}$$

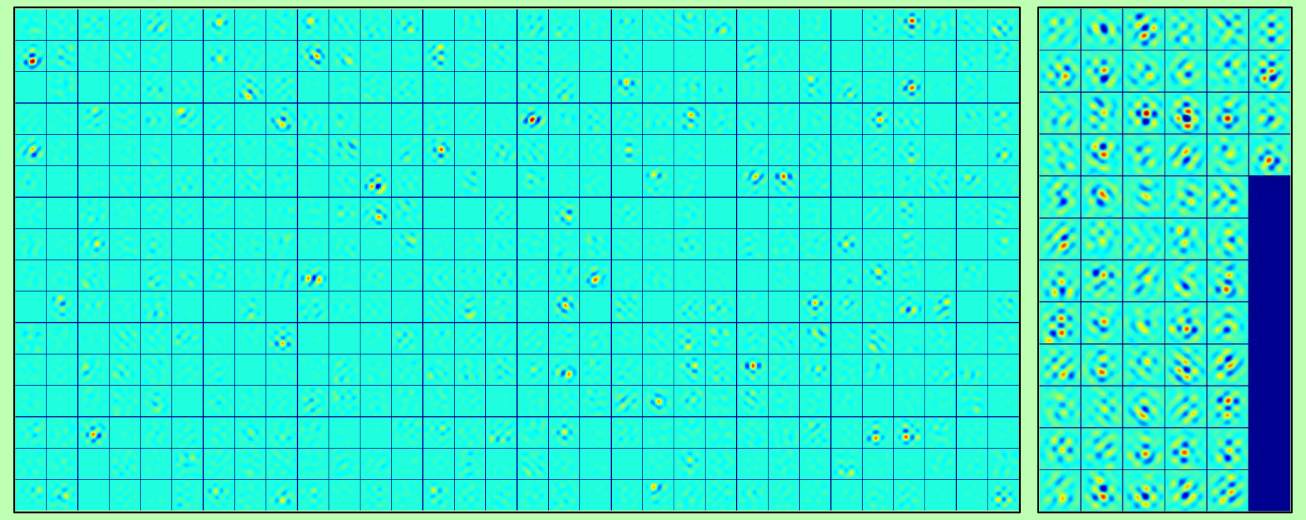
$$E(f) = \sum_{j=1}^{M} \alpha_j \|S_f\{x_j\} - y_j\|^2 + \sum_{d=1}^{D} \|w f^d\|^2$$

Our Approach

Factorized Convolution:

- **Previous Work:** Large number of excessive filters containing negligible energy (right).
- Leads to slower optimization and overfitting.
- **Our Method:** We learn a smaller set of filters f^c and a coefficient matrix $P = (p_{c,d})$.
- Factorized convolution operator:

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



CCOT filters   ECO filters

- We train f and P **jointly** by minimizing the regression loss in the first frame.
- The loss is optimized in the Fourier domain using Gauss-Newton and Conjugate Gradient.
- **Gain:** 6-fold reduction in number of filters.

	Conv-1	Conv-5	HOG	CN
Feature dim., D	96	512	31	11
Filter dim., C	16	64	10	3

Generative Sample Space Model:

- **Previous Work:** employ a fix learning rate $\alpha_j \sim (1-\gamma)^{-j}$.
- Oldest sample is replaced.
- Requires a large sample limit $M_{\max}$.
- Costly learning and poor diversity of training samples (see figure).
- **Our Method:** A Gaussian Mixture Model of the sample distribution

$$p(x) = \sum_{l=1}^{L} \pi_l \mathcal{N}(x; \mu_l, I)$$

- Updated using an efficient online algorithm [2].

- We optimize an approximate expected regression loss by replacing α_j and x_j with π_j and μ_j.
- **Gain:** 8-fold reduction in the number of training samples.

Model Update and Optimization Strategy

- **Previous Work:** Most DCF methods update the tracking model in each frame.
- In CCOT, a few (typically five) Conjugate Gradient (CG) iterations is performed each frame.
- **Our Method:** We only optimize every N_S frame for faster tracking.
- This also causes less overfitting to recent frames, leading to better tracking performance.
- We further propose to use the Polak-Ribière formula in CG for faster convergence.
- **Gain:** 6-fold reduction in the number of Conjugate Gradient iterations.

Experiments

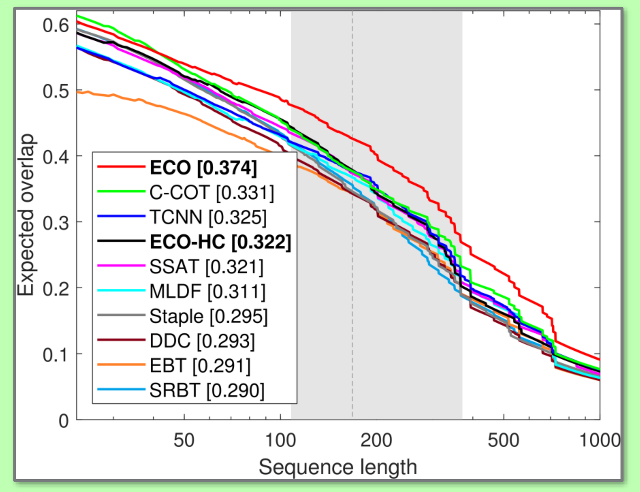
Baseline Comparison on VOT2016 dataset, deep feature version:

	Baseline C-COT	$\Longrightarrow$ Factorized Convolution	$\Longrightarrow$ Sample Space Model	$\Longrightarrow$ Model Update
EAO	0.331	0.342	0.352	0.374
FPS (CPU)	0.3	1.1	2.6	6.0
Compl. red.	-	6×	8×	6×

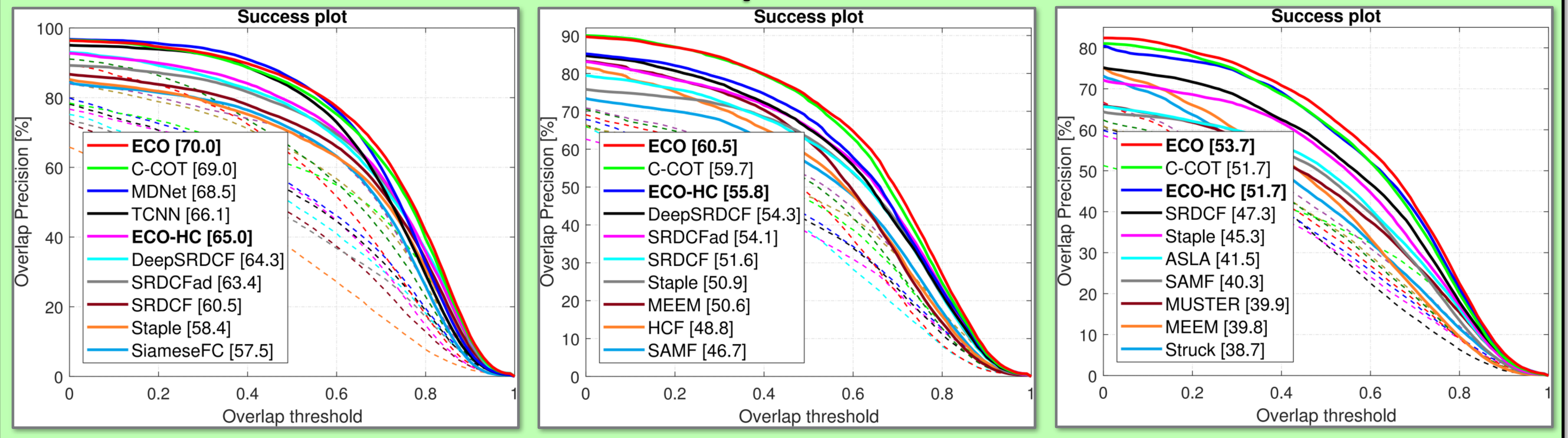
ECO:
- Deep features (VGG) + HOG
- **15 FPS** on GPU

ECO-HC:
- Hand-crafted features: HOG and CN
- **60 FPS** on CPU
- Optimal for **UAV** and other robotics applications

VOT2016



OTB-100   TempleColor   UAV123



CVPR 2017 Trackers		OTB-100 AUC (%)	Speed (FPS)	
ECO	(Ours)	**70.0**	15	(GPU)
ECO-HC	(Ours)	65.0	60	(CPU)
ACFN	(J. Choi et al.)	57.5	15	(GPU)
ADNet	(S. Yun et al.)	64.6	3	(GPU)
CSR-DCF	(A. Lukežič et al.)	58.7	13	(CPU)
CFNet	(J. Valmadre et al.)	58.6	43	(GPU)
LMCF	(M. Wang et al.)	56.8	80	(CPU)
MCPF	(T. Zhang et al.)	62.8	0.54	(GPU)
Obli-RaF	(L. Zhang et al.)	56.5	2	(GPU)
SANet	(H. Fan, H. Ling)	69.2	1	(GPU)
Staple_CA	(M. Mueller et al.)	59.8	35	(CPU)

Best result in CVPR 2017!

References

[1] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In ECCV, 2016.

[2] A. Declercq and J. H. Piater. Online learning of Gaussian mixture models - a two-level approach. In VISAPP, 2008.