Weighted Update and Comparison for Channel-Based Distribution Field Tracking

Kristoffer Öfjäll and Michael Felsberg

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University, Linköping

Abstract. There are three major issues for visual object trackers: model representation, search and model update. In this paper we address the last two issues for a specific model representation, grid based distribution models by means of channel-based distribution fields. Particularly we address the comparison part of searching. Previous work in the area has used standard methods for comparison and update, not exploiting all the possibilities of the representation. In this work we propose two comparison schemes and one update scheme adapted to the distribution model. The proposed schemes significantly improve the accuracy and robustness on the Visual Object Tracking (VOT) 2014 Challenge dataset.

1 Introduction and Related Work

For online appearance-based object tracking, there are three primary concerns: how to represent the object to be tracked (model), how to find the object in a new frame (search/comparison) and finally how to update the model given the information obtained from the new frame (update). These are not independent, choosing one component influences the choice of the other two. There are other approaches to tracking, such as using a classifier for discriminating the target object from the background, however, only template-based methods will be considered here.

Several different categories of target models for representing the tracked object have been proposed in literature. One obvious appearance-based representation of the object is by means of an image patch cut out from the first frame according to the bounding box defining the object to be tracked. The locations of the object in the following frames are estimated by finding patches best corresponding to this target patch, employing some suitable distance function. Letting this simple model be linearly updated after every frame leads to a first order (weighted mean) model. A natural extension is a second order (Gaussian) approximation, where also the variance of each pixel is estimated.

Another approach is to represent the full distribution of values within the target patch, illustrated in Fig. 1. Such a tracker, Distribution Field Tracking, DFT, was proposed by Sevilla et al. [13] where histograms are used for representing distributions. However, as was shown by Felsberg [4], replacing the histograms

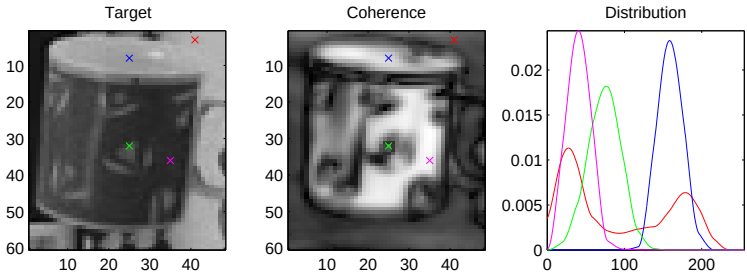


Fig. 1. Target and target model representation at the end of the VOT2013 cup-sequence. Left: found target patch. Middle: coherence of the target model (black: low, white: high), see Sect. 4.2. Right: represented pixel value distributions for a selection of points marked in left and middle images. Large coherence correspond to static pixel values on the tracked object and narrow distributions (blue, magenta). Low coherence correspond to background pixels (red, multimodal distribution) and varying pixels on the target (green, single wide mode).

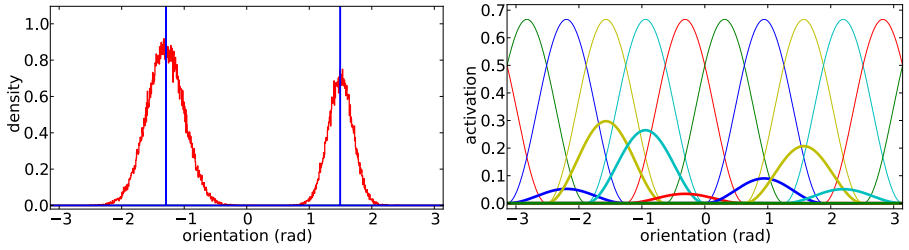
with channel representations [6] increases tracker performance, resulting in the Enhanced Distribution Field Tracker, EDFT.

In both cases, the model update is performed by a linear convex combination and the comparison uses an L_1 norm. However, the distribution view of the channel representation allows for other types of comparisons and update schemes compared to the direct pixel value representation. These possibilities were not used in previously proposed trackers.

In this work we evaluate a novel update scheme and novel comparison methods, exploiting the potential of the channel representation. We restrict ourselves to online methods implying: *i*) the tracking system should be causal, frames are made available to the tracker sequentially one by one and tracking results should be provided for one frame before the next frame is presented, and *ii*) the computational demands of the tracker, per frame, should not increase with sequence length. Further, the proposed trackers will be evaluated and compared to the baseline tracker from which they originate. Thorough comparisons to other state of the art trackers are available through the VOT 2014 Challenge[1].

As the ideas of channel representations may not be generally known, a brief introduction is presented in Sect. 2. The general tracker framework and target model representation is presented in Sect. 3. These sections also serve the purpose of introducing the notation used. The main contributions of the paper are presented in Sections 4 and 5. In Sect. 6, the effect of using the proposed methods in the tracker is evaluated. Sect. 7 concludes the paper. A video illustrating the approach is available as supplementary material[2].

[1] *http://votchallenge.net/vot2014/*
[2] Also available at *http://users.isy.liu.se/cvl/ofjall/vot2014.mp4*



(a) (b)

Fig. 2. Illustration of a channel representation for orientation data. (a) the orientation data (density in red) stems from two modes with different variance and probability mass. The blue lines indicate the mode estimates as extracted from the channel representation. (b) the channel representation of the distribution of orientation data. The thin plots indicate the kernel functions (the channels) and the bold lines illustrate the corresponding channel coefficients as weighted kernel functions.

2 Channel Representations

This section provides a brief introduction to channel representations at an intuitive level, since these methods will be required for our proposed contributions in Sections 4 and 5. Readers unfamiliar with these methods are referred to more comprehensive descriptions in literature [6, 2, 3] for details.

2.1 Channel Encoding

Channel representations have been proposed in 2000 [6]. The idea is to encode image features (e.g. intensity, orientation) in a vector of soft quantization levels, the channels. An example is given in Fig. 2, where orientation values are encoded.

Readers familiar with population codes [10, 14], soft/averaged histograms [12], or Parzen estimators will find similarities. The major difference is that channel representations are very efficient to encode (because of the regular spacing of the channels) and decode (by applying frame theory [5]).

This computational efficiency allows for computing channel representations at each image pixel or for small image neighborhoods, as used in channel smoothing [2] as a variant of bilateral filtering [8], and tracking using distribution fields [4].

The kernel function, $b(\cdot)$, is defined to be non-negative, smooth and has compact support. In this paper, $\cos^2$ kernels with bandwidth parameter h are used:

$$b(\xi) = \frac{2}{3}\cos^2(\pi\xi/h) \qquad \text{for } |\xi| < h/2 \quad \text{and 0 otherwise.} \tag{1}$$

The components of the *channel vector* $\mathbf{x} = (x_1, x_2 \ldots, x_K)^T$ are obtained by shifting the kernel function K times with increments $h/3$. The range of the variable to be binned, ξ, together with the spacing of bins, v, determine the number

of required kernel functions $K = (\max(\xi) - \min(\xi))/v + 2$. In most cases $v \gg 1$ such that K is of moderate size. The smooth kernel of the channel representation reduces the quantization effect compared to histograms by a factor of up to 20 in practice [2]. This allows reduction of the computational load by using fewer bins or to increase the accuracy for the same number of bins.

2.2 Robust Decoding

Using channel decoding [5], the modes of a channel representation can be obtained. Decoding is not required for the operation of the tracker, however concepts from the decoding are required for presenting the proposed coherence measure. Decoding is used for visualization of the target model in the supplementary video. Since $\cos^2$-channels establish a tight frame, the local maximum is obtained using three orthogonal vectors [5] $\mathbf{w}_1 \propto (\ldots, 0, 2, -1, -1, 0, \ldots)^T, \mathbf{w}_2 \propto (\ldots, 0, 0, 1, -1, 0, \ldots)^T, \mathbf{w}_3 \propto (\ldots, 0, 1, 1, 1, 0, \ldots)^T$ and

$$r_1 \exp(i2\pi\hat{\xi}/h) = (\mathbf{w}_1 + i\mathbf{w}_2)^T\mathbf{x} \qquad r_2 = \mathbf{w}_3^T\mathbf{x} \tag{2}$$

where i denotes the imaginary unit, $\hat{\xi}$ is the estimate (modulo an integer shift determined by the position of the three non-zero elements in $\mathbf{w}_k$, the *decoding window*), and r_1, r_2 are two confidence measures. The decoding window is chosen to maximize r_2 when only one mode is decoded. In particular, when decoding a channel representation with only one encoded value ξ, it can be shown that $\hat{\xi} = \xi$ if ξ is within the representable range of the channel representation [5]. For a sequence of single encoded values, the channel vector traces out a third of a circle with radius r_1 within each decoding window, however, a comprehensive description of this geometrical interpretation is out of scope.

3 General Tracking Framework and Representation

The general tracker framework is not different from DFT [13] and EDFT [4] and is briefly presented here, further details are available in [4, 13]. In the first frame, the given bounding box of the object to be tracked is cut from the image. The intensity image patch of the target is channel encoded pixel wise using $K = 15$ channels, generating an I by J by K array denoted $\mathbf{C}$, where I and J are the height and width of the supplied bounding box. The 3D arrays generated from two channel encoded images are illustrated in Fig. 3.

In the next frame, the target representation $\mathbf{C}$ is compared to channel encoded patches (denoted $\mathbf{D}_{mn}$) from the new frame, where m and n represent a shift of the selected patch over the image. Gradient descent is used to find a minimum of a given comparison function, $d(\mathbf{C}, \mathbf{D}_{mn})$, with respect to the shift (m, n). Finally, the target representation is updated, $\mathbf{C} \leftarrow \mathbf{g}(\mathbf{C}, \mathbf{D}_{mn})$ and tracking continues in the next frame.

Prior to comparison, i.e. calculation of $d(\mathbf{C}, \mathbf{D}_{mn})$, the channel planes of $\mathbf{C}$ and $\mathbf{D}_{mn}$ are smoothed. This was shown to increase the size of the basin

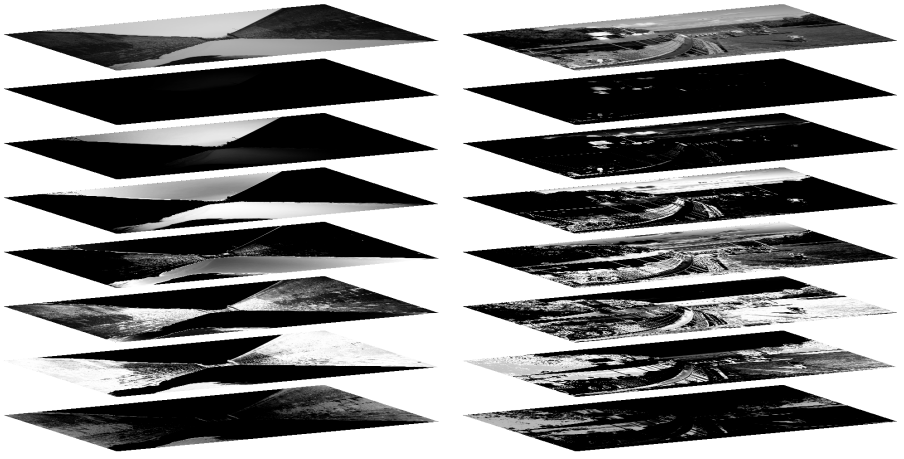


Fig. 3. Illustration of a pixel wise channel representation (with $K = 7$) of two images of canals. The top planes show the grayscale images while the lower seven planes indicate the activation of each channel (black: no activation, white: full activation). The lowest plane represents low image values (dark) while the seventh plane represents high image values (light).

of attraction for the correct solution [13]. Also, as in DFT and EDFT, a simple motion model (constant velocity model in the image plane) is used for initializing the gradient descent.

The main contribution of this work is a generalized model update function, $\mathbf{g}(\mathbf{C}, \mathbf{D}_{mn})$ and two proposals for the comparison function, $d(\mathbf{C}, \mathbf{D}_{mn})$. Earlier work has used a linearly weighted update, $\mathbf{g}(\mathbf{C}, \mathbf{D}_{mn}) = (1-\gamma)\mathbf{C} + \gamma\mathbf{D}_{mn}$, and the IJK dimensional L_1 norm for comparison. The function $\mathbf{g}$ is a 3D array valued function of two 3D arrays. In this work, multiplication of a 3D array with a scalar is taken to be multiplication of each element in the array with the scalar, similar to regular matrix-scalar multiplication. Further, $[\cdot]_{ijk}$ denotes the element at index i, j, k and $[\cdot]_{ij}$ denotes the channel vector (with K coefficients) corresponding to pixel i, j in the bounding box.

4 Target Model Comparison

As previously mentioned, previous work has used the L_1 norm extended to 3D arrays for comparison. However, as is visualized in Fig. 1, the target model representation contains (after a few frames) a representation of the distribution of values of each pixel within the bounding box. This should be exploited in the comparison function.

Since objects to be tracked are rarely rectangular, background pixels will be present in the bounding box. These pixels will generally vary more than the pixels on the object and such background pixels may disturb the tracker. This leads to a hypothesis that a weighted norm where the influence of inconsistent pixels is reduced, will improve the tracking results. Further, there may be areas of the tracked object which frequently change appearance, a weighted norm should also put more emphasis on parts of the tracked object showing more static appearance.

Two approaches will be presented. The first approach uses the reciprocals of the standard deviations of the represented distributions, which can be obtained directly from the channel coefficients. The second approach uses the *coherence*, which will be defined later.

4.1 Moments of Channel Representations

It can be shown that the average of several channel vectors, encoding values drawn from a specific distribution, tend to (up to scale) the probability density function convolved with the basis function, evaluated at channel centers[3](a sampled kernel density estimate) [5]. In this section, results based on a slightly different view of the distribution representation are presented. Here, the channel vector is assumed to represent a distribution, however, it is not necessarily the distribution from which a set of encoded values are drawn.

Let $b_k(\xi) \geq 0 \ \forall \ \xi$ be a set of regularly spaced channel basis functions normalized such that $\int_{-\infty}^{\infty} b_k(\xi)\,\mathrm{d}\xi = 1$, without loss of generality[4], and let $a_k \geq 0$ be the channel coefficients representing the distribution, $p(\xi)$, of a pixel,

$$p(\xi) = \sum_{k=1}^{K} a_k b_k(\xi) \qquad . \tag{3}$$

Let the coefficients be normalized such that $\sum_{k=1}^{K} a_k = 1$, from which $p(\xi) \geq 0 \ \forall \ \xi$ and $\int_{-\infty}^{\infty} p(\xi)\,\mathrm{d}\xi = 1$ follow. Let the random variable $X : P(X < z) = \int_{-\infty}^{z} p(\xi)\,\mathrm{d}\xi$, then expectations of functions $g(X)$ become scalar products with the channel coefficient vector since

$$E\left[g(X)\right] = \int_{-\infty}^{\infty} g(\xi)p(\xi)\,\mathrm{d}\xi = \sum_{k=1}^{K} a_k \int_{-\infty}^{\infty} g(\xi)b_k(\xi)\,\mathrm{d}\xi = \sum_{k=1}^{K} a_k g_{b_k} \tag{4}$$

with $g_{b_k} = \int_{-\infty}^{\infty} g(\xi)b_k(\xi)\,\mathrm{d}\xi$ (note: independent of the channel coefficients a_k).

Let μ and σ^2 denote the mean and variance of the represented distribution. For the mean, $g(X) = X$ and $\mu = E\left[X\right] = \sum_{k=1}^{K} a_k \mu_{b_k}$ with basis function

[3] Assuming symmetric channels and that the support of the density function is within the representable range of the channel representation.

[4] Conventionally, the basis functions and channel vectors are normalized differently, however, rescaling of the basis functions is compensated by a scaling factor and the channel vectors can be normalized beforehand.

means $\mu_{b_k} = \int_{-\infty}^{\infty} \xi b_k(\xi)\,\mathrm{d}\xi$, which for symmetric kernels coincide with channel centers. For the variance, $g(X) = (X-\mu)^2$ and

$$\begin{aligned}
\sigma^2 &= E\left[(X-\mu)^2\right] = \sum_{k=1}^{K} a_k \int_{-\infty}^{\infty} \left(\xi^2 - 2\mu\xi + \mu^2\right) b_k(\xi)\,\mathrm{d}\xi = \\
&= \sum_{k=1}^{K} a_k \left(\underbrace{\int_{-\infty}^{\infty} \xi^2 b_k(\xi)\,\mathrm{d}\xi}_{=\sigma_b^2+\mu_{b_k}^2} - 2\mu \underbrace{\int_{-\infty}^{\infty} \xi b_k(\xi)\,\mathrm{d}\xi}_{=\mu_{b_k}} + \mu^2 \underbrace{\int_{-\infty}^{\infty} b_k(\xi)\,\mathrm{d}\xi}_{=1} \right) = \\
&= \sigma_b^2 \underbrace{\sum_{k=1}^{K} a_k}_{=1} + \sum_{k=1}^{K} a_k \mu_{b_k}^2 - 2\mu \underbrace{\sum_{k=1}^{K} a_k \mu_{b_k}}_{=\mu} + \mu^2 \underbrace{\sum_{k=1}^{K} a_k}_{=1} = \\
&= \sigma_b^2 - \mu^2 + \sum_{k=1}^{K} a_k \mu_{b_k}^2
\end{aligned} \tag{5}$$

where $\sigma_b^2 = \int_{-\infty}^{\infty} (\xi - \mu_{b_k})^2 b_k(\xi)\,\mathrm{d}\xi \;\forall k$. Hence the mean and variance (and thus the standard deviation) of a channel represented distribution can be obtained through scalar products of channel coefficients and weight vectors. Further, these weight vectors only depend on the chosen channel basis functions and can be calculated in advance. The weighted comparison function thus is

$$d(\mathbf{C},\ \mathbf{D}) = \sum_{i,j,k} \frac{1}{\sigma_{ij}} \left| [\mathbf{C}]_{ijk} - [\mathbf{D}]_{ijk} \right| \tag{6}$$

where each σ_{ij} is the estimated standard deviation of each channel vector $[\mathbf{C}]_{ij}$ in the target model. The sum is over all pixels in the bounding box and all channel coefficients.

4.2 Coherence

For combinations of multiple channel encoded measurements of an entity, two properties characterizing the combined channel vector are of interest. Here we refer to them as *evidence* and *coherence*.

Evidence is what is referred to as r_2 in Sect. 2.2, the L_1 norm of the decoding window. When combining channel vectors by addition, r_2 is proportional to the number of samples accumulated within the current decoding window.

Coherence, which we define as r_1^2/r_2^2, is a measure of the consistency of samples resulting in a mode, see Fig. 2, where the right mode has higher coherence than the left mode. Coherence as just defined is a property related to a specific decoding window, and we define the coherence of a full channel vector as the coherence of the strongest mode. The strongest mode is defined as the decoding window with largest evidence [5].

Motivation of the Definition of Coherence Several norms and properties of channel encoded entities have been proposed and evaluated in the literature [7, 5], however, coherence has not previously been suggested, although it has been suggested for the structure tensor [1]. For notational and conceptual clarity and without loss of generality, basis functions are assumed to be centered at integer positions in this section ($h = 3$).

As shown in [5] and indicated in Sect. 2.2, decoding of $\cos^2$ channel vectors (determining estimates of ξ) are carried out according to

$$\begin{pmatrix} r_1 \cos(\frac{2\pi}{3}(\xi - l)) \\ r_1 \sin(\frac{2\pi}{3}(\xi - l)) \\ r_2 \end{pmatrix} = \begin{pmatrix} 2 & -1 & -1 \\ 0 & \sqrt{3} & -\sqrt{3} \\ 1 & 1 & 1 \end{pmatrix} \mathbf{x}_l \tag{7}$$

where l selects the decoding window and $\mathbf{x}_l$ is the corresponding three elements from the channel vector $\mathbf{x}$ to be decoded. It follows that when all elements in $\mathbf{x}_l$ are equal, $r_1 = 0$ and decoding is ambiguous. When the values within the decoding window are such that r_1 is large, the estimate of ξ is less dependent on small perturbations of the channel coefficients, however, the absolute value of r_1 varies with the scaling of the channel coefficients.

The proposed coherence measure, $\mathrm{coh}(\cdot)$, can be expressed as

$$\mathrm{coh}(\mathbf{x}_l) = \frac{r_1^2}{r_2^2} = \frac{1}{\mathbf{1}^{\mathrm{T}} \mathbf{x}_l \mathbf{x}_l^{\mathrm{T}} \mathbf{1}} \, \mathbf{x}_l^{\mathrm{T}} \begin{pmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{pmatrix} \mathbf{x}_l \tag{8}$$

with $\mathbf{1} = (1\ 1\ 1)^{\mathrm{T}}$ and where the last equality follows from (7). It can easily be verified that $\mathrm{coh}(\mathbf{x}_l) = 0$ when decoding is ambiguous and $\mathrm{coh}(\mathbf{x}_l) = 1$ for a single encoded value or for a combination of encodings of the same value. Further, $\mathrm{coh}(\alpha \mathbf{x}_l)$ is independent of scale ($\alpha > 0$) and, $\mathrm{coh}(\mathbf{x}_l)$ decreases monotonically with a wider distribution of the encoded values within the decoding window. These results build upon properties of the $\cos^2$ kernel, namely that for any value ξ within the representable range of a channel representation, the L_1 and L_2 norms of the corresponding channel vector are constant [5] (and specifically independent of the position of ξ with respect to channel centers). These properties do not hold for Gaussian or B-spline kernels.

Using coherence weighting gives a proposed comparison function

$$d(\mathbf{C},\ \mathbf{D}) = \sum_{i,j,k} |[\mathbf{C}]_{ijk} - [\mathbf{D}]_{ijk}| \ \ (\mathrm{coh}([\mathbf{C}]_{ij}) + \kappa) \tag{9}$$

where $[\cdot]_{ijk}$ denotes the element at index i, j, k and $[\cdot]_{ij}$ denotes the channel vector (with K coefficients) corresponding to pixel i, j in the bounding box. As defined earlier, the coherence of a full channel vector is the coherence of the decoding window corresponding to the strongest mode. $\kappa \geq 0$ is a parameter representing the trust level of the coherence estimate. The sum is over all pixels in the bounding box and all channel coefficients.

For single-mode distributions, the coherence is inversely related to the variance of the distribution where wide distributions generate low coherence and

vice versa. However, for multi-modal distributions, variance is generally large as it is a global property of the distribution. On the contrary, coherence may still be large (corresponding to low variance of the strongest mode) as it is a local property of the individual mode.

5 Target Model q-Update

In this section, the proposed target model update is presented. Here, a discrete time index t is used such that $\mathbf{C}_t$ is the target model obtained after applying all updates up to and including frame t. Similarly, $\mathbf{D}_t$ is the encoded bounding box found in frame t (the best match found by the tracking framework, hence removing the translation subscripts m, n from Sect. 3).

Previous approaches to channel (or distribution field) tracking have used a linear convex combination update

$$\mathbf{C}_t = (1-\gamma)\mathbf{C}_{t-1} + \gamma\mathbf{D}_t \tag{10}$$

with a learning rate parameter $0 < \gamma < 1$. This parameter also determines the forgetting factor $(1-\gamma)$. This update rule is also applicable to image based target representations, where $\mathbf{C}_t$ becomes a weighted mean of the target found in the last frames. However, for the channel based target representation, non-linear update rules are allowed as the update operates on channel coefficients and not directly on intensity values. We propose a power update rule

$$\mathbf{C}_t = \left((1-\gamma)\mathbf{C}_{t-1}^q + \gamma\mathbf{D}_t^q\right)^{\frac{1}{q}} \tag{11}$$

where array exponentiation is to be taken element-wise. The power function is strictly monotonic for positive arguments and thus the order of the channel coefficients is not affected. This bears some resemblance to α-divergences of distributions [11], however, the use is different.

All coefficients in $\mathbf{D}_t$ are non-negative and bounded by the maximum channel activation, $\max_\xi b(\xi)$. Also, from (11) follows that $\mathbf{C}_t \leq \max(\mathbf{C}_{t-1}, \mathbf{D}_t)$ (element wise), ensuring that all elements will remain bounded.

Increasing q shifts the weight towards the larger of each two corresponding elements in $\mathbf{C}_{t-1}$ and $\mathbf{D}_t$. If $[\mathbf{D}_t]_{ijk} > [\mathbf{C}_{t-1}]_{ijk}$, i.e. the current training sample is dominating, increased q leads to faster adaptation to new information. On the other hand, if $[\mathbf{D}_t]_{ijk} < [\mathbf{C}_{t-1}]_{ijk}$, increased q leads to slower forgetting. Increasing γ on the other hand, leads to faster learning and faster forgetting. Using both q and γ, learning rate and forgetting rate can be set independently. Letting $q \to \infty$, (11) becomes $\mathbf{C}_t = \max(\mathbf{C}_{t-1}, \mathbf{D}_t)$, i.e. learning is immediate and the model never forgets. The linear update is a special case ($q = 1$).

Note that for $\mathbf{C}_t$ to become true sampled kernel density estimates of the pixel values, a few more conditions have to be fulfilled in addition to the normalization requirements previously mentioned. In particular, a time dependent learning rate $\gamma = 1/t$ (ensuring equally weighted samples) and $q = 1$ is required. Using a fixed learning rate, more emphasis is given to more recent samples, which usually is beneficial in practice however, with processes not approximately stationary over longer time periods.

6 Experiments

Trackers enhanced with the proposed update scheme and the weighted comparison functions are implemented (in MATLAB) and compared to previous trackers on the VOT2014 challenge benchmark, according to the rules of, and using the evaluation framework provided by the challenge [9].

In the following, DFT and EDFT refer to the previously published trackers by Sevilla et al. [13], and Felsberg [4], respectively. NCC is an example normalized cross correlation tracker distributed with the evaluation framework. Trackers using the proposed q-update scheme are prefixed with a q, and followed by an indication of the value of q. Infinite q is denoted by *max*, a special case as the q-update tend to the max-operation for increasing q. Trackers using the proposed coherence weighted comparison are prefixed with a w and trackers using the proposed standard deviation weighted comparison are prefixed with $w\sigma$. Unmarked trackers use the L_1 norm comparison. For coherence weighting, the parameter κ was set to 2. For all trackers, learning rate γ is set to 0.05 and 15 channels are used.

Three performance measures are available, these are briefly presented here. For the comprehensive version, we refer to [9]. *Accuracy* is the ratio of the joint area of tracker output and ground truth and the union of the two, averaged over each sequence (larger is better). *Robustness* is the reset count, the evaluated tracker is reset as soon as there is no overlap between tracker output and ground truth (smaller is better). *Speed* is the average framerate of the tracker (larger is better).

Two experiments are performed. In the first, denoted *baseline*, each tracker is initialized using the ground truth bounding box of the first frame. In the second experiment, *region noise*, each tracker is initialized with the ground truth bounding box with a random offset. In the second experiment each sequence is evaluated 15 times with different offsets and the mean is reported by the VOT evaluation framework. The results for the baseline experiment are presented in table 1, and the results for the region noise experiment are presented in table 2. For each tracker, both the average and median score over all sequences are presented.

For the baseline experiment (table 1), all channel-based trackers outperform the tracker based on normalized cross correlation (NCC) in accuracy and robustness. For evaluation of the proposed extensions, the trackers using these will be compared to the baseline channel-based tracker (EDFT). Introducing the non-linear update (qEDFT) increases accuracy and slightly increases robustness (decreasing failure rate) for increasing q up to $q = 5$. For $q = 6$, performance decreases slightly. Only using the proposed weighted comparison (wEDFT), robustness decreases slightly while accuracy stays similar to EDFT.

The best performance is achieved by combining the non-linear update with the weighted comparison. Using non-linear update and coherence weighted comparison (qwEDFT with $q = 4$), mean accuracy increases more than 5% and mean robustness is 15% better than EDFT. For larger q, accuracy increases further while the robustness degrades. The corresponding standard deviation weighted

Table 1. Summarized results for the baseline experiment, comparison to competing methods (best scores in boldface).

	Mean			Median		
Method	Accuracy	Robustness	Speed	Accuracy	Robustness	Speed
NCC	0.467	2.960	**14.8**	0.423	2.0	**11.5**
DFT	0.531	2.200	6.3	0.534	2.0	6.9
EDFT	0.521	1.840	10.0	0.528	2.0	10.8
qEDFT (q=2)	0.525	2.000	10.6	0.534	2.0	10.8
qEDFT (q=3)	0.536	1.720	7.0	0.541	**1.0**	6.8
qEDFT (q=4)	0.547	1.720	7.1	0.553	**1.0**	7.0
qEDFT (q=5)	0.552	1.720	7.2	0.560	**1.0**	7.1
qEDFT (q=6)	0.540	1.920	6.6	0.553	**1.0**	6.4
wEDFT	0.523	2.040	6.8	0.536	2.0	6.5
qwEDFT (q=2)	0.544	1.560	7.1	0.535	**1.0**	7.1
qwEDFT (q=3)	0.547	1.600	5.5	0.539	**1.0**	5.4
qwEDFT (q=4)	0.550	1.560	5.6	**0.565**	**1.0**	5.5
qwEDFT (q=5)	0.554	1.640	5.1	0.561	**1.0**	5.2
qwEDFT (q=6)	**0.558**	1.920	5.4	0.561	**1.0**	5.4
qwEDFT (q=7)	**0.558**	1.640	5.4	0.560	**1.0**	5.5
maxwEDFT	0.545	1.960	6.4	0.538	2.0	6.0
qwσEDFT (q=2)	0.522	1.400	8.8	0.534	**1.0**	9.1
qwσEDFT (q=3)	0.522	1.440	6.7	0.533	**1.0**	7.0
qwσEDFT (q=4)	0.540	**1.360**	6.2	0.560	**1.0**	6.2
qwσEDFT (q=5)	0.545	1.520	6.5	0.549	**1.0**	6.6
qwσEDFT (q=6)	0.541	1.480	6.8	0.558	**1.0**	7.1
qwσEDFT (q=7)	0.545	1.600	6.9	0.555	**1.0**	7.2
maxwσEDFT	0.547	1.960	7.3	0.549	2.0	7.0

methods perform slightly inferior to the best methods (the coherence weighted) in terms of mean accuracy. However, the best robustness is achieved by a standard deviation weighted method (qwσEDFT with $q = 4$). In general, accuracy seem to improve with larger q while the best robustness is achieved for q close to 4. For median accuracy, $q = 4$ gives the best performance for both coherence weighted trackers and standard deviation weighted trackers, with better results for coherence weighting.

For the region noise experiments (table 2), accuracy generally increases with increasing q while the best robustness is achieved for $q = 4$ for the standard deviation weighted tracker and for $q = 6$ for the coherence weighted tracker. Contrary to the baseline experiments, in the region noise experiments the coherence weighted methods perform best with respect to robustness while the standard deviation weighted methods perform best with respect to accuracy.

In table 3, the results for each sequence for three trackers are presented. A comprehensive description of the sequences themselves is available at the VOT challenge site[5]. Both proposed trackers outperform the EDFT tracker with re-

[5] *http://votchallenge.net/vot2014/dataset.html*

Table 2. Summarized results for the region noise experiment, comparison to competing methods (best scores in boldface).

	Mean			Median		
Method	Accuracy	Robustness	Speed	Accuracy	Robustness	Speed
NCC	0.456	2.973	**14.0**	0.414	1.8	**12.0**
DFT	0.493	2.389	6.0	0.512	2.4	5.9
EDFT	0.486	1.973	10.1	0.486	1.9	10.5
qEDFT (q=2)	0.492	2.059	10.3	0.488	1.9	10.8
qEDFT (q=3)	0.497	2.000	6.9	0.518	1.8	6.8
qEDFT (q=4)	0.498	2.032	6.7	0.492	1.7	6.6
qEDFT (q=5)	0.502	2.008	6.7	0.512	**1.3**	6.7
qEDFT (q=6)	0.499	2.093	6.4	0.521	1.6	6.5
wEDFT	0.489	2.088	6.2	0.492	1.9	6.3
qwEDFT (q=2)	0.501	1.835	6.7	0.500	1.9	6.8
qwEDFT (q=3)	0.508	1.819	5.4	0.520	1.6	5.4
qwEDFT (q=4)	0.509	**1.747**	5.1	0.502	1.5	4.9
qwEDFT (q=5)	0.515	1.819	5.2	0.499	1.5	5.1
qwEDFT (q=6)	0.516	1.837	5.2	0.530	1.5	5.2
qwEDFT (q=7)	0.514	1.923	5.1	0.515	1.5	5.1
maxwEDFT	0.514	2.163	6.3	0.500	2.0	6.2
qwσEDFT (q=2)	0.500	2.029	8.6	0.520	1.5	8.8
qwσEDFT (q=3)	0.502	1.832	6.6	0.510	1.5	6.6
qwσEDFT (q=4)	**0.521**	1.893	7.1	**0.534**	1.6	6.7
qwσEDFT (q=5)	0.506	1.803	6.5	0.515	1.7	6.4
qwσEDFT (q=6)	0.510	1.787	6.8	0.517	1.7	6.7
qwσEDFT (q=7)	0.511	1.795	6.6	0.517	1.8	6.4
maxwσEDFT	0.516	2.109	7.9	0.529	2.0	7.8

spect to accuracy on 15 out of 25 sequences. On four sequences the EDFT tracker outperforms the proposed trackers and in three cases, performance is equal among the three trackers. With respect to robustness, all three trackers perform equal on 18 out of 25 sequences. The improvement compared to EDFT with respect to robustness is largest on the sequences where EDFT performs worst. On the *hand2* sequence, EDFT looses track of the object seven times while the proposed qwσEDFT tracker looses track of the object three times.

No parameters have been changed from those used in the baseline trackers, with the exception of the newly introduced parameter q. Since q and the learning rate γ together determine the effective learning and forgetting rates of the final tracker, a further increase in performance should be possible by jointly optimizing these parameters. Also, by avoiding recomputation, primarily of weights in the search phase, an increase in framerate should be possible. Currently the proposed extensions slows down the tracker to the level of the DFT tracker. Implementing the trackers in C++ should allow video rates on the sequences.

As a final remark, a selection of different comparison functions were evaluated such as L_2, variance weighed L_2 and Hellinger distance. However, these performed inferior to the weighted L_1 norms.

Table 3. Detailed *baseline* experiment results for three trackers. Best scores in bold.

	EDFT		qwσEDFT (q=4)		qwEDFT (q=4)	
	accuracy	**robustness**	**accuracy**	**robustness**	**accuracy**	**robustness**
ball	0.51	0	0.57	0	**0.59**	0
basketball	0.56	3	0.57	**1**	**0.59**	3
bicycle	**0.44**	0	0.43	0	0.43	0
bolt	0.51	3	0.56	3	0.56	3
car	0.53	1	0.53	1	0.53	1
david	0.68	0	0.71	0	**0.72**	0
diving	0.16	3	0.16	3	0.16	3
drunk	**0.51**	0	0.49	0	0.50	0
fernando	0.40	2	0.40	2	**0.43**	2
fish1	0.38	4	0.40	4	**0.42**	4
fish2	0.28	6	0.30	**5**	**0.32**	6
gymnastics	**0.55**	2	0.54	2	0.53	2
hand1	0.59	2	0.56	0	**0.60**	0
hand2	0.42	7	0.44	**3**	0.44	6
jogging	0.79	2	0.80	2	0.80	2
motocross	0.18	3	**0.23**	4	0.20	3
polarbear	0.53	0	0.58	0	**0.59**	0
skating	0.61	1	0.61	1	**0.62**	1
sphere	0.62	1	0.69	0	**0.71**	0
sunshade	0.65	3	0.70	1	**0.71**	1
surfing	0.85	0	0.89	0	**0.90**	0
torus	**0.82**	0	0.77	0	0.80	0
trellis	0.51	2	0.52	1	**0.56**	1
tunnel	0.31	0	0.31	0	0.31	0
woman	0.61	1	**0.72**	1	0.69	1

7 Conclusion

In the present work, we have addressed two significant parts of a tracking system, comparison and model update. We have proposed a generalized update rule (q-update) and two weighted comparison functions (coherence weighted and reciprocal standard deviation weighted). The proposals aim to exploit the distribution representation of the target model. On the VOT challenge benchmark, trackers extended with these proposals showed significant increase in tracking performance. We thus conclude that the proposed methods better utilize the possibilities of the model representation since these proposed methods rely on properties of the channel representation that do not hold for image representations or mean/variance (Gaussian approximation) representations.

Acknowledgements. This work has been supported by SSF through a grant for the project CUAS, by VR through a grant for the project ETT, through the Strategic Area for ICT research CADICS, and ELLIIT.

References

1. Bigün, J., Granlund, G.H.: Optimal orientation detection of linear symmetry. In: Proceedings of the IEEE First International Conference on Computer Vision. pp. 433–438. London, Great Britain (June 1987)
2. Felsberg, M., Forssén, P.E., Scharr, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(2), 209–222 (2006)
3. Felsberg, M., Larsson, F., Wiklund, J., Wadströmer, N., Ahlberg, J.: Online learning of correspondences between images. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
4. Felsberg, M.: Enhanced distribution field tracking using channel representations. In: IEEE ICCV workshop on visual object tracking challenge (2013)
5. Forssén, P.E.: Low and Medium Level Vision using Channel Representations. Ph.D. thesis, Linköping University, Sweden (2004)
6. Granlund, G.H.: An Associative Perception-Action Structure Using a Localized Space Variant Information Representation. In: Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC). Germany (September 2000)
7. Johansson, B., Elfving, T., Kozlov, V., Censor, Y., Forssén, P.E., Granlund, G.: The application of an oblique-projected landweber method to a model of supervised learning. Mathematical and Computer Modelling 43, 892–909 (April 2006)
8. Kass, M., Solomon, J.: Smoothed local histogram filters. In: ACM SIGGRAPH 2010 papers. pp. 100:1–100:10. SIGGRAPH '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1833349.1778837
9. Kristan, M., Čehovin, L., Vojir, T., Nebehay, G.: Visual object tracking challenge 2014 evaluation kit. http://votchallenge.net/vot2014/download/vot2014-guidelines.pdf
10. Pouget, A., Dayan, P., Zemel, R.S.: Inference and computation with population codes. Annu. Rev. Neurosci. 26, 381–410 (2003)
11. Rényi, A.: On measures of entropy and information. In: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability. pp. 547–561. University of California Press, Berkeley, Calif. (1961)
12. Scott, D.W.: Averaged shifted histograms: Effective nonparametric density estimators in several dimensions. Annals of Statistics 13(3), 1024–1040 (1985)
13. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: IEEE Computer Vision and Pattern Recognition (2012)
14. Zemel, R.S., Dayan, P., Pouget, A.: Probabilistic interpretation of population codes. Neural Computation 10(2), 403–430 (1998)