

Discriminative Correlation Filter Tracker with Channel and Spatial Reliability

Alan Lukežič¹, Tomáš Vojtáš², Luka Čehovin Zajc¹, Jiří Matas² and Matej Kristan¹

¹Faculty of Computer and Information Science, University of Ljubljana, Slovenia

²Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

{alan.lukezic, luka.cehovin, matej.kristan}@fri.uni-lj.si

{vojirtom, matas}@cmp.felk.cvut.cz

Abstract

Short-term tracking is an open and challenging problem for which discriminative correlation filters (DCF) have shown excellent performance. We introduce the channel and spatial reliability concepts to DCF tracking and provide a learning algorithm for its efficient and seamless integration in the filter update and the tracking process. The spatial reliability map adjusts the filter support to the part of the object suitable for tracking. This both allows to enlarge the search region and improves tracking of non-rectangular objects. Reliability scores reflect channel-wise quality of the learned filters and are used as feature weighting coefficients in localization. Experimentally, with only two simple standard feature sets, HoGs and Colormnames, the novel CSR-DCF method – DCF with Channel and Spatial Reliability – achieves state-of-the-art results on VOT 2016, VOT 2015 and OTB100. The CSR-DCF runs close to real-time on a CPU.

Keywords— Visual tracking, Correlation filters, Channel reliability, Constrained optimization

1 Introduction

Short-term, model-free visual object tracking is the problem of continuously localizing a target in a video-sequence given a single example of its appearance. It has received significant attention of the computer vision community which is reflected in the number of papers published on the topic and the existence of multiple performance evaluation benchmarks (Wu et al, 2013; Kristan

et al, 2013, 2014, 2015, 2016c; Liang et al, 2015; Smeulders et al, 2014; Mueller et al, 2016). Diverse factors – occlusion, illumination change, fast object or camera motion, appearance changes due to rigid or non-rigid deformations and similarity to the background – make short-term tracking challenging.

Recent short-term tracking evaluations (Wu et al, 2013; Kristan et al, 2013, 2014, 2015) consistently confirm the advantages of semi-supervised discriminative tracking approaches (Grabner et al, 2006; Babenko et al, 2011; Hare et al, 2011; Bolme et al, 2010). In particular, trackers based on the discriminative correlation filter (DCF) method (Bolme et al, 2010; Danelljan et al, 2014a; Henriques et al, 2015; Li and Zhu, 2014a; Danelljan et al, 2015a) have shown state-of-the-art performance in all standard benchmarks. Discriminative correlation methods learn a filter with a pre-defined response on the training image. The latter is obtained by slightly extending the region around the target to include background samples.

The standard formulation of DCF uses circular correlation which allows to implement learning efficiently by Fast Fourier transform (FFT). However, the FFT requires the filter and the search region size to be equal which limits the detection range. Due to the circularity, the filter is trained on many examples that contain unrealistic, wrapped-around circularly-shifted versions of the target. A naive approach to the reduction of the windowing problems is to learn the filter from a larger region. However, due to the large area of the background in the region, the tracking performance of the DCF drops significantly as shown in Figure 2.

The windowing problems were recently addressed by

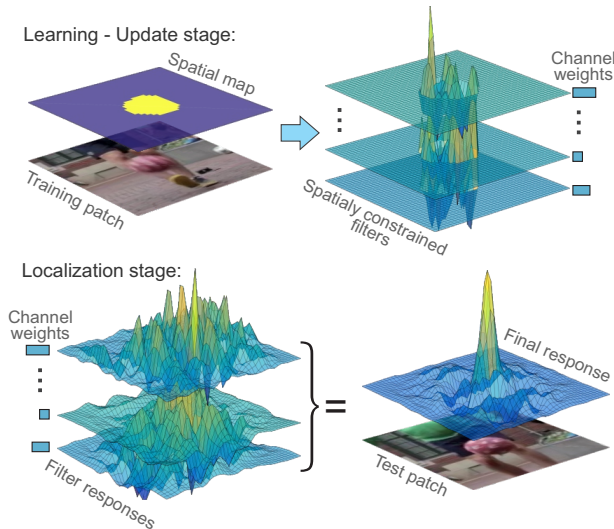


Figure 1: Overview of the CSR-DCF approach. An automatically estimated spatial reliability map restricts the correlation filter to the parts suitable for tracking (top) improving localization within a larger search region and performance for irregularly shaped objects. Channel reliability weights calculated in the constrained optimization step of the correlation filter learning reduce the noise of the weight-averaged filter response (bottom).

Kiani Galoogahi et al (2015) who propose zero-padding the filter during learning and by Danelljan et al (2015a) who introduce spatial regularization to penalize filter values outside the target boundaries. Both approaches train from image regions much larger than the target and thus increase the detection range.

Another limitation of the published DCF methods is the assumption that the target shape is well approximated by an axis-aligned rectangle. For irregularly shaped objects or those with a hollow center, the filter eventually learns the background, which may lead to drift and failure. The same problem appears for approximately rectangular objects in the case of occlusion. The Kiani Galoogahi et al (2015) and Danelljan et al (2015a) methods both suffer from this problem.

In this paper we introduce the CSR-DCF, the Discriminative Correlation Filter with Channel and Spatial Reliability. The spatial reliability map adapts the filter support to the part of the object suitable for tracking which

overcomes both the problems of circular shift enabling an arbitrary search (and training) region size and the limitations related to the rectangular shape assumption. An important benefit of a large training region is that background samples from a wider area around the target are obtained to improve the filter discriminative power. The spatial reliability map is estimated using the output of a graph labeling problem solved efficiently in each frame. An efficient optimization procedure is applied for learning a correlation filter with the support constrained by the spatial reliability map since the standard closed-form solution cannot be generalized to this case. Figure 2 shows that tracking performance of our spatially constrained correlation filter (denoted as S-DCF) does not degrade with increasing training and search region size as is the case with the standard DCF. In contrast, the performance of S-DCF improves from better treatment of training samples and increased search region size. Experiments show that the novel filter optimization procedure outperforms related approaches for constrained learning in DCFs.

Channel reliability is the second novelty the CSR-DCF tracker introduces. The reliability is estimated from the properties of the constrained least-squares solution to filter design. The channel reliability scores are used for weighting the per-channel filter responses in localization (Figure 1). The CSR-DCF shows state-of-the-art performance on standard benchmarks – OTB100 (Wu et al, 2015), VOT2015 (Kristan et al, 2015) and VOT2016 (Kristan et al, 2015) while running close to real-time on a single CPU. The spatial and channel reliability formulation is general and can be used in most modern correlation filters, e.g. those using deep features.

The remainder of the paper is structured as follows. In Section 2 we review most closely related work, our approach is described in Section 3, experimental results are presented in Section 4 and conclusions are drawn in Section 5.

2 Related work

The discriminative correlation filters for object detection date back to the 80’s with seminal work of Hester and Casasent (1980). They have been popularized only recently in the tracking community, starting with the Bolme et al (2010) MOSSE tracker. Using a gray-scale tem-

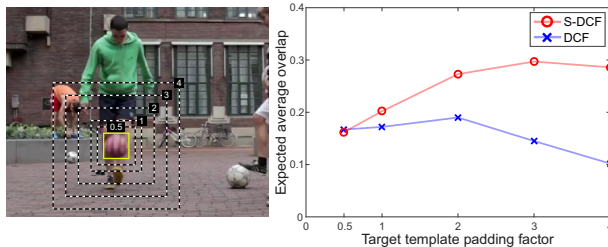


Figure 2: Tracking performance measured by the Expected Average Overlap (EAO) of the standard DCF and our spatially constrained DCF (S-DCF) as a function of search region size, expressed as the multiple of the target size (right, x-axis). The filter is learned from a training region equal in size to the search region. The search region sizes are visualized by black-white dashed rectangles (left image) and the target bounding box is shown in yellow.

plate, MOSSE achieved state-of-the-art performance on a tracking benchmark (Wu et al, 2013) at a remarkable processing speed. Significant improvements have been made since and in 2014 the top-performing trackers on a recent benchmark (Kristan et al, 2014) were all from this class of trackers. DCF improvements fall into two categories, introduction of new features and conceptual improvements in filter learning.

In the first group, Henriques et al (2015) replaced the grayscale templates by HoG (Dalal and Triggs, 2005), Danelljan et al (2014b) proposed multi-dimensional color attributes and Li and Zhu (2014b) applied feature combination. Recently, convolutional network features learned for object detection have been applied (Ma et al, 2015; Danelljan et al, 2015b, 2016), leading to a performance boost, but at a cost of significant speed reduction.

Conceptually, the first successful theoretical extension of the standard DCF was the kernelized formulation by Henriques et al (2015) which achieved remarkable tracking performance, but still preserved high speed. Later, a correlation filter based scale adaptation was proposed by Danelljan et al (2014a) introduced a scale-space pyramid learned within a correlation filter framework. Zhang et al (2014) introduced spatio-temporal context learning in the DCFs. To improve localization with correlation filters, Bertinetto et al (2016a) proposed a tracking method that combines the output of the correlation filter with the tar-

get segmentation probability map. Danelljan et al (2016) addressed a multiple-resolution feature map issue in correlation filters by formulating filter learning in continuous space, while Qi et al (2016) proposed a mechanism to combine correlation responses from multiple convolutional layers. A correlation filter tracker which is able to handle drifts in longer sequences was proposed by Wang et al (2016). It clusters similar target appearances together and uses the clusters for target localization instead of a single online learned filter.

Since most of the correlation filter trackers represent the target with a single filter, it can easily get corrupted when occlusion or a target deformation happen. In general, part-based trackers are better in addressing these issues. Therefore several part-based correlation filter methods were proposed. Liu et al (2015) use an efficient method to combine correlation outputs of multiple parts and Liu et al (2016) proposed a tracking method for modeling the target structure with multiple parts using multiple correlation filters. Lukežič et al (2017) treat the parts correlation filter responses and their constellation constraints jointly as an equivalent spring system. They derive a highly efficient optimization to infer the most probable target deformation.

Recently, Kiani Galoogahi et al (2015) addressed the problem that occurs due to learning with circular correlation from small training regions. They proposed a learning framework that artificially increases the filter size by implicitly zero padding the filter. This reduces the boundary artifacts by increasing the number of training examples in constrained filter learning. Danelljan et al (2015a) reformulate the learning cost function to penalize non-zero filter values outside the object bounding box. Performance better than (Kiani Galoogahi et al, 2015) is reported, but the learned filter is still a trade-off between the correlation response and regularization, and it does not guarantee that filter values are zero outside of object bounding box.

3 Spatially constrained correlation filters

The use of multiple channels in correlation filters (Henriques et al, 2015; Danelljan et al, 2017; Galoogahi et al,

2013) has become very popular in visual tracking. In the following we present the main ideas behind learning these filters. Given a set of N_c channel features $\mathbf{f} = \{\mathbf{f}_d\}_{d=1:N_c}$ and corresponding target templates (filters) $\mathbf{h} = \{\mathbf{h}_d\}_{d=1:N_c}$, the object position is estimated as the location of the maximum of correlation response $\tilde{\mathbf{g}}(\mathbf{h})$,

$$\tilde{\mathbf{g}}(\mathbf{h}) = \sum_{d=1}^{N_c} \mathbf{f}_d \star \mathbf{h}_d. \quad (1)$$

The symbol \star represents circular correlation between $\mathbf{f}_d \in \mathcal{R}^{c_w \times c_h}$ and $\mathbf{h}_d \in \mathcal{R}^{c_w \times c_h}$, where c_w and c_h are the training/search region width and height, respectively. The optimal correlation filter \mathbf{h} is estimated by minimizing

$$\varepsilon(\mathbf{h}) = \|\tilde{\mathbf{g}}(\mathbf{h}) - \mathbf{g}\|^2 + \lambda \|\mathbf{h}\|^2, \quad (2)$$

where \mathbf{g} is the desired output $\mathbf{g} \in \mathcal{R}^{c_w \times c_h}$, which is typically a 2-D Gaussian function centered at the target location. Efficient tracking performance is achieved by expressing the cost (2) into the Fourier domain

$$\varepsilon(\mathbf{h}) = \left\| \sum_{d=1}^{N_c} \text{diag}(\hat{\mathbf{f}}_d) \bar{\mathbf{h}}_d - \hat{\mathbf{g}} \right\|^2 + \lambda \sum_{d=1}^{N_c} \|\hat{\mathbf{h}}_d\|^2, \quad (3)$$

where the operator $\hat{\mathbf{a}} = \text{vec}(\mathcal{F}[\mathbf{a}])$ is a Fourier transform of \mathbf{a} reshaped into a column vector, i.e., $\hat{\mathbf{a}} \in \mathcal{R}^{D \times 1}$, with $D = c_w \cdot c_h$, $\text{diag}(\hat{\mathbf{a}})$ being a $D \times D$ diagonal matrix formed from $\hat{\mathbf{a}}$ and (\cdot) is the complex-conjugate operator. The closed-form solution for d -th filter channel $\hat{\mathbf{h}}_d$ which minimizes the cost function (3) is equal to

$$\hat{\mathbf{h}}_d = \left(\text{diag}(\hat{\mathbf{f}}_d) \bar{\hat{\mathbf{g}}} \right) \odot^{-1} \left(\sum_{d=1}^{N_c} \text{diag}(\hat{\mathbf{f}}_d) \bar{\hat{\mathbf{f}}}_d + \lambda \right), \quad (4)$$

where \odot^{-1} is element-wise division. The solution (4) considers all feature channels jointly and is used in most of the recent correlation filter trackers. Note that the final response is obtained as summation over correlation responses of all channels (1) and the location of the maximum in the final response represents the new position of the target.

Note that a filter for the d -th channel is computed in (4) by dividing d -th feature with the sum over all feature channels. This means that the feature *scale* crucially impacts the level by which a channel contributes to the final

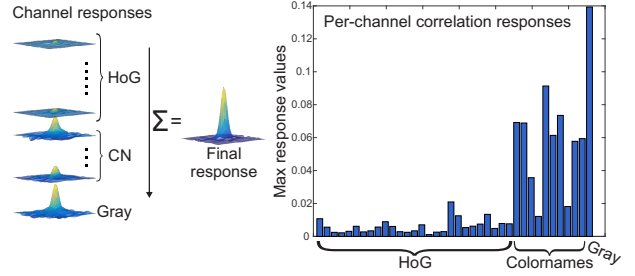


Figure 3: Correlation responses of different feature channels (27 HoG, 10 colormnames and one grayscale channel) are summed to obtain the final (single channel) correlation response (middle). Note that maximum values of channel responses may vary by orders of magnitude (right).

response, irrespective of its discriminative power. Since features (e.g., HoG, colormnames and grayscale template) vary in scale, some channels might suppress the others by an order of magnitude. This is demonstrated in Figure 3 where each HoG channel on its own contributes to the final response very little.

To avoid the issue with different scales we consider each channel independently. This means that each filter channel is optimized to fit the desired output separately. The cost function is thus defined as

$$\varepsilon(\mathbf{h}) = \sum_{d=1}^{N_c} \|\mathbf{f}_d \star \mathbf{h}_d - \mathbf{g}\|^2 + \lambda \|\mathbf{h}_d\|^2. \quad (5)$$

Additionally, we introduce channel weights $\mathbf{w} = \{\tilde{w}_d\}_{d=1:N_c}$ which can be considered as scaling factors based on the discriminative power of each feature channel. These weights are called *channel reliability weights* in the rest of the paper and they are applied when correlation response is calculated in the target localization stage:

$$\tilde{\mathbf{g}} = \sum_{d=1}^{N_c} \mathbf{f}_d \star \mathbf{h}_d \cdot \tilde{w}_d. \quad (6)$$

We present our method for constrained correlation filter learning in Section 3.1. The most reliable parts of the filter are identified by introducing the spatial reliability map (Section 3.2). The method for channel reliability w_d estimation is presented in Section 3.3, the proposed tracker is described in Section 3.4.

3.1 Constrained correlation filter learning

Since filter learning is independent across the channels in our formulation (5), we assume only a single channel in the following derivation (i.e., $N_c = 1$) and drop the channel index for clarity.

Let $\mathbf{m} \in \{0, 1\}$ be a spatial reliability map with elements either zero or one, that identifies pixels which should be set to zero in the learned filter. The constraint can be formalized as $\mathbf{h} \equiv \mathbf{m} \odot \mathbf{h}$, where \odot represents the Hadamard (element-wise) product. Such constraint does not lead to a closed-form solution, but an iterative approach akin to Kiani Galoogahi et al (2015) can be derived for efficiently solving the optimization problem. In the following we summarize the main steps of our approach and report the full derivation in Appendix 6.

We start by introducing a dual variable \mathbf{h}_c and the constraint

$$\mathbf{h}_c - \mathbf{m} \odot \mathbf{h} \equiv 0, \quad (7)$$

which leads to the following augmented Lagrangian (Boyd et al, 2011)

$$\mathcal{L}(\hat{\mathbf{h}}_c, \mathbf{h}, \hat{\mathbf{l}}|\mathbf{m}) = \|\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{g}}\|^2 + \frac{\lambda}{2}\|\mathbf{h}_m\|^2 + \quad (8)$$

$$[\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \hat{\mathbf{h}}_m) + \overline{\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \hat{\mathbf{h}}_m)}] + \mu\|\hat{\mathbf{h}}_c - \hat{\mathbf{h}}_m\|^2,$$

where $\hat{\mathbf{l}}$ is a complex Lagrange multiplier, $\mu > 0$, and we use the definition $\mathbf{h}_m = (\mathbf{m} \odot \mathbf{h})$ for compact notation. The augmented Lagrangian (8) can be iteratively minimized by the alternating direction method of multipliers, e.g. Boyd et al (2011), which sequentially solves the following sub-problems at each iteration:

$$\hat{\mathbf{h}}_c^{i+1} = \arg \min_{\mathbf{h}_c} \mathcal{L}(\hat{\mathbf{h}}_c, \mathbf{h}^i, \hat{\mathbf{l}}^i|\mathbf{m}), \quad (9)$$

$$\mathbf{h}^{i+1} = \arg \min_{\mathbf{h}} \mathcal{L}(\hat{\mathbf{h}}_c^{i+1}, \mathbf{h}, \hat{\mathbf{l}}^i|\mathbf{m}), \quad (10)$$

and the Lagrange multiplier is updated as

$$\hat{\mathbf{l}}^{i+1} = \hat{\mathbf{l}}^i + \mu(\hat{\mathbf{h}}_c^{i+1} - \hat{\mathbf{h}}^{i+1}). \quad (11)$$

Minimizations in (9,10) have at each iteration a closed-form solution, i.e.,

$$\hat{\mathbf{h}}_c^{i+1} = (\hat{\mathbf{f}} \odot \hat{\mathbf{g}} + (\mu\hat{\mathbf{h}}_m^i - \hat{\mathbf{l}}^i)) \odot^{-1} (\hat{\mathbf{f}} \odot \hat{\mathbf{f}} + \mu^i), \quad (12)$$

$$\mathbf{h}^{i+1} = \mathbf{m} \odot \mathcal{F}^{-1}[\hat{\mathbf{l}}^i + \mu^i\hat{\mathbf{h}}_c^{i+1}]/(\frac{\lambda}{2D} + \mu^i). \quad (13)$$

A standard scheme for updating the constraint penalty μ values (Boyd et al, 2011) is applied, i.e., $\mu^{i+1} = \beta\mu^i$.

Computations of (12,11) are fully carried out in the frequency domain, the solution for (13) requires a single inverse FFT and another FFT to compute the $\hat{\mathbf{h}}^{i+1}$. A single optimization iteration thus requires only two calls of the Fourier transform, resulting in a very fast optimization. The computational complexity is that of the Fourier transform, i.e., $\mathcal{O}(D \log D)$. Filter learning is implemented in less than five lines of Matlab code and is summarized in the Algorithm 1.

Algorithm 1 : Constrained filter optimization.

Require:

Features extracted from training region \mathbf{f} , ideal correlation response \mathbf{g} ,
binary mask \mathbf{m} .

Ensure:

Optimized filter $\hat{\mathbf{h}}$.

Procedure:

- 1: Initialize filter $\hat{\mathbf{h}}^0$ by \mathbf{h}_{t-1} .
 - 2: Initialize Lagrangian coefficients: $\hat{\mathbf{l}}^0 \leftarrow$ zeros.
 - 3: **while** stop condition **do**
 - 4: Calculate $\hat{\mathbf{h}}_c^{i+1}$ from $\hat{\mathbf{h}}^i$ and $\hat{\mathbf{l}}^i$ using (12).
 - 5: Calculate \mathbf{h}^{i+1} from $\hat{\mathbf{h}}_c^{i+1}$ and $\hat{\mathbf{l}}^i$ using (13).
 - 6: Update the Lagrangian $\hat{\mathbf{l}}^{i+1}$ from $\hat{\mathbf{h}}_c^{i+1}$ and \mathbf{h}^{i+1} (11).
 - 7: **end while**
-

3.2 Constructing spatial reliability map

Once the target is localized, a training region is extracted and used to update the filter. Our constrained filter learning (13) requires estimation of spatial reliability map \mathbf{m} (i.e., segmentation) that identifies pixels in the training region which likely belong to the target (see Figure 4). In the following we briefly outline the segmentation model which is used to estimate \mathbf{m} .

During tracking, the object foreground/background color models are maintained as color histograms $\mathbf{c} = \{\mathbf{c}^f, \mathbf{c}^b\}$. Let $\mathbf{y}_i = [\mathbf{y}_i^c, \mathbf{y}_i^x]$ be the observation, i.e., the color \mathbf{y}_i^c and position \mathbf{y}_i^x at i -th pixel in the training region and let $m_i \in \{0, 1\}$ be a random variable denoting the unknown foreground/background label. The joint

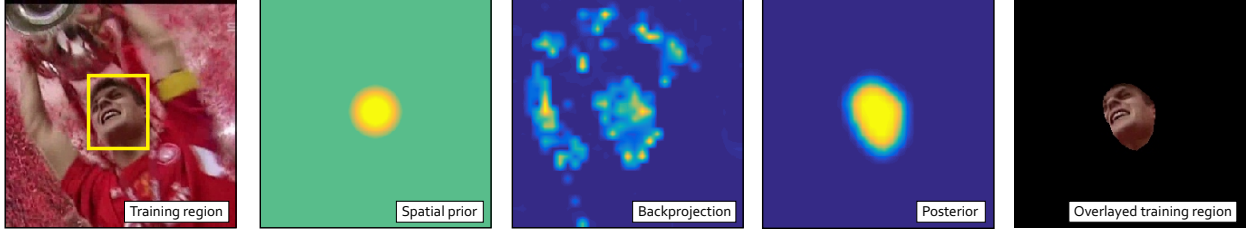


Figure 4: Spatial reliability map construction from the training region. From left to right: a training region with the target bounding box, the foreground-background color models, the posterior object probability after Markov random field regularization, and the training region masked with the final binary reliability map. The probabilities are color-coded in a blue (0.0) – green (0.5) – yellow (1.0) colormap.

probability of observing y_i is defined as

$$\begin{aligned} p(\mathbf{y}_i) &= \sum_{j=0}^1 p(\mathbf{y}_i | m_i = j) p(m_i = j) = \\ &= \sum_{j=0}^1 p(\mathbf{y}_i^c | m_i = j) p(\mathbf{y}_i^x | m_i = j) p(m_i = j), \end{aligned} \quad (14)$$

where $p(\mathbf{y}_i^c | m_i = j)$, $p(\mathbf{y}_i^x | m_i = j)$ and $p(m_i = j)$ are the appearance likelihood, the spatial likelihood and the foreground/background prior probability. The appearance likelihood term $p(\mathbf{y}_i^c | m_i = j)$ is computed by Bayes rule from the object foreground/background color models \mathbf{c}^f and \mathbf{c}^b . The prior probability $p(m_i = j)$ is defined by the ratio between the region sizes for foreground/background histogram extraction.

The central pixels in axis-aligned approximations of an elongated rotating, articulated or deformable object are likely to contain the object regardless of the specific deformation. On the other hand, in the absence of measurements, pixels away from the center belong to the object or background equally likely. This deformation invariance of central elements reliability is enforced in our approach by defining a weak spatial prior

$$p(\mathbf{y}_i^x | m_i = j) = k(\mathbf{x}; \sigma), \quad (15)$$

where $k(\mathbf{x}; \sigma)$ is a modified Epanechnikov kernel, $k(r; \sigma) = 1 - (r/\sigma)^2$, with size parameter σ equal to the minor bounding box axis and clipped to interval $[0.5, 0.9]$ such that the object prior probability at center is 0.9 and changes to a uniform prior away from the center (Figure 4).

3.2.1 Inference

In practice the likelihood $p(\mathbf{y}_i | m_i)$ is noisy and requires regularization for our filter learning. We thus apply a MRF from (Diplaros et al, 2007; Kristan et al, 2016a), which treats the prior and posterior label distributions over pixels as random variables and applies a MRF constraint over these. This formulation affords an efficient inference which avoids hard label assignment during optimization and can be implemented as a series of convolutions.

The prior over the i -th pixel is defined compactly as $\pi_i = [\pi_{i0}, \pi_{i1}]$ with $\pi_{ij} = p(m_i = j)$ and a standard approximation is made (Diplaros et al, 2007) that decomposes the joint pdf over priors $\pi = [\pi_1, \dots, \pi_M]$ into a product of local conditional distributions $p(\pi) = \prod_{i=1}^M p(\pi_i | \pi_{N_i})$, where M is number of pixels, π_{N_i} is a mixture distribution over the priors of i -th pixel's neighbors, i.e., $\pi_{N_i} = \sum_{j \in N_i, j \neq i} \lambda_{ij} \pi_j$ and λ_{ij} are fixed weights satisfying $\sum_j \lambda_{ij} = 1$. In Diplaros et al (2007) the weights are fixed to a normalized Gaussian and are shared across all pixel locations. The potentials in the MRF are defined as $p(\pi_i | \pi_{N_i}) \propto \exp(-\frac{1}{2} E(\pi_i, \pi_{N_i}))$, with exponent defined as $E(\pi_i, \pi_{N_i}) = D(\pi_i || \pi_{N_i}) + H(\pi_i)$. The term $D(\pi_i || \pi_{N_i})$ is the Kullback-Leibler divergence which penalizes the difference between prior distributions over the neighboring pixels (π_i and π_{N_i}), while the term $H(\pi_i)$ is the entropy defined as $H(\pi_i) = -\sum_{j=0}^1 \pi_{ij} \log \pi_{ij}$, which penalizes uninformative priors π_i .

For smooth solutions Diplaros et al (2007) propose us-

ing a similar constraint over the posteriors $p_i = [p_{i0}, p_{i1}]$ with p_{ij} being the posterior probability of class j at i -th pixel, leading to the following energy function

$$F = \sum_{i=1}^M \left[\log p(y_i) - \frac{1}{2} \left(E(\pi_i, \pi_{N_i}) + E(p_i, p_{N_i}) \right) \right]. \quad (16)$$

Minimization of the energy (16) w.r.t. π and p is efficiently solved by the solver from Diplaros et al (2007). The final mask \mathbf{m} for learning the filter in Section 3.1 is obtained by thresholding the posterior at 0.5.

3.3 Channel reliability estimation

Channel reliability \tilde{w}_d in (6) reflects the importance of each channel at the target localization stage. In our approach it consists of two types of reliability measures: (i) *channel learning reliability* $\tilde{w}_d^{(\text{lrn})}$, which is calculated in the filter learning stage, and (ii) *channel detection reliability* $\tilde{w}_d^{(\text{det})}$ which is calculated in the target localization stage. The joint channel reliability \tilde{w}_d in (6) at target localization stage is computed as the product of both reliability measures, i.e.,

$$\tilde{w}_d = \tilde{w}_d^{(\text{lrn})} \cdot \tilde{w}_d^{(\text{det})} \quad (17)$$

and normalized s.t. $\sum_d \tilde{w}_d = 1$. The reliability measures are described in following paragraphs.

Channel learning reliability. Constrained minimization of (8) solves a least squares problem averaged over all circular displacements of the filter on a feature channel. A discriminative feature channel \mathbf{f}_d produces a filter \mathbf{h}_d whose output $\mathbf{f}_d * \mathbf{h}_d$ nearly exactly fits the ideal response \mathbf{g} . On the other hand, since the response is highly noisy on channels with low discriminative power, a global error reduction in the least squares significantly reduces the maximal response. This effect is demonstrated in Figure 5, which shows correlation responses for a highly discriminative and non-discriminative channels. Thus a straightforward measure of channel learning reliability $\tilde{w}_d^{(\text{lrn})}$ is the maximum response value of a learned channel filter, which is computed as

$$\tilde{w}_d^{(\text{lrn})} = \max(\mathbf{f}_d * \mathbf{h}_d). \quad (18)$$

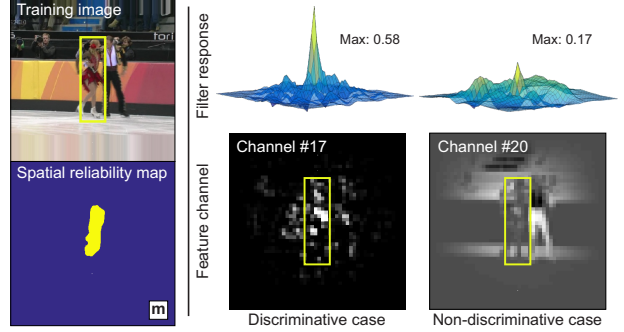


Figure 5: A filter is learned on feature channels from a training region using the constrained optimization with a binary segmentation mask \mathbf{m} . Correlation responses between the learned filter and the training region for two feature channels are shown on the right. On a discriminative feature channel the filter response is much stronger and less noisy than on a non-discriminative channel.

Channel detection reliability. The second part of the channel reliability reflects how uniquely each channel votes for a single target location. Note that Bolme et al (2010) proposed a similar approach to detect loss of target. Our measure is based on the ratio between the second and first highest non-adjacent peaks in the channel response map, i.e., $1 - \rho_d^{\max 2} / \rho_d^{\max 1}$. The two largest peaks in the response map are obtained as two largest values after a 3×3 non-maximum suppression. Note that this ratio penalizes situations in which multiple similar objects appear in the target vicinity (i.e., response map contains many well expressed modes), even if the major mode accurately depicts the target position. To mitigate such penalizations, the final values are not allowed to fall below 0.5. The detection reliability of d -th channel is estimated as

$$\tilde{w}_d^{(\text{det})} = \max(1 - \rho_d^{\max 2} / \rho_d^{\max 1}, 0.5). \quad (19)$$

3.4 Tracking with channel and spatial reliability

A single tracking iteration of the proposed channel and spatial reliability correlation filter tracker (CSR-DCF) is summarized in Algorithm 2 and visualized in Figure 6. The localization and update steps proceed as follows.

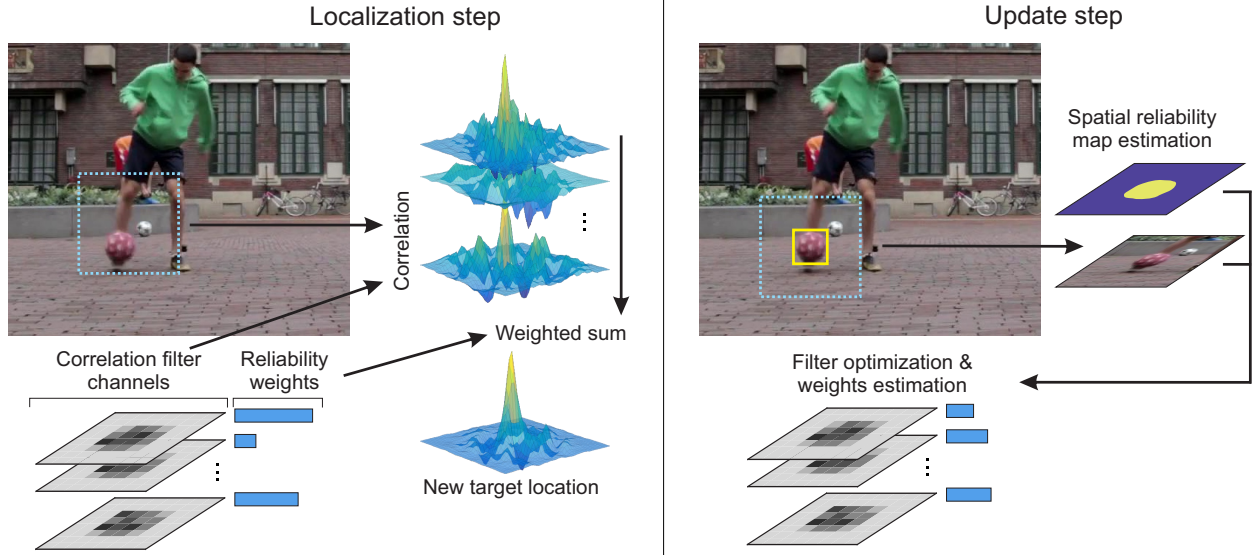


Figure 6: The CSR-DCF tracking iteration: localization step is shown on the left and update step on the right side of the image.

Localization step. Features are extracted from a search region centered at the target estimated position in the previous time-step and correlated with the learned filter \mathbf{h}_{t-1} . The object is localized by summing the correlation responses weighted by the estimated channel reliability scores \mathbf{w}_{t-1} . The scale is estimated by a single scale-space correlation filter as in Danelljan et al (2014a). Per-channel filter responses are used to compute the corresponding detection reliability values $\tilde{\mathbf{w}}^{(\text{det})} = [\tilde{w}_1^{(\text{det})}, \dots, \tilde{w}_{N_c}^{(\text{det})}]^T$ according to (19).

Update step. The training region is centered at the target location estimated at localization step. The foreground and background histograms $\tilde{\mathbf{c}}$ are extracted and updated by exponential moving average with learning rate η_c (step 5 in Algorithm 2). The foreground histogram is extracted by an Epanechnikov kernel within the estimated object bounding box and the background is extracted from the neighborhood twice the object size. The spatial reliability map \mathbf{m} (Sect. 3.2) is constructed and the optimal filters \mathbf{h} are computed by optimizing (8). The per-channel learning reliability weights $\tilde{\mathbf{w}}^{(\text{lrn})} = [\tilde{w}_1^{(\text{lrn})}, \dots, \tilde{w}_{N_c}^{(\text{lrn})}]^T$ are estimated from the correlation responses (18). Current frame reliability weights $\tilde{\mathbf{w}}$ are computed from detection and

learning reliability (17). The filters and channel reliability weights are updated by exponential moving average (current and from previous frame) with learning rate η (steps 10 and 11 in the Algorithm 2). Note that we compute the spatial reliability map in each frame independently to capture large target appearance changes, e.g. caused by rotation or deformation.

3.5 Comparison with prior work

Kiani Galoogahi et al (2015) and Danelljan et al (2015a) have previously considered constrained filter learning. Here we highlight the differences of our approach.

The LBCF tracker (Kiani Galoogahi et al, 2015) addresses the circular boundary effect of the Fourier transform and implicitly increases the filter search region size. In contrast, the CSR-DCF primarily reduces the impact of the background in the filter. The solution of Kiani Galoogahi et al (2015) is similar to our filter optimization, but it is derived for a rectangular mask only. Since rotating and deformable targets are poorly approximated by an axis-aligned bounding box their filter is contaminated by background leading to a reduced performance. The LBCF updates the auto-spectral and cross-spectral ener-

Algorithm 2 : The CSR-DCF tracking algorithm.

Require:

Image \mathbf{I}_t , object position on previous frame \mathbf{p}_{t-1} , scale s_{t-1} , filter \mathbf{h}_{t-1} , color histograms \mathbf{c}_{t-1} , channel reliability \mathbf{w}_{t-1} .

Ensure:

Position \mathbf{p}_t , scale s_t and updated models.

Localization and scale estimation:

- 1: New target location \mathbf{p}_t : position of the maximum in correlation between \mathbf{h}_{t-1} and image patch features \mathbf{f} extracted on position \mathbf{p}_{t-1} and weighted by the channel reliability scores \mathbf{w} (Sect. 3.3).
- 2: Using per-channel responses, estimate detection reliability $\tilde{\mathbf{w}}^{(\text{det})}$ (Sect. 3.3).
- 3: Using location \mathbf{p}_t , estimate new scale s_t .

Update:

- 4: Extract foreground and background histograms $\tilde{\mathbf{c}}^f, \tilde{\mathbf{c}}^b$.
 - 5: Update foreground and background histograms $\mathbf{c}_t^f = (1 - \eta_c)\mathbf{c}_{t-1}^f + \eta_c\tilde{\mathbf{c}}^f, \mathbf{c}_t^b = (1 - \eta_c)\mathbf{c}_{t-1}^b + \eta_c\tilde{\mathbf{c}}^b$.
 - 6: Estimate reliability map \mathbf{m} (Sect. 3.2).
 - 7: Estimate a new filter $\tilde{\mathbf{h}}$ using \mathbf{m} (Algorithm 1).
 - 8: Estimate learning channel reliability $\tilde{\mathbf{w}}^{(\text{lrn})}$ from \mathbf{h} (Sect. 3.3).
 - 9: Calculate channel reliability $\tilde{\mathbf{w}} = \tilde{\mathbf{w}}^{(\text{lrn})} \odot \tilde{\mathbf{w}}^{(\text{det})}$
 - 10: Update filter $\mathbf{h}_t = (1 - \eta)\mathbf{h}_{t-1} + \eta\tilde{\mathbf{h}}$.
 - 11: Update channel reliability $\mathbf{w}_t = (1 - \eta)\mathbf{w}_{t-1} + \eta\tilde{\mathbf{w}}$.
-

gies ($\hat{\mathbf{f}} \odot \tilde{\mathbf{f}}$ and $\hat{\mathbf{f}} \odot \tilde{\mathbf{g}}$ in (12)) separately, which approximates computation of a single filter from a weighted sum of errors over past training samples. This adaptation is reasonable since it is derived for a rectangular mask that remains constant throughout tracking. The CSR-DCF estimates the mask separately for each training sample and learns a corresponding filter. For articulated objects in particular the mask varies significantly with time, therefore it is beneficial to compute the exact filter for each frame. Robustness is increased by moderately averaging the filters temporally.

Similarly to our approach, the SRDCF (Danelljan et al, 2015a) uses a spatial map in filter learning. In contrast to our approach, their map does not adapt to the target and is required to be highly smooth for their optimization

to converge. In CSR-DCF the map serves as a hard constraint resulting in a filter with values off the target set to zero. In contrast, the SRDCF (Danelljan et al, 2015a) filter is a compromise between target position regression and a penalty term that *prefers* potentially non-zero values in the filter center and close-to-zero values away from the center, but does not guarantee zero values outside the mask.

4 Experimental analysis

This section presents a comprehensive experimental evaluation of the CSR-DCF tracker. Implementation details are discussed in Section 4.1, convergence of the filter optimization method is presented in Section 4.2, Section 4.3 reports comparison of the proposed constrained learning to the related state-of-the-art and the ablation study is provided in Section 4.4. Tracking performance on three recent benchmarks: OTB-100 (Wu et al, 2015), VOT2015 (Kristan et al, 2015) and VOT2016 (Kristan et al, 2016b) is reported in Sections 4.6, 4.7 and 4.8, respectively. The detailed analysis of the tracker, including per-attribute tracking performance is presented in Section 4.9 and tracking speed analysis in Section 4.10.

4.1 Implementation details and parameters

A popular implementation Felzenszwalb et al (2010) of the standard HoG (Dalal and Triggs, 2005) and Colornames (van de Weijer et al, 2009) features are used in the correlation filter and HSV foreground/background color histograms with 16 bins per color channel are used in reliability map estimation with parameter $\alpha_{\min} = 0.05$. All the parameters are set to values commonly used in literature (Danelljan et al, 2015a; Kiani Galoogahi et al, 2015). Histogram adaptation rate is set to $\eta_c = 0.04$, correlation filter adaptation rate is set to $\eta = 0.02$, and the regularization parameter is set to $\lambda = 0.01$. The augmented Lagrangian optimization parameters are set to $\mu^0 = 5$ and $\beta = 3$. All parameters have a straight-forward interpretation, do not require fine-tuning, and were kept constant throughout all experiments. Our Matlab implementation¹

¹The CSR-DCF Matlab source is publicly available on: <https://github.com/alanlukezic/csr-dcf>

runs at 13 frames per second on an Intel Core i7 3.4GHz standard desktop.

4.2 Convergence of constrained learning

The constrained filter learning described in Section 3.1 is an iterative optimization method that minimizes the cost function (8). This experiment demonstrates how the cost changes with the number of iterations during filter optimization.

Figure 7 shows the average squared difference between the result of the correlation of the filter constrained by the spatially constrained function and the ideal output. This graph was obtained by averaging 60 examples of initializing a filter on a target (one per VOT2015 sequence) and scaling each to an interval between zero and one. It is clear that the error drops by 80% within the first few iterations. Already after four iterations, the performance improvements become negligible, therefore we set number of iterations to $N = 4$.

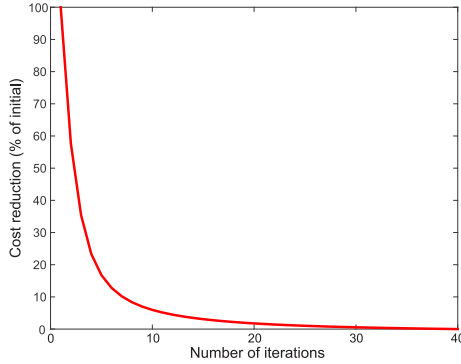


Figure 7: Convergence speed of constrained filter learning from Section 3.1 shown as a relative drop of the initial cost.

4.3 Impact of the boundary constraint formulation

This section compares our proposed boundary constraints formulation (Sect. 3) with recent state-of-the-art approaches (Danelljan et al, 2015a; Kiani Galoogahi et al, 2015). In the first experiment, three variants of the standard single-scale HoG-based correlation filter were im-

plemented to emphasize the difference in boundary constraints: the first uses our spatial reliability boundary constraint formulation from Section 3 (T_{SC}) the second applies the spatial regularization constraint (Danelljan et al, 2015a) (T_{SR}) and the third applies the limited boundaries constraint (Kiani Galoogahi et al, 2015) (T_{LB}).

The three variants were compared on the challenging VOT2015 dataset (Kristan et al, 2015) by applying a standard no-reset one-pass evaluation from OTB (Wu et al, 2013) and computing the AUC on the success plot. The tracker with our constraint formulation T_{SC} achieved 0.32 AUC, while the alternatives achieved 0.28 (T_{SR}) and 0.16 (T_{LB}). The only difference between these trackers is in the constraint formulation, which indicates superiority of the proposed spatial-reliability-based constraints formulation over the recent alternatives (Kiani Galoogahi et al, 2015; Danelljan et al, 2015a).

4.3.1 Robustness to non-axis-aligned target initialization

The CSR-DCF tracker from Section 3 was compared to the original recent state-of-the-art trackers SRDCF (Danelljan et al, 2015a) and LBCF (Kiani Galoogahi et al, 2015) that apply alternative boundary constraints. For fair comparison, the source code of SRDCF and LBCF was obtained from the authors, all three trackers used only HoG features and tracked on the same single scale. An experiment was designed to evaluate initialization and tracking of non axis-aligned targets, which is the case for most realistic deforming and non-circular objects. Trackers were initialized on frames with non-axis aligned targets and left to track until the sequence end, resulting in a large number of tracking trajectories.

The VOT2015 dataset (Kristan et al, 2015) contains non-axis-aligned annotations, which allows automatic identification of tracker initialization frames, i.e., frames in which the ground truth bounding box significantly deviates from an axis-aligned approximation. Frames with overlap (intersection over union of predicted and ground-truth bounding boxes) of the ground truth and the axis-aligned approximation lower than 0.5 were identified and filtered to obtain a set of initialization frames at least hundred frames apart. This constraint fits half the typical short-term sequence length (Kristan et al, 2015) and reduces the potential correlation across the initializations

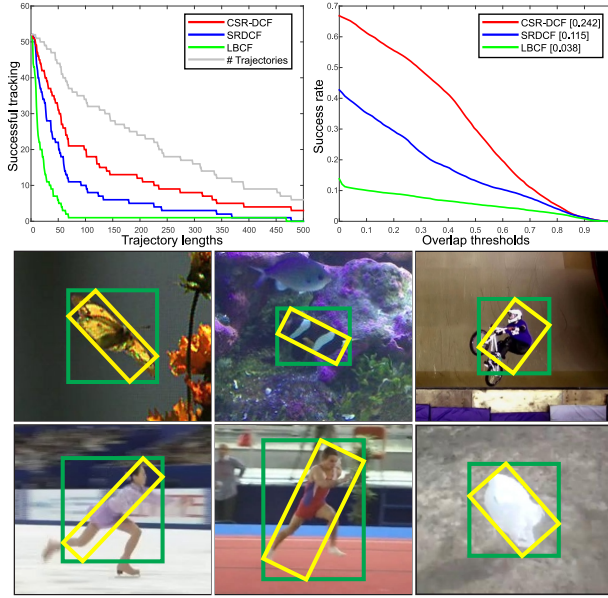


Figure 8: The number of trajectories with tracking successful up to frame Θ_{frm} (upper left), the success plots (upper right) and initialization examples of non-axis-aligned targets (bottom).

Table 1: Comparison of three most related trackers on non-axis-aligned initialization experiment: weighted average tracking length in frames Γ_{frm} and proportions Γ_{prp} , and weighted average overlaps using the original and axis-aligned ground truth, Φ_{rot} and Φ_{aa} , respectively.

Tracker	Γ_{prp}	Γ_{frm}	Φ_{aa}	Φ_{rot}
CSR-DCF	① 0.58	① 221	① 0.31	① 0.24
SRDCF (ICCV2015)	② 0.31	② 95	② 0.16	② 0.12
LBCF (CVPR2015)	③ 0.12	③ 37	③ 0.06	③ 0.04

(see Figure 8 (bottom) for examples).

Initialization robustness is estimated by counting the number of trajectories in which the tracker was still tracking (overlap with ground truth greater than 0) Θ_{frm} frames after initialization. The graph in Figure 8 (top-left) shows these values with increasing the threshold Θ_{frm} . The CSR-DCF graph is consistently above the SRDCF and LBCF for all thresholds. The performance is summarized

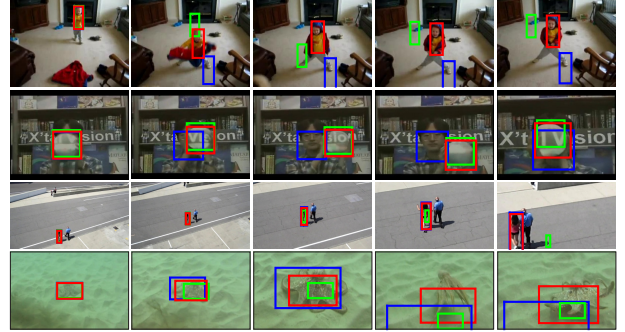


Figure 9: Qualitative results for trackers CSR-DCF (red) tracker, SRDCF (blue) and LBCF (green).

by the average tracking length (number of frames before the overlap drops to zero) weighted by trajectory lengths. The weighted average tracking lengths in frames, Γ_{frm} , and proportions of full trajectory lengths, Γ_{prp} , are shown in Table 1. The CSR-DCF by far outperforms SRDCF and LBCF in all measures indicating significant robustness in the initialization of challenging targets that deviate from axis-aligned templates. This improvement is further confirmed by the graph in Figure 8 (top-right) which shows the OTB success plots (Wu et al, 2013) calculated on these trajectories and summarized by the AUC values, which are equal to the average overlaps (Čehovin et al, 2016). Table 1 shows the average overlaps computed on the original ground truth on VOT2015 (Φ_{rot}) and on ground truth approximated by the axis-aligned bounding box (Φ_{aa}). Again, the CSR-DCF by far outperforms the competing alternatives SRDCF and LBCF. Tracking examples for the three trackers are shown in Figure 9.

In summary, the results show that the quality of spatial constraints significantly affects the relative tracking performance when a large portion of the training region in the target vicinity is occupied by background. The relative performance of LBCF (Kiani Galoogahi et al, 2015) is lowest among the three trackers since this tracker treats all pixels within axis-aligned bounding box equally as target. The SRDCF (Danelljan et al, 2015a) mostly focuses on the central pixels of the training region and suppresses the filter values at the borders, thus outperforming the LBCF (Kiani Galoogahi et al, 2015). The spatial reliability map in CSR-DCF most successfully reduces the in-

fluence of background in filter learning resulting in considerable robustness to poor initializations.

4.4 Spatial and channel reliability ablation study

An ablation study on VOT2016 was conducted to evaluate the contribution of spatial and channel reliability in CSR-DCF. Results of the VOT primary measure expected average overlap (EAO) and two supplementary measures accuracy and robustness (A, R) are summarized in Table 2. For the details of performance measures and evaluation protocol we refer the reader to the Section 4.7. Performance of the various modifications of CSR-DCF is discussed in the following.

Channel reliability weights. Setting the channel reliability weights to uniform values (CSR-DCF_{c-}) is equivalent to treating all channels as independent and equally important. The performance drop in EAO compared to CSR-DCF is 12%.

Spatial reliability map. Replacing the spatial reliability map in CSR-DCF by a constant map with uniform values within the bounding box and zeros elsewhere (CSR-DCF_{s^u}), results in a 21% drop in EAO. The other parts of the tracker remained unchanged in this experiment, including the channel reliability. This clearly shows the importance of our segmentation-based spatial reliability map estimation from Section 3.2.

Channel and spatial reliability. Making both replacements in the original tracker means that this version (CSR-DCF_{c-s^u}) does not use channel reliability weights and it uses uniform spatial reliability map (uniform values within the bounding box and zeros elsewhere). The performance drops by 24% compared to CSR-DCF. Removal of the uniform spatial reliability map from CSR-DCF_{c-s^u} results in the CSR-DCF_{c-s-} . This version reduces our tracker to a standard DCF with a large receptive field. Since the learned filter captures a significant amount of background, the performance drops by over 50%.

ADMM Filter optimization method. To demonstrate the importance of the constrained optimization method we modify the proposed tracker as follows. The filter \mathbf{h} is calculated with a naive approach, i.e., a closed-form solution followed by masking with the spatial reliability map \mathbf{m} : $\hat{\mathbf{h}} = \mathcal{F}(\mathcal{F}^{-1}(\hat{\mathbf{h}}) \odot \mathbf{m})$. For a fair comparison the tracker,

Table 2: Ablation study of CSR-DCF. The use of channel reliability is indicated in the *Chan.* column, the type of spatial reliability map in the *Spat.* column. The *Opt.* column indicates whether the constrained optimization is used.

Tracker	Chan.	Spat.	Opt.	EAO	R_{av}	A_{av}
CSR-DCF	x	segm.	x	① 0.338	① 0.85	① 0.51
CSR-DCF_{c-}	–	segm.	x	② 0.297	② 1.08	② 0.51
CSR-DCF_{s^u}	x	unif.	x	③ 0.264	③ 1.18	③ 0.49
CSR-DCF_{c-s^u}	–	unif.	x	0.256	1.33	② 0.51
CSR-DCF_{c-o-}	–	segm.	–	0.251	1.47	② 0.51
CSR-DCF_{c-s-}	–	–	–	0.152	2.85	0.47

denoted as CSR-DCF_{c-o-} , does not use channel reliability weights. The performance drop in EAO compared to CSR-DCF_{c-} is 15%.

4.5 Spatial reliability map quality analysis

In this section we evaluate the quality of our spatial reliability map estimation (Section 3.2) from a visual tracking perspective. We compare the CSR-DCF tracker with the version of CSR-DCF that uses *ideal* spatial reliability map (the tracker is denoted as $\text{CSR}^*\text{-DCF}$). In the VOT2016 challenge (Kristan et al, 2016b), the ground truth bounding boxes were automatically computed by optimizing coverage over manually segmented targets in each frame. The VOT2016 has recently made their per-frame segmentations freely available (Vojir and Matas, 2017). We use these per-frame segmentation masks in $\text{CSR}^*\text{-DCF}$ as spatial reliability map \mathbf{m} .

Results of evaluation on VOT2016 (Kristan et al, 2016b) are reported in Table 3. The performances of the CSR-DCF and $\text{CSR}^*\text{-DCF}$ are very similar. The trackers achieve an equal expected average overlap (EAO) and average accuracy (A_{av}). But the $\text{CSR}^*\text{-DCF}$ has a single failure less than CSR-DCF on 60 sequences which is 0.02 on average. In Table 3 the average number of failures is denoted as robustness (R_{av}). These results show that our approach for spatial reliability estimation (Section 3.2) generates near ideal maps from a tracking perspective.

Figure 10 qualitatively compares the spatial reliabil-

Table 3: Tracking performance comparison of the two versions of CSR-DCF on VOT2016. The proposed method is denoted as CSR-DCF while the version using ground-truth segmentation masks instead of color-based spatial reliability map is denoted as CSR*-DCF.

Tracker	EAO	A_{av}	R_{av}
CSR-DCF	0.338	0.51	0.85
CSR*-DCF	0.338	0.51	0.83



Figure 10: Qualitative comparison of the spatial reliability maps during tracking. The dashed bounding box represents area from which correlation filter is obtained. This is also the area where spatial reliability map is calculated. In addition, the ground-truth segmentation masks are visualized on the right side under each frame.

ity maps to the ground-truth segmentation masks on VOT2016 (Kristan et al, 2016b). Note that at pixel level,

the maps are different. But from the perspective of tracking they are nearly equivalent since the tracking performance remains unchanged. For example, in the case of a basketball player, the legs are not well segmented by our approach. But since the legs constantly move, they are in fact non-informative for object localization from the perspective of the correlation filter template matching and do not contribute to improved tracking.

4.6 The OTB100 benchmark (Wu et al, 2015)

The OTB100 (Wu et al, 2015) benchmark contains results of 29 trackers evaluated on 100 sequences by a no-reset evaluation protocol. Tracking quality is measured by precision and success plots. The success plot shows the fraction of frames with the overlap between the predicted and ground truth bounding box greater than a threshold with respect to all threshold values. The precision plot shows similar statistics on the center error. The results are summarized by areas under these plots. To reduce clutter, we show here only the results for top-performing recent baselines, i.e., Struck (Hare et al, 2011), TLD (Kalal et al, 2012), CXT (Dinh et al, 2011), ASLA (Xu Jia, 2012), SCM (Wei Zhong, 2012), LSK (Liu et al, 2011), CSK (Henriques et al, 2012) and results for recent top-performing state-of-the-art trackers SRDCF (Danelljan et al, 2015a) and MUSTER (Hong et al, 2015).

The CSR-DCF is ranked top on the benchmark (Fig. 11). It significantly outperforms the best performers reported in (Wu et al, 2015) and outperforms the current state-of-the-art SRDCF (Danelljan et al, 2015a) and MUSTER (Hong et al, 2015). The average CSR-DCF performance on success plot is slightly lower than SRDCF (Danelljan et al, 2015a) due to poorer scale estimation, but yields better performance in the average precision (center error). Both, precision and success plot, show that the CSR-DCF tracks on average longer than competing methods.

4.7 The VOT2015 benchmark (Kristan et al, 2015)

The VOT2015 (Kristan et al, 2015) benchmark contains results of 63 state-of-the-art trackers evaluated on 60 chal-

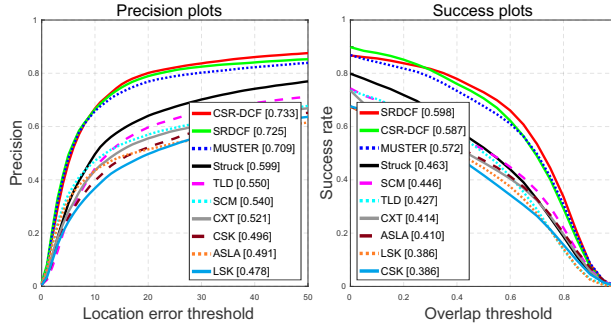


Figure 11: Evaluation on OTB100 (Wu et al, 2015) benchmark.

lensing sequences. In contrast to related benchmarks, the VOT2015 dataset was constructed from over 300 sequences by an advanced sequence selection methodology that favors objects difficult to track and maximizes a visual attribute diversity cost function (Kristan et al, 2015). This makes it arguably the most challenging sequence set available. The VOT methodology (Kristan et al, 2016c) resets a tracker upon failure to fully use the dataset. The basic VOT measures are the number of failures during tracking (robustness) and average overlap during the periods of successful tracking (accuracy), while the primary VOT2015 measure is the expected average overlap (EAO) on short-term sequences. The latter can be thought of as the expected no-reset average overlap (AUC in OTB methodology), but with reduced bias and the variance as explained in (Kristan et al, 2015).

Figure 12 shows the VOT EAO plots with the CSR-DCF and the VOT2015 state-of-the-art approaches considering the VOT2016 rules that do not consider trackers learned on video sequences related to VOT to prevent over-fitting. The CSR-DCF outperforms all trackers and achieves the top rank. The CSR-DCF significantly outperforms the related correlation filter trackers like SRDCF (Danelljan et al, 2015a) as well as trackers that apply computationally-intensive state-of-the-art deep features e.g., deepSRDCF (Danelljan et al, 2015b) and SO-DLT (Wang et al, 2015b). For completeness, detailed results for the ten top-performing trackers are shown in Table 4.

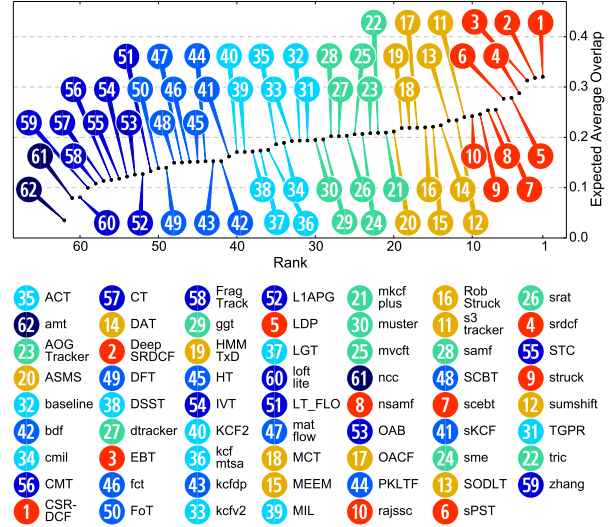


Figure 12: Expected average overlap (EAO) plot for CSR-DCF (#1) and all trackers participating in the VOT 2015 (Kristan et al, 2015) benchmark listed below the plot in alphabetical order with their numerical codes.

Table 4: The ten top-performing trackers on the VOT2015 benchmark.

Tracker	EAO	A_{av}	R_{av}
CSR-DCF	① 0.320	③ 0.55	② 0.93
DeepSRDCF	② 0.318	② 0.56	③ 1.00
EBT	③ 0.313	0.45	① 0.81
srdfc	0.288	③ 0.55	1.18
LDP	0.278	0.49	1.30
sPST	0.277	0.54	1.42
scebt	0.255	0.54	1.72
nsamf	0.254	0.53	1.45
struck	0.246	0.46	1.50
rajssc	0.242	① 0.57	1.75

4.8 The VOT2016 benchmark (Kristan et al, 2016b)

Finally, we assess our tracker on the most recent visual tracking benchmark, VOT2016 (Kristan et al, 2016b). The dataset contains 60 sequences from VOT2015 (Kris-

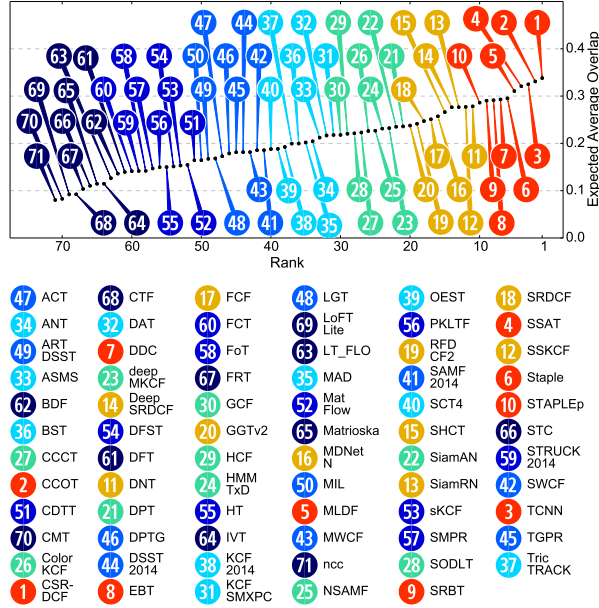


Figure 13: Expected average overlap (EAO) plot for CSR-DCF (#1) and all trackers participating in the VOT 2016 (Kristan et al, 2016b) benchmark listed below the plot in alphabetical order with their numerical codes.

Table 5: The ten top-performing trackers on the VOT2016 benchmark.

Tracker	EAO	A_{av}	R_{av}
CSR-DCF	① 0.338	0.51	② 0.85
CCOT	② 0.331	0.52	② 0.85
TCNN	③ 0.325	③ 0.54	0.96
SSAT	0.321	① 0.57	1.04
MLDF	0.311	0.48	① 0.83
Staple	0.295	③ 0.54	1.35
DDC	0.293	0.53	1.23
EBT	0.291	0.44	③ 0.90
SRBT	0.290	0.50	1.25
STAPLEp	0.286	② 0.55	1.32

tan et al, 2015) with improved annotations. The benchmark evaluated a set of 70 trackers which includes the recently published and yet unpublished state-of-the-art trackers. The set is indeed diverse, the top-

performing trackers come from various classes e.g., correlation filter methods: CCOT (Danelljan et al, 2016), Staple (Bertinetto et al, 2016a), DDC (Kristan et al, 2016b), deep convolutional network based: TCNN (Kristan et al, 2016b), SSAT (Kristan et al, 2016b; Nam and Han, 2016), MLDF (Kristan et al, 2016b; Wang et al, 2015a), FastSiamnet (Bertinetto et al, 2016b) and different detection-based approaches: EBT (Zhu et al, 2016) and SRBT (Kristan et al, 2016b).

Figure 13 shows the EAO performance on the VOT2016. The CSR-DCF outperforms all 70 trackers with the EAO score equal to 0.338. The CSR-DCF significantly outperforms correlation filter approaches that do not apply deep ConvNets. Even though the CSR-DCF applies only simple features, it outperforms all trackers that apply computationally intensive deep features. Detailed performance scores for the ten top-performing trackers are shown in Table 5.

4.9 Per-attribute analysis

The VOT2016 (Kristan et al, 2016b) dataset is per-frame annotated with visual attributes and allows detailed analysis of per-attribute tracking performance. Figure 14 shows per-attribute plot for ten top-performing trackers on VOT2016 in EAO. The CSR-DCF is consistently ranked among top three trackers on five out of six attributes. In four attributes (size change, occlusion, camera motion, unassigned) the tracker is ranked top. The only attribute on which our CSR-DCF is outperformed by four of the compared trackers (MLDF, CCOT, SSAT and TCNN) is illumination change. All of these trackers use deep CNN features which are much more complex than the features in CSR-DCF and better handle illumination change.

4.10 Tracking speed analysis

Tracking speed is an important factor of many real-world tracking problems. Table 6 thus compares several related and well-known trackers (including the best-performing tracker on the VOT2016 challenge) in terms of speed and VOT performance measures. Speed measurements on a single CPU are computed on Intel Core i7 3.4GHz standard desktop.

The CSR-DCF performs on par with the VOT2016 best-performing CCOT (Danelljan et al, 2016), which

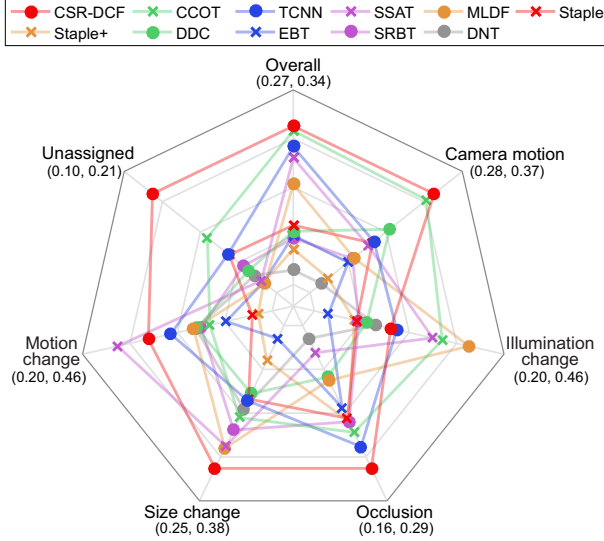


Figure 14: Expected averaged overlap performance on different visual attributes on the VOT2016 (Kristan et al, 2016b) benchmark. The CSR-DCF and the top 10 performing trackers from VOT2016 are shown. The scales of visual attribute axes are shown below the attribute labels.

applies deep ConvNets, with respect to VOT measures, while being 20 times faster than the CCOT. The CCOT was modified by replacing the computationally intensive deep features with the same simple features used in CSR-DCF. The resulting tracker, indicated by CCOT*, is still ten times slower than CSR-DCF, while the performance drops by over 15%. The CSR-DCF performs twice as fast as the related SRDCF (Danelljan et al, 2015a), while achieving approximately 25% better tracking results. The speed of baseline real-time trackers like DSST (Danelljan et al, 2014a) and Struck (Hare et al, 2011) is comparable to CSR-DCF, but their tracking performance is significantly poorer. The fastest compared tracker, KCF (Henriques et al, 2015) runs much faster than real-time, but delivers a significantly poorer performance than CSR-DCF.

The experiments show that the CSR-DCF tracks comparably to the state-of-the-art trackers which apply computationally demanding high-dimensional features, but runs considerably faster and delivers top tracking performance among the real-time trackers.

Table 6: Speed in frames per second (fps) of correlation trackers and Struck – a baseline. The EAO, average accuracy (A_{av}) and average failures (R_{av}) are shown for reference.

Tracker	EAO	A_{av}	R_{av}	fps
CSR-DCF	① 0.338	② 0.51	① 0.85	③ 13.0
CCOT	② 0.331	① 0.52	① 0.85	0.6
CCOT*	③ 0.274	① 0.52	② 1.18	1.0
SRDCF	0.247	① 0.52	③ 1.50	7.3
KCF	0.192	③ 0.48	2.03	① 115.7
DSST	0.181	③ 0.48	2.52	② 18.6
Struck	0.142	0.42	3.37	8.5

The average speed of our tracker measured on the VOT 2016 dataset is approximately 13 frames-per-second² or 77 milliseconds per-frame. Figure 15 shows the processing time required by each step of the CSR-DCF. A tracking iteration is divided into two steps: (i) target localization and (ii) the visual model update. Target localization takes in average 35 milliseconds at each frame and is composed of two sub-steps: estimation of object translation (23ms) and scale change estimation (12ms). The visual model update step takes on average 42 milliseconds. It consists of three sub-steps: spatial reliability map estimation (16ms), filter update (12ms) and scale model update (14ms). Filter optimization, which is part of the filter update step, takes on average 7 milliseconds.

4.11 Qualitative evaluation

Figure 16 shows four examples of tracking with the CSR-DCF. In the following we describe tracking performance on each sequence.

The first example shows tracking of an octopus along with channel reliability weights. The first eighteen weights correspond to HoG channels, the 19th weight is reliability of a grayscale template and the last ten weights correspond to colornames. Note that the colors in boxes are not the actual colors of the colornames, because these features are subspace of original colornames, designed to

²With some basic code optimization and refactoring we speed-up our algorithm to 19 FPS without significant performance drop (only one additional failure on VOT2016 dataset).

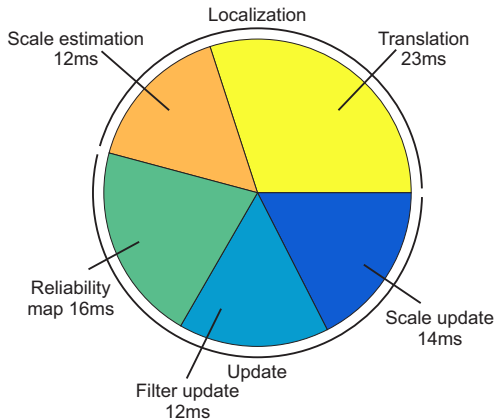


Figure 15: A single iteration processing time decomposed across the main steps of the CSR-DCF.

improve correlation filter tracking (see (Danelljan et al, 2014b)). Observe that when the octopus changes shape significantly, some channels become more discriminative than the others – this is particularly evident in the first eighteen channels that represent the HoG features.

Tracking a gymnast is shown in the second example. The target is deforming and rotating over the sequence significantly, while our tracker is able to successfully track it. Additionally, the correlation response from the localization step is shown for each frame. The peak in the response is well expressed, which means that the filter accurately represents the target and that the discriminative channels overrule the less discriminative ones by our channel reliability estimation approach.

The third example shows tracking a sprinter. The spatial reliability map is visualized next to each frame. In the bottom-right corner of each frame the tracking patch is overlaid with the spatial reliability map. The reliability maps fit the target well and prevent the filter from learning the background.

The last example shows tracking under occlusion, i.e., a motorcyclist driving on the road while being repeatedly occluded by the trees. This example demonstrates that our tracker is robust to short-term full occlusions and that it is able to recover and localize the target despite the full occlusion. This is possible due to the robust learning with channel and spatial reliability map and the large capture range that our learning scheme provides.

5 Conclusion

The Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF) was introduced. The spatial reliability map adapts the filter support to the part of the object suitable for tracking which overcomes both the problems of circular shift enabling an arbitrary search region size and the limitations related to the rectangular shape assumption. A novel efficient spatial reliability map estimation method was proposed and an efficient optimization procedure was used for learning a correlation filter with the support constrained by the estimated map. The second novelty of CSR-DCF is the channel reliability. The reliability is estimated from the properties of the constrained least-squares solution. The channel reliability scores were used for weighting the per-channel filter responses in localization.

Experimental comparison with recent related state-of-the-art boundary-constraints formulations showed significant benefits of using our formulation. The CSR-DCF achieves state-of-the-art performance on standard benchmarks: OTB100 (Wu et al, 2015), VOT2015 (Kristan et al, 2015) and VOT2016 (Kristan et al, 2016b) while running close to the real-time on a single CPU. Despite using simple features like HoG and Colornames, the CSR-DCF performs on par with trackers that apply computationally complex deep ConvNet, but is significantly faster.

To the best of our knowledge, the proposed approach is the first of its kind to introduce constrained filter learning with arbitrary spatial reliability map and the use of channel reliabilities. The spatial and channel reliability formulation is general and can be used in most modern correlation filters, e.g. those using deep features.

6 Appendix 1: Derivation of the augmented Lagrangian minimizer

This section provides a complete derivation of the relations (12, 13) in the Section 3.1. The augmented La-

grangian from Equation (8) is

$$\mathcal{L}(\hat{\mathbf{h}}_c, \mathbf{h}, \hat{\mathbf{l}}) = \|\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{g}}\|^2 + \frac{\lambda}{2}\|\mathbf{h}_m\|^2 + \quad (20)$$

$$[\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \hat{\mathbf{h}}_m) + \overline{\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \hat{\mathbf{h}}_m)}] + \mu\|\hat{\mathbf{h}}_c - \hat{\mathbf{h}}_m\|^2,$$

with $\mathbf{h}_m = (\mathbf{m} \odot \mathbf{h})$. For the purposes of derivation we will rewrite (20) into a fully vectorized form

$$\mathcal{L}(\hat{\mathbf{h}}_c, \mathbf{h}, \hat{\mathbf{l}}) = \|\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{g}}\|^2 + \frac{\lambda}{2}\|\mathbf{h}_m\|^2 + \quad (21)$$

$$\left[\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}) + \overline{\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h})} \right] +$$

$$\mu\|\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}\|^2,$$

where \mathbf{F} denotes $D \times D$ orthonormal matrix of Fourier coefficients, such that the Fourier transform is defined as $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x}) = \sqrt{D}\mathbf{F}\mathbf{x}$ and $\mathbf{M} = \text{diag}(\mathbf{m})$. For clearer representation we denote the four terms in the summation (21) as

$$\mathcal{L}(\hat{\mathbf{h}}_c, \mathbf{h}, \hat{\mathbf{l}}) = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_4, \quad (22)$$

where

$$\mathcal{L}_1 = \overline{(\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{g}})}^T (\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{g}}), \quad (23)$$

$$\mathcal{L}_2 = \frac{\lambda}{2}\|\mathbf{h}_m\|^2, \quad (24)$$

$$\mathcal{L}_3 = \hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}) + \overline{\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h})}, \quad (25)$$

$$\mathcal{L}_4 = \mu\|\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}\|^2. \quad (26)$$

Minimization of Equation (8) in Section 3.1 is an iterative process at which the following minimizations are required:

$$\hat{\mathbf{h}}_c^{\text{opt}} = \arg \min_{\mathbf{h}_c} \mathcal{L}(\hat{\mathbf{h}}_c, \mathbf{h}, \hat{\mathbf{l}}), \quad (27)$$

$$\mathbf{h}^{\text{opt}} = \arg \min_{\mathbf{h}} \mathcal{L}(\hat{\mathbf{h}}_c^{\text{opt}}, \mathbf{h}, \hat{\mathbf{l}}). \quad (28)$$

Minimization w.r.t. to $\hat{\mathbf{h}}_c$ is derived by finding $\hat{\mathbf{h}}_c$ at which the complex gradient of the augmented Lagrangian vanishes, i.e.,

$$\nabla_{\hat{\mathbf{h}}_c} \mathcal{L} \equiv 0, \quad (29)$$

$$\nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_1 + \nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_2 + \nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_3 + \nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_4 \equiv 0. \quad (30)$$

The partial complex gradients are:

$$\nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_1 = \quad (31)$$

$$= \frac{\partial}{\partial \hat{\mathbf{h}}_c} \left[\overline{(\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{g}})}^T (\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{g}}) \right] =$$

$$= \frac{\partial}{\partial \hat{\mathbf{h}}_c} \left[\hat{\mathbf{h}}_c^T \text{diag}(\hat{\mathbf{f}})^H \text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \hat{\mathbf{h}}_c^T \text{diag}(\hat{\mathbf{f}})^H \hat{\mathbf{g}} - \right.$$

$$\left. \hat{\mathbf{g}}^H \text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c + \hat{\mathbf{g}}^H \hat{\mathbf{g}} \right] =$$

$$= \text{diag}(\hat{\mathbf{f}})^H \text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \text{diag}(\hat{\mathbf{f}})\hat{\mathbf{g}},$$

$$\nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_2 = 0, \quad (32)$$

$$\nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_3 = \quad (33)$$

$$= \frac{\partial}{\partial \hat{\mathbf{h}}_c} \left[\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}) + \overline{\hat{\mathbf{l}}^H(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h})} \right] =$$

$$= \frac{\partial}{\partial \hat{\mathbf{h}}_c} \left[\hat{\mathbf{l}}^H \hat{\mathbf{h}}_c - \hat{\mathbf{l}}^H \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h} + \hat{\mathbf{l}}^T \hat{\mathbf{h}}_c - \hat{\mathbf{l}}^T \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h} \right] =$$

$$= \hat{\mathbf{l}},$$

$$\nabla_{\hat{\mathbf{h}}_c} \mathcal{L}_4 = \quad (34)$$

$$= \frac{\partial}{\partial \hat{\mathbf{h}}_c} \left[\mu \overline{(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h})}^T (\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}) \right] =$$

$$= \frac{\partial}{\partial \hat{\mathbf{h}}_c} \left[\mu (\hat{\mathbf{h}}_c^H \hat{\mathbf{h}}_c - \hat{\mathbf{h}}_c^H \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h} - \right.$$

$$\left. \sqrt{D}\mathbf{h}^T \mathbf{M}\mathbf{F}^H \hat{\mathbf{h}}_c + D\mathbf{h}^T \mathbf{M}\mathbf{F}^H \mathbf{F}\mathbf{M}\mathbf{h}) \right] =$$

$$= \mu \hat{\mathbf{h}}_c - \mu \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}.$$

Note that $\sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h} = \hat{\mathbf{h}}_m$ according to our original definition of $\hat{\mathbf{h}}_m$. Plugging (31-34) into (30) yields

$$\text{diag}(\hat{\mathbf{f}})^H \text{diag}(\hat{\mathbf{f}})\hat{\mathbf{h}}_c - \text{diag}(\hat{\mathbf{f}})\hat{\mathbf{g}} + \hat{\mathbf{l}} + \mu \hat{\mathbf{h}}_c - \mu \hat{\mathbf{h}}_m = 0, \quad (35)$$

$$\hat{\mathbf{h}}_c = \frac{\text{diag}(\hat{\mathbf{f}})\hat{\mathbf{g}} + \mu \hat{\mathbf{h}}_m - \hat{\mathbf{l}}}{\text{diag}(\hat{\mathbf{f}})^H \text{diag}(\hat{\mathbf{f}}) + \mu},$$

which can be rewritten into

$$\hat{\mathbf{h}}_c = \frac{\hat{\mathbf{f}} \odot \bar{\hat{\mathbf{g}}} + \mu \hat{\mathbf{h}}_m - \hat{\mathbf{I}}}{\bar{\hat{\mathbf{f}}} \odot \hat{\mathbf{f}} + \mu}. \quad (36)$$

Next we derive the closed-form solution of (28). The optimal \mathbf{h} is obtained when the complex gradient w.r.t. \mathbf{h} vanishes, i.e.,

$$\nabla_{\bar{\mathbf{h}}} \mathcal{L} \equiv 0 \quad (37)$$

$$\nabla_{\bar{\mathbf{h}}} \mathcal{L}_1 + \nabla_{\bar{\mathbf{h}}} \mathcal{L}_2 + \nabla_{\bar{\mathbf{h}}} \mathcal{L}_3 + \nabla_{\bar{\mathbf{h}}} \mathcal{L}_4 \equiv 0. \quad (38)$$

The partial gradients are

$$\nabla_{\bar{\mathbf{h}}} \mathcal{L}_1 = 0, \quad (39)$$

$$\begin{aligned} \nabla_{\bar{\mathbf{h}}} \mathcal{L}_2 &= \\ &= \frac{\partial}{\partial \bar{\mathbf{h}}} \left[\frac{\lambda}{2} (\bar{\mathbf{M}}\mathbf{h})^T (\mathbf{M}\mathbf{h}) \right] = \frac{\partial}{\partial \bar{\mathbf{h}}} \left[\frac{\lambda}{2} \mathbf{h}^H \bar{\mathbf{M}}\mathbf{M}\mathbf{h} \right]. \end{aligned} \quad (40)$$

Since we defined mask \mathbf{m} as a binary mask, the product $\bar{\mathbf{M}}\mathbf{M}$ can be simplified into \mathbf{M} and the result for $\nabla_{\bar{\mathbf{h}}} \mathcal{L}_2$ is

$$\nabla_{\bar{\mathbf{h}}} \mathcal{L}_2 = \frac{\lambda}{2} \mathbf{M}\mathbf{h}. \quad (41)$$

The remaining gradients are as follows:

$$\begin{aligned} \nabla_{\bar{\mathbf{h}}} \mathcal{L}_3 &= \\ &= \frac{\partial}{\partial \bar{\mathbf{h}}} \left[\hat{\mathbf{I}}^H (\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}) + \overline{\hat{\mathbf{I}}^H (\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h})} \right] = \\ &= \frac{\partial}{\partial \bar{\mathbf{h}}} \left[\hat{\mathbf{I}}^H \hat{\mathbf{h}}_c - \hat{\mathbf{I}}^H \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h} + \hat{\mathbf{I}}^T \bar{\hat{\mathbf{h}}}_c - \hat{\mathbf{I}}^T \sqrt{D}\bar{\mathbf{F}}\bar{\mathbf{M}}\bar{\mathbf{h}} \right] = \\ &= -\sqrt{D}\mathbf{M}\mathbf{F}^H \hat{\mathbf{I}}, \end{aligned} \quad (42)$$

$$\begin{aligned} \nabla_{\bar{\mathbf{h}}} \mathcal{L}_4 &= \\ &= \frac{\partial}{\partial \bar{\mathbf{h}}} \left[\mu \overline{(\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h})}^T (\hat{\mathbf{h}}_c - \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h}) \right] = \\ &= \frac{\partial}{\partial \bar{\mathbf{h}}} \left[\mu (\hat{\mathbf{h}}_c^H \hat{\mathbf{h}}_c - \hat{\mathbf{h}}_c^H \sqrt{D}\mathbf{F}\mathbf{M}\mathbf{h} - \sqrt{D}\mathbf{h}^H \mathbf{M}\mathbf{F}^H \hat{\mathbf{h}}_c + D\mathbf{h}^H \mathbf{M}\mathbf{h}) \right] = \\ &= -\mu \sqrt{D}\mathbf{M}\mathbf{F}^H \hat{\mathbf{h}}_c + \mu D\mathbf{M}\mathbf{h}. \end{aligned} \quad (43)$$

Plugging (39-43) into (38) yields

$$\frac{\lambda}{2} \mathbf{M}\mathbf{h} - \sqrt{D}\mathbf{M}\mathbf{F}^H \hat{\mathbf{I}} - \mu \sqrt{D}\mathbf{M}\mathbf{F}^H \hat{\mathbf{h}}_c + \mu D\mathbf{M}\mathbf{h} = 0, \quad (44)$$

$$\mathbf{M}\mathbf{h} = \mathbf{M} \frac{\sqrt{D}\mathbf{F}^H (\hat{\mathbf{I}} + \mu \hat{\mathbf{h}}_c)}{\frac{\lambda}{2} + \mu D}.$$

Using the definition of the inverse Fourier transform, i.e., $\mathcal{F}^{-1}(\hat{\mathbf{x}}) = \frac{1}{\sqrt{D}} \mathbf{F}^H \hat{\mathbf{x}}$, (44) can be rewritten into

$$\mathbf{m} \odot \mathbf{h} = \mathbf{m} \odot \frac{\mathcal{F}^{-1}(\hat{\mathbf{I}} + \mu \hat{\mathbf{h}}_c)}{\frac{\lambda}{2D} + \mu}. \quad (45)$$

The values in \mathbf{m} are either zero or one. Elements in \mathbf{h} that correspond to the zeros in \mathbf{m} can in principle not be recovered from (45) since this would result in division by zero. But our initial definition of the problem was to seek solutions for the filter that satisfies the following relation $\mathbf{h} \equiv \mathbf{h} \odot \mathbf{m}$. This means the values corresponding to zeros in \mathbf{m} should be zero in \mathbf{h} . Thus the proximal solution to (45) is

$$\mathbf{h} = \mathbf{m} \odot \frac{\mathcal{F}^{-1}(\hat{\mathbf{I}} + \mu \hat{\mathbf{h}}_c)}{\frac{\lambda}{2D} + \mu}. \quad (46)$$

Acknowledgements

This work was supported in part by the following research programs and projects: Slovenian research agency research programs and projects P2-0214 and L2-6765. Jiří Matas and Tomáš Vojtř were supported by The Czech Science Foundation Project GACR P103/12/G084 and Toyota Motor Europe. We would also like to thank dr. Rok Žitko for discussion on complex differentiation.

References

- Babenko B, Yang MH, Belongie S (2011) Robust object tracking with online multiple instance learning. *IEEE Trans Pattern Anal Mach Intell* 33(8):1619–1632
- Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr PHS (2016a) Staple: Complementary learners for real-time tracking. In: *Comp. Vis. Patt. Recognition*, pp 1401–1409

- Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH (2016b) Fully-convolutional siamese networks for object tracking. arXiv preprint arXiv:1606.09549
- Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters. In: *Comp. Vis. Patt. Recognition*, IEEE, pp 2544–2550
- Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122
- Čehovin L, Leonardis A, Kristan M (2016) Visual object tracking performance measures revisited. *IEEE Trans Image Proc* 25(3):1261–1274
- Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *Comp. Vis. Patt. Recognition*, vol 1, pp 886–893
- Danelljan M, Häger G, Khan FS, Felsberg M (2014a) Accurate scale estimation for robust visual tracking. In: *Proc. British Machine Vision Conference*, pp 1–11
- Danelljan M, Khan FS, Felsberg M, van de Weijer J (2014b) Adaptive color attributes for real-time visual tracking. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23–28, 2014*, pp 1090–1097
- Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015a) Learning spatially regularized correlation filters for visual tracking. In: *Int. Conf. Computer Vision*, pp 4310–4318
- Danelljan M, Häger G, Khan FS, Felsberg M (2015b) Convolutional features for correlation filter based visual tracking. In: *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp 621–629
- Danelljan M, Robinson A, Khan FS, Felsberg M (2016) Beyond correlation filters: learning continuous convolution operators for visual tracking. In: *Proc. European Conf. Computer Vision*, Springer, pp 472–488
- Danelljan M, Häger G, Khan FS, Felsberg M (2017) Discriminative scale space tracking. *IEEE Trans Pattern Anal Mach Intell* 39(8):1561–1575
- Dinh TB, Vo N, Medioni G (2011) Context tracker: Exploring supporters and distracters in unconstrained environments. In: *Comp. Vis. Patt. Recognition*, pp 1177–1184
- Diplaros A, Vlassis N, Gevers T (2007) A spatially constrained generative model and an em algorithm for image segmentation. *IEEE Trans Neural Networks* 18(3):798 – 808
- Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. *IEEE Trans Pattern Anal Mach Intell* 32(9):1627–1645
- Galoogahi HK, Sim T, Lucey S (2013) Multi-channel correlation filters. In: *Int. Conf. Computer Vision*, pp 3072–3079
- Grabner H, Grabner M, Bischof H (2006) Real-time tracking via on-line boosting. In: *Proc. British Machine Vision Conference*, vol 1, pp 47–56
- Hare S, Saffari A, Torr PHS (2011) Struck: Structured output tracking with kernels. In: *Int. Conf. Computer Vision*, IEEE Computer Society, Washington, DC, USA, pp 263–270
- Henriques JF, Caseiro R, Martins P, Batista J (2012) Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C (eds) *Proc. European Conf. Computer Vision*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 702–715
- Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell* 37(3):583–596
- Hester CF, Casasent D (1980) Multivariant technique for multiclass pattern recognition. *Applied Optics* 19(11):1758–1761
- Hong Z, Chen Z, Wang C, Mei X, Prokhorov D, Tao D (2015) Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: *Comp. Vis. Patt. Recognition*, pp 749–758

- Kalal Z, Mikolajczyk K, Matas J (2012) Tracking-learning-detection. *IEEE Trans Pattern Anal Mach Intell* 34(7):1409–1422
- Kiani Galoogahi H, Sim T, Lucey S (2015) Correlation filters with limited boundaries. In: *Comp. Vis. Patt. Recognition*, pp 4630–4638
- Kristan M, Pflugfelder R, Leonardis A, Matas J, Porikli F, Čehovin L, Nebehay G, Fernandez G, Vojir Tea (2013) The visual object tracking vot2013 challenge results. In: *Vis. Obj. Track. Challenge VOT2013*, In conjunction with ICCV2013, pp 98–111
- Kristan M, Pflugfelder R, Leonardis A, Matas J, Čehovin L, Nebehay G, Vojir T, et al Fernandez G (2014) The visual object tracking vot2014 challenge results. In: *Proc. European Conf. Computer Vision*, pp 191–217
- Kristan M, Matas J, Leonardis A, Felsberg M, Čehovin L, Fernandez G, Vojir T, Häger G, Nebehay G, et al Pflugfelder R (2015) The visual object tracking vot2015 challenge results. In: *Int. Conf. Computer Vision*
- Kristan M, Kenk VS, Kovačič S, Perš J (2016a) Fast image-based obstacle detection from unmanned surface vehicles. *IEEE Transactions on Cybernetics* 46(3):641–654
- Kristan M, Leonardis A, Matas J, Felsberg M, Pflugfelder R, Čehovin L, Vojir T, Häger G, Lukežič A, et al Fernandez G (2016b) The visual object tracking vot2016 challenge results. In: *Proc. European Conf. Computer Vision*
- Kristan M, Matas J, Leonardis A, Vojir T, Pflugfelder R, Fernandez G, Nebehay G, Porikli F, Čehovin L (2016c) A novel performance evaluation methodology for single-target trackers. *IEEE Trans Pattern Anal Mach Intell*
- Li Y, Zhu J (2014a) A scale adaptive kernel correlation filter tracker with feature integration. In: *Proc. European Conf. Computer Vision*, pp 254–265
- Li Y, Zhu J (2014b) A scale adaptive kernel correlation filter tracker with feature integration. In: *Proc. European Conf. Computer Vision*, pp 254–265
- Liang P, Blasch E, Ling H (2015) Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans Image Proc* 24(12):5630–5644
- Liu B, Huang J, Yang L, Kulikowsk C (2011) Robust tracking using local sparse appearance model and k-selection. In: *Comp. Vis. Patt. Recognition*, pp 1313–1320
- Liu S, Zhang T, Cao X, Xu C (2016) Structural correlation filter for robust visual tracking. In: *Comp. Vis. Patt. Recognition*, pp 4312–4320
- Liu T, Wang G, Yang Q (2015) Real-time part-based visual tracking via adaptive correlation filters. In: *Comp. Vis. Patt. Recognition*, pp 4902–4912
- Lukežič A, Č Zajc L, Kristan M (2017) Deformable parts correlation filters for robust visual tracking. *IEEE Transactions on Cybernetics* PP(99):1–13
- Ma C, Huang JB, Yang X, Yang MH (2015) Hierarchical convolutional features for visual tracking. In: *Int. Conf. Computer Vision*, pp 3074–3082
- Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for uav tracking. In: *Proc. European Conf. Computer Vision*
- Nam H, Han B (2016) Learning multi-domain convolutional neural networks for visual tracking. In: *Comp. Vis. Patt. Recognition*, pp 4293–4302
- Qi Y, Zhang S, Qin L, Yao H, Huang Q, Lim J, Yang MH (2016) Hedged deep tracking. In: *CVPR*, pp 4303–4311
- Smeulders A, Chu D, Cucchiara R, Calderara S, Dehghan A, Shah M (2014) Visual tracking: An experimental survey. *IEEE Trans Pattern Anal Mach Intell* 36(7):1442–1468
- Vojir T, Matas J (2017) Pixel-wise object segmentations for the VOT 2016 dataset. Research Report CTU–CMP–2017–01, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic

- Wang L, Ouyang W, Wang X, Lu H (2015a) Visual tracking with fully convolutional networks. In: Int. Conf. Computer Vision, pp 3119–3127
- Wang N, Li S, Gupta A, Yeung D (2015b) Transferring rich feature hierarchies for robust visual tracking. CoRR abs/1501.04587
- Wang S, Zhang S, Liu W, Metaxas DN (2016) Visual tracking with reliable memories. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp 3491–3497
- Wei Zhong MHY Huchuan Lu (2012) Robust object tracking via sparsity-based collaborative model. In: Comp. Vis. Patt. Recognition, pp 1838–1845
- van de Weijer J, Schmid C, Verbeek J, Larlus D (2009) Learning color names for real-world applications. IEEE Trans Image Proc 18(7):1512–1523
- Wu Y, Lim J, Yang MH (2013) Online object tracking: A benchmark. In: Comp. Vis. Patt. Recognition, pp 2411–2418
- Wu Y, Lim J, Yang MH (2015) Object tracking benchmark. IEEE Trans Pattern Anal Mach Intell 37(9):1834–1848
- Xu Jia MHY Huchuan Lu (2012) Visual tracking via adaptive structural local sparse appearance model. In: Comp. Vis. Patt. Recognition, pp 1822–1829
- Zhang K, Zhang L, Liu Q, Zhang D, Yang MH (2014) Fast visual tracking via dense spatio-temporal context learning. In: Proc. European Conf. Computer Vision, Springer International Publishing, pp 127–141
- Zhu G, Porikli F, Li H (2016) Beyond local search: Tracking objects everywhere with instance-specific proposals. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 943–951

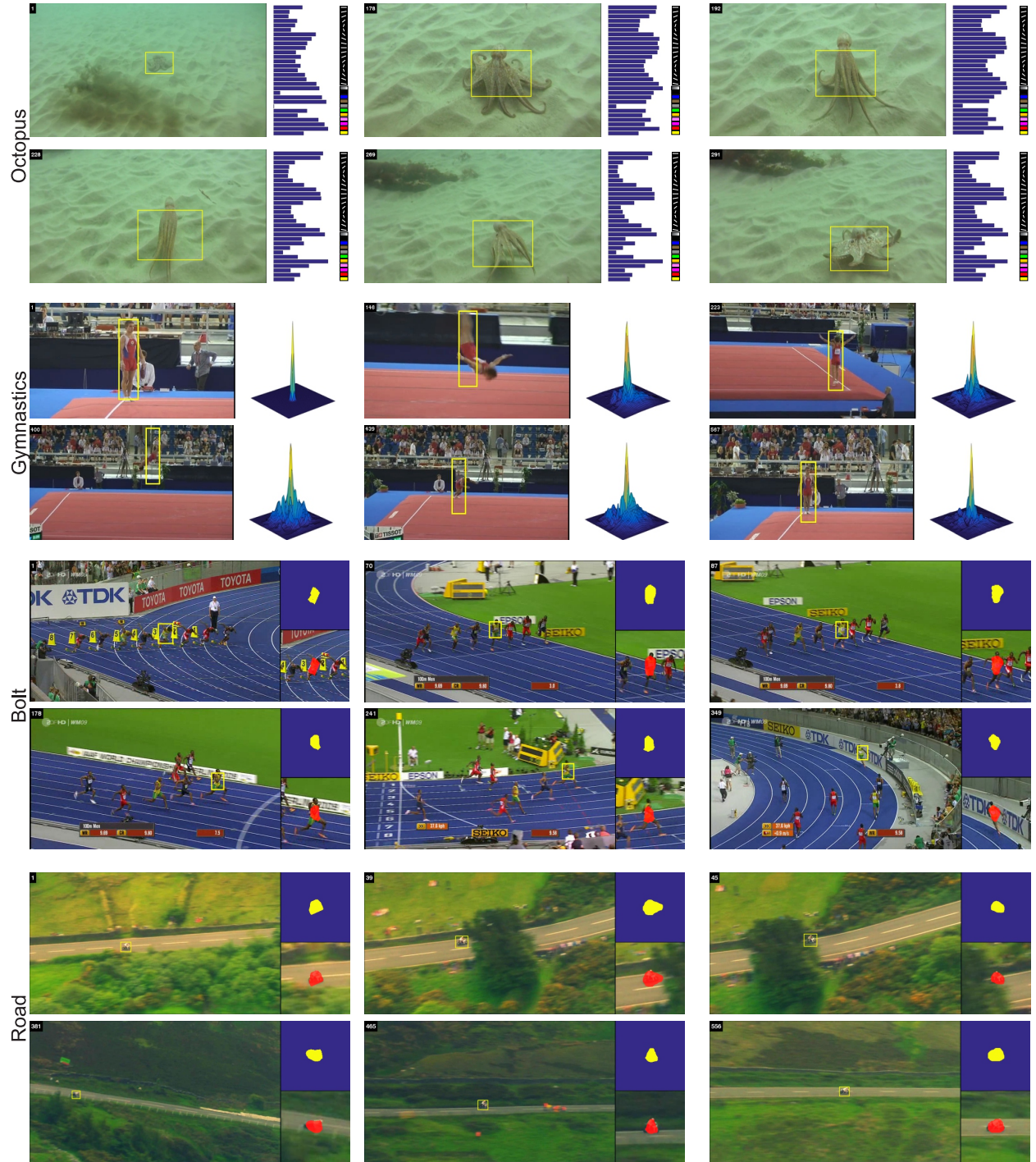


Figure 16: Qualitative results of tracking with the CSR-DCF on four video sequences.