Enhanced Distribution Field Tracking using Channel Representations

Michael Felsberg
Linköping University
58183 Linköping, Sweden
michael.felsberg@liu.se

Abstract

Visual tracking of objects under varying lighting conditions and changes of the object appearance, such as articulation and change of aspect, is a challenging problem. Due to its robustness and speed, distribution field tracking is among the state-of-the-art approaches for tracking objects with constant size in grayscale sequences. According to the theory of averaged shifted histograms, distribution fields are an approximation of kernel density estimates. Another, more efficient approximation are channel representations, which are used in the present paper to derive an enhanced computational scheme for tracking. This enhanced distribution field tracking method outperforms several state-of-the-art methods on the VOT2013 challenge, which evaluates accuracy, robustness, and speed.

1. Introduction

This paper addresses the problem of visual object tracking, *i.e.* local methods of searching for a matching object in a grayscale image, which is a fundamental step in many computer vision systems, *e.g.* visual surveillance and structure from motion.

The visual matching requires a suitable representation of the object appearance and the current local image patch. In [20] the authors propose to compare Distribution Fields (DFs) of the object and the local image region. The resulting DF Tracking (DFT) method outperforms several state-of-the-art methods.

The particular goal of the present work is to improve the estimation process of the DFs and the local matching scheme, in order to improve the overall performance. For this purpose, we extend DFT in three ways:

A) By careful analysis of the DFT implementation, we improve the *search window selection* and the *motion prediction computation*.

B) Using the theory of *Averaged Shifted Histograms* (ASH) [19], we replace the smoothed histograms with a coarse grid of smooth bins as used in Channel Representations (CRs) [10].

C) For the channel-based method, we systematically derive the *parameters* that are equivalent to the optimized parameters of the DFT.

The resulting variants are step-wise compared to the original DFT algorithm, using the challenging VOT2013 evaluation kit [1]. For each change of the original algorithm, the performance improves. The so proposed final version of the algorithm, *Enhanced Distribution Field Tracking* (EDFT), is faster than the original method and is more robust against outliers, while achieving at least the same accuracy. EDFT compares also favorably to other state-of-the-art methods on the VOT2013 challenge.

1.1. Related Work

Visual object tracking is based on various aspects including appearance representation, motion modeling, object modeling, and update rules for all previous aspects. In the present work we focus on the appearance representation, which can be in terms of a template or kernel-based. Hybrid approaches become increasingly common, though.

Template-based approaches make use of intensity values, color values, gradient information, or other simple features in a spatial grid [3]. These methods suffers typically from outliers, which may be addressed using robust metrics [15], and non-smooth objective functions, which may be addressed by smoothing [14].

Kernel-based approaches, in the simplest case histogram-based methods, integrate information over the image patch [5]. These methods are often more robust than template-based ones, but suffer from ambiguities caused by the loss of spatial structure [11]. This is mitigated by the use of spatially varying kernels or higher order statistics [4], thus going toward hybrid methods.

Combining spatial structure and kernel-based estimates in a hybrid approach allows to balance the trade-off between specificity of the template-based approach and the sensitivity of the kernel-based approach. Indeed, the resolution of the spatial grid and the bandwidth of the kernel are

strongly coupled and can be recomputed on the fly [6]. In Distribution Field Tracking (DFT) [20], a method based on smoothed local histograms, it has been shown that a careful choice of parameters lead to state-of-the-art tracking results. The computation of local kernel-based features becomes very efficient if using integral images, and real-time performance can be achieved [16].

Smoothed local histograms, or Averaged Shifted Histograms (ASH), are closely related to kernel density estimators [19] and smooth histograms, in particular Channel Representations (CRs) [10]. Due to the frame-properties of the latter, transformations of the template (rotation, scaling) can be directly mapped to localized CRs, called Channel-Coded Feature Maps, which can then be used for tracking under affine transformations [12] – however at a larger computational effort. Similar to frame-based features, wavelet features such as Haar features have been used successfully, however not in combination with simple distance measures, but using boosting techniques [2].

1.2. Contributions

In the present work we concentrate on fast hybrid approaches for tracking using a fixed size window. In particular, we extend Distribution Field Tracking (DFT) in three ways:

- By careful analysis of the DFT implementation, we improve the search window selection and the motion prediction computation.
- Using the theory of Averaged Shifted Histograms (ASH) [19], we replace the smoothed histograms with smooth bins as used in Channel Representations (CRs) [10].
- For the channel-based method, we systematically derive the parameters that are equivalent to the optimized parameters of the DFT.

The resulting method, Enhanced DFT (EDFT), is evaluated on the VOT2013 Challenge Dataset [1] and is compared to state-of-the-art methods.

2. Methods

In order to make the paper self-contained, we add short technical descriptions of the DFT method (section 2.1), ASH (section 2.2), and CRs (section 2.3). Our contributions and the EDFT method are explained in detail in section 2.4.

2.1. Distribution Field Tracking

Distribution Field Tracking (DFT) [20] is a visual region tracking method that is based on comparing smoothed local histograms of the image patch. Histograms are one of the most simple forms of non-parametric density representation. In case of DFT, the image value (grayscale) is the stochastic variable and its distribution is estimated in three steps: a) quantizing and binning the value domain; b) spatial smoothing at different scales; and c) smoothing the bins. Step a) results in a one-out-of b coding $d(i,j,k)$ $(k = 1,\dots,b)$ of the image $I(i,j)$

$$d(i,j,k) = \begin{cases} 1 & \text{if } I(i,j) == k \\ 0 & \text{otherwise} \end{cases} . \quad (1)$$

In the original work it is suggested to use $b = 16$ quantization levels.

The spatial smoothing in step b) makes use of 2D Gaussian kernels $h_{\sigma_s}(i,j)$ at two scales $\sigma_s = 1$ and $\sigma_s = 2$

$$d_s(i,j,k) = (d(\cdot,\cdot,k) * h_{\sigma_s})(i,j) \qquad \text{for all } i,j. \quad (2)$$

The smoothing in step c) makes use of a 1D Gaussian kernel $h_{\sigma_f}(k)$ with $\sigma_f = 10$ (with respect to 255 grayscales, thus $\sigma_f = 0.625$ if $b = 16$)

$$d_{ss}(i,j,k) = (d_s(i,j,\cdot) * h_{\sigma_f})(k) \qquad \text{for } k = 1,\dots,b. \quad (3)$$

The so computed function $d_{ss}(i,j,k)$ is denoted as distribution field (DF) in the sequel and its subscript is omitted for simplifying the notation.

During the tracking, the DF of the template, d_{model}, is compared to the DF of a local window in the current frame, d_f, within a local search and a coarse-to-fine strategy. The distance measure used is the sum of absolute differences, *i.e.* the L_1 distance between d_{model} and d_f

$$L_1(d_{\text{model}}, d_f) = \sum_{i,j,k} |d_{\text{model}}(i,j,k) - d_f(i,j,k)| \ . \quad (4)$$

The displacement is estimated coarse-to-fine by local search of the minimum L_1 error within a window of maximum displacement (30 pixels in the original work). When the best-fitting position has been found, the current template $d_{\text{model},t}$ is updated using the current DF d_f using linear weights $\lambda = 0.95$ for the previous template and $(1-\lambda) = 0.05$ for the update

$$d_{\text{model},t+1}(i,j,k) = \lambda d_{\text{model},t}(i,j,k) + (1-\lambda) d_f(i,j,k). \quad (5)$$

Due to the density-based comparison, the method is robust against outliers, and due to the template-update, the method can also deal with continuous changes of object aspects and the lighting. All parameters have been optimized using cross-validation on a dataset with 11 sequences [20], which shares 2 sequences with the VOT2013 dataset (david and face).

2.2. Averaged Shifted Histograms

The construction of DFs combines pooling steps and weighted averaging. The exact statistical characterization of the final descriptor is thus not straightforward, but falls within the theory of Averaged Shifted Histograms (ASH) [19]. At each spatial position, the result from (2) in step b) is a (weighted) histogram. Thus, the bin smoothing in (3) results in an averaging of histograms.

According to ASH theory [19], asymptotic properties of density estimates by averaged histograms are superior to ordinary histograms. To start with, a set of m shifted histograms[1] $\hat{f}_1(x), \ldots, \hat{f}_m(x)$ with bin-width h is generated, such that the relative shift is h/m. For the unweighted ASH, the m histograms are averaged in each point

$$\hat{f}_{\mathrm{ASH}}(x;m) = \frac{1}{m} \sum_{i=1}^{m} \hat{f}_i(x) \quad . \tag{6}$$

The ASH is piece-wise constant in intervals of width h/m. If we now assume a fine histogram $\hat{g}(x)$ on these intervals, we may calculate the ASH from the latter as [18]

$$\hat{f}_{\mathrm{ASH}}(x;m) = \frac{1}{m} \sum_{i=1-m}^{m-1} \left(1 - \frac{|i|}{m}\right) \hat{g}(x+i) \quad . \tag{7}$$

Or to put it into signal processing terms, the histogram values $\hat{f}_i$ are obtained by convolving $\hat{g}$ with a rectangular kernel of width m, and the ASH is obtained by convolving $\hat{g}$ with a triangle-kernel of width $2m-1$ (which is obtained by convolving the rectangular kernel with itself).

For the weighted ASH, the isosceles triangle in (7) is replaced with a more general weighting function $w_m(i) \geq 0$ that sums to one

$$\hat{f}_{\mathrm{ASH}}(x;m) = \sum_{i=1-m}^{m-1} w_m(i) \hat{g}(x+i) \quad . \tag{8}$$

Compared to ordinary histograms with comparable computational effort, the ASH has a lower Asymptotic Mean Integrated Squared Error (AMISE) [18], pp. 119–121. If w_m is chosen as a Gaussian kernel, the weighted ASH is identical to the DF feature pooling (3). On the other hand, taking the limit of infinitesimal narrow bins, we obtain a kernel density estimator (KDE) if we place the kernel at the n data samples x_j, $j \in \{1, \ldots, n\}$, see (5.14) in [18]

$$\lim_{m\to\infty} \hat{f}_{\mathrm{ASH}}(x;m) = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{x - x_j}{h}\right) \quad , \tag{9}$$

where $K(\cdot/h)$ is obtained from $\lim_{m\to\infty} w_m(\cdot)$. Thus, in expectation values sense, DFs become kernel density estimators.

[1] In our text we throughout identify histograms and the corresponding density estimate since the latter is defined as the relative count divided by the bin width.

2.3. Channel Representations

Another efficient method to approximate kernel density estimators in expectation sense are Channel Representations (CRs) [10], also called population codings [17], a biologically inspired data representation. Regular CRs have a probabilistic interpretation in terms of smooth histograms [22, 8], *i.e.*, the kernel functions used for binning the data are smooth instead of rectangle functions. Possible choices are (besides Gaussian kernels) $\cos^2$ kernels or quadratic B-splines [9]. An easily accessible introduction to the topic is given in [7].

If we assume bin centers with spacing h, the coefficients of the CR are computed as

$$c_k = \frac{1}{nh} \sum_{j=1}^{n} K(x_j/h - k) \qquad k \in \mathbb{N} \quad . \tag{10}$$

Thus, CRs are similar to KDEs, but establish a discrete function over the bin-centers (like histograms) instead of a continuous function. Therefore, the sum above is evaluated only once when adding the sample x_j, in contrast to KDEs, where the sum over all x_j has to be evaluated for each x. Thus, CRs are more efficient to compute if the number of samples is large and the dimensionality is moderate.

CRs are also a limit case $m \to \infty$ of ASHs, but instead of placing the kernel functions at the samples, they are placed at the coarse grid of the ASH. From sampling theory it is known that lowpass-filtered signals can be subsampled without loss of information. In this case, the kernel function K acts as a lowpass-filter on the underlying distribution (a regularizer) and therefore the coarser grid of the ASH is sufficient to represent the density with high accuracy. In expectation sense, the coefficients in the channel representation are equivalent to the kernel density estimator evaluated at discrete points [8] and thus the limit of the ASH

$$E\{c_k\} = \lim_{m\to\infty} \hat{f}_{\mathrm{ASH}}(k;m) \quad . \tag{11}$$

The major practical question in this context is whether CRs or DFs result in a better, *i.e.* faster and more accurate, approximation of a kernel density estimator. Or in other words: is it better to smooth ordinary histograms or to compute histograms with smooth bins? This question will be addressed below and for this purpose we will calculate the CR parameters that are equivalent to the DF parameters as described above. For this calculation, we need the functional description of the applied kernel function. We chose a quadratic B-spline over $\cos^2$ kernels in order to obtain simpler calculations below:

$$K(x) = \begin{cases} 3/4 - x^2 & |x| \leq 1/2 \\ (|x| - 3/2)^2/2 & 1/2 < |x| \leq 3/2 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{12}$$

2.4. Enhance Distribution Field Tracking

This section contains the main contributions of the present paper and successively leads to the proposed Enhanced Distribution Field Tracking (EDFT) algorithm. The EDFT algorithm is obtained from the DFT method in three steps, cf. figure 1.

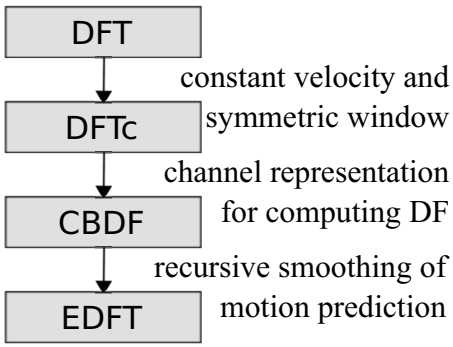

Figure 1. Overview of proposed changes and the resulting new variants of the DFT algorithm.

DFTc In a first step, we improve some details of the DFT algorithm, resulting in the DFT with constant velocity prediction (DFTc). The original code from the authors extracts a slightly asymmetric window. This has been corrected in the DFTc.

For the initialization of the minimization, the original algorithm uses a prediction based on the correction of the previous prediction. Instead, the DFTc starts at the position predicted by a constant motion model. If the previous estimated object position (upper left corner of the bounding box) is denoted $\mathbf{p}_{\text{old}}$ and the current one as $\mathbf{p}_{\text{new}}$, the motion prediction $\mathbf{m}_p$ is given as

$$\mathbf{m}_p = \mathbf{p}_{\text{new}} - \mathbf{p}_{\text{old}} \tag{13}$$

so that the predicted position $\mathbf{p}_p$ becomes

$$\mathbf{p}_p = \mathbf{p}_{\text{new}} + \mathbf{m}_p = 2\mathbf{p}_{\text{new}} - \mathbf{p}_{\text{old}} \ . \tag{14}$$

CBDF In a second step, we replace the explicit histogram averaging in the DF feature pooling (3) with the encoding into the equivalent CR (10). The implementation of the CR is based on the Matlab toolbox (GPL) that is based on [13]. The channel-based DFT is denoted as CBDF.

In order to get the CBDF that is equivalent to the DFT in expectation sense, we choose the bandwidth parameter h of the CR from the parameters of the DF. Since we consider only one parameter, we obtain the best approximation by choosing the same effective variance of the combined kernel. For the DF, this is the Gaussian kernel h_{σ_f} ($\sigma_f = 10$) convolved with the original bins used in (1), a rectangle of width $w = 16$. Thus we obtain

$$\sigma_{\text{eff}} = \sqrt{\int_{t=-w/2}^{w/2} \frac{1}{w}(\sigma_f^2 + t^2)\, dt} = \sqrt{\sigma_f^2 + \frac{w^2}{12}} \ . \tag{15}$$

On the other hand, the variance of $h^{-1}K(x/h)$ is

$$\sigma_K(h) = \sqrt{\int_{-3h/2}^{3h/2} h^{-1}K(x/h)x^2\, dx} = \frac{h}{2} \tag{16}$$

so that the bandwidth (in relation to values $x \in \{0, \ldots, 255\}$) should be chosen as

$$h = 2\sigma_{\text{eff}} = \sqrt{4\sigma_f^2 + \frac{w^2}{3}} = 4\sqrt{\frac{91}{3}} \approx 22.03 \ . \tag{17}$$

However, small variations of h have no significant impact on the final results.

Finally, since the channels with lowest and highest index are outside the encoded interval, the CR consists of $b = \lceil 256/h \rceil + 2 = 14$ channels, encoding the interval $I_{\text{value}} = [-4.68;\ 259.68]$ (the interval of length $(b-2)h$ centered around the original interval).

EDFT In a third step, we regularize the predicted motion by a simple smoothing filter. This improves the continuity of the motion prediction beyond the second frame and helps to avoid being trapped in local minima. The new predicted motion $\mathbf{m}_{p,\text{new}}$ is computed from the previous one $\mathbf{m}_{p,\text{old}}$ and the old and new positions ($\mathbf{p}_{\text{old}}$ respectively $\mathbf{p}_{\text{new}}$) as

$$\mathbf{m}_{p,\text{new}} = \frac{1}{2}(\mathbf{m}_{p,\text{old}} + \mathbf{p}_{\text{new}} - \mathbf{p}_{\text{old}}) \ . \tag{18}$$

This simple recursive filter leads to a long sustain of motion predictions. The resulting method, Enhanced DFT (EDFT) has exactly the same number of independent parameters as the original DFT method, with the only difference that the feature kernel variance σ_f is replaced with the interval length of I_{value}, implicitly determining the bandwidth parameter h.

3. Experiments

All considered methods have been evaluated according to the rules of the VOT2013 challenge as specified in the VOT2013 evaluation kit document [1]. The authors guarantee that they have exactly followed the guidelines and have not modified the obtained results in any way that would violate the challenge rules.

3.1. Dataset and Evaluation

The VOT2013 challenge consists of 16 color image sequences with 172 to 770 frames: bicycle, bolt, car, cup,

david, diving, face, gymnastics, hand, iceskater, juice, jump, singer, sunshade, torus, and woman. The sequences have been selected to make the tracking a challenging task: objects change aspect or are articulated, the scale and orientation vary, illumination changes and occlusions occur. Some example frames are shown in figure 2.

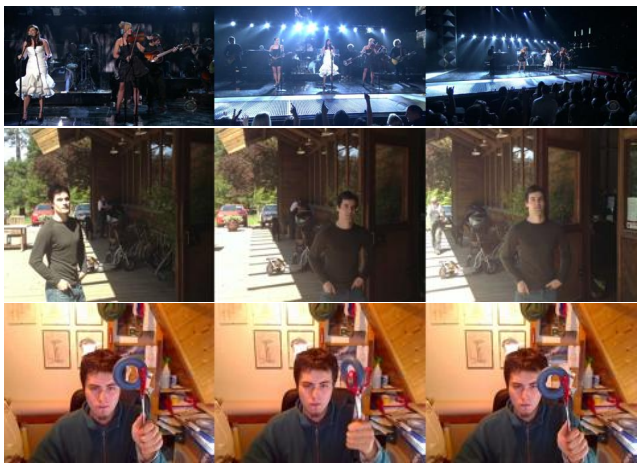

Figure 2. Examples from the dataset: singer, sunshade, torus.

The VOT2013 challenge foresees to evaluate three aspects of tracking: accuracy, robustness, and speed. The *speed* is computed as the average number of frames that are processed per second. The *robustness* is determined by the failure rate. Each time the tracker fails to achieve at least a partial overlap, the failure count increases and the tracker is re-initialized after five frames.

The *accuracy* is computed from the relative overlap of the ground truth bounding box and the tracked object bounding box, *i.e.* the intersection area of the two bounding boxes divided by their joined area. The accuracy is only evaluated if the tracking has not failed and therefore even methods that regularly fail do not necessarily have a poor accuracy. Accuracy evaluation starts 10 frames after (re-) initialization.

In total, three different experiments are preconfigured. Experiment 1, *baseline*, runs the tracker on the sequences as they are with ground truth position as initialization. Experiment 2, *region noise*, runs the tracker as before but with perturbated initialization (10% of the bounding box size) in order to test stability with respect to the initialization. Experiment 3, *grayscale*, repeats experiment 1 after converting the sequences to grayscale. Since the considered methods only consider grayscale information, Experiment 1 and 3 are supposed to give the same result. All runs are repeated 15 times, which however only differ if the tracking method is non-deterministic or for experiment 2.

3.2. Comparisons

We compare the results of the proposed EDFT method with three state-of-the-art methods: the original DFT method [20] as published on the authors' project page[2], Mean-Shift Tracking [5] as provided by the VOT2013 challenge, also available at Matlab Central[3], and Multiple-Instance Learning [2], available as OpenCV beta version[4].

Since EDFT has been derived from DFT in three steps, cf. section 2.4 for details, we also evaluate the impact of each step: a) We compare the original DFT algorithm with our constant velocity prediction version DFTc. b) We compare DFTc with the, in expectation value sense, equivalent channel-based algorithm CBDF. c) We compare CBDF with EDFT using its modified motion prediction.

All methods have been evaluated using their standard parameter settings: The DFT parameters are chosen according to the cross-validation optimization [20], the proposed methods use the same parameters or the equivalent parameters as derived in section 2.4. MST and MIL are applied with the standard parameters as downloaded.

The experimental environment consists of one Intel Xeon X5675 (3.07GHz), in a multicore machine with108GB shared RAM, running CentOS 6.4 64-bit. All algorithms except for MIL, which is implemented in C++/OpenCV, are native Matlab implementations and are executed with Matlab R2013a.

3.3. Results

Table 1. Experiment DFT region_noise

	accuracy	**robustness**	**speed (fps)**
bicycle	0.46	0.73	8.53
bolt	0.68	4.27	7.87
car	0.42	0.80	9.61
cup	0.70	0.20	7.06
david	0.60	1.07	5.09
diving	0.34	3.87	7.55
face	0.75	0.00	4.70
gymnastics	0.52	2.93	5.96
hand	0.45	2.73	7.46
iceskater	0.33	1.27	4.54
juice	0.72	1.27	7.03
jump	0.58	0.13	6.70
singer	0.36	0.27	3.17
sunshade	0.59	3.20	6.70
torus	0.73	0.67	7.09
woman	0.61	1.40	7.21

[2] http://people.cs.umass.edu/~lsevilla/trackingDF.html

[3] http://www.mathworks.com/matlabcentral/fileexchange/35520-mean-shift-video-tracking

[4] https://github.com/opencv-gsoc/gsoc11_tracking

Table 2. Experiment MST region_noise

	accuracy	robustness	speed (fps)
bicycle	0.37	1.93	15.50
bolt	0.45	1.27	11.43
car	0.43	1.00	19.40
cup	0.69	0.00	21.55
david	0.50	2.07	30.69
diving	0.31	6.40	18.17
face	0.67	0.00	15.91
gymnastics	0.39	8.93	20.61
hand	0.48	0.93	17.30
iceskater	0.55	3.87	25.42
juice	0.63	1.20	23.43
jump	0.54	0.00	17.16
singer	0.44	4.27	21.80
sunshade	0.65	1.80	14.52
torus	0.42	0.87	14.11
woman	0.53	5.53	44.54

Table 3. Experiment MIL region_noise

	accuracy	robustness	speed (fps)
bicycle	0.51	0.07	7.49
bolt	0.58	6.87	7.73
car	0.41	0.00	7.68
cup	0.63	0.53	7.89
david	0.48	0.07	8.33
diving	0.36	3.20	8.70
face	0.53	0.00	7.90
gymnastics	0.53	3.73	9.08
hand	0.42	1.93	7.86
iceskater	0.54	0.07	8.48
juice	0.58	0.00	7.95
jump	0.56	0.27	7.78
singer	0.33	0.00	9.48
sunshade	0.54	2.60	7.76
torus	0.50	3.07	7.78
woman	0.59	3.80	9.72

Table 4. Experiment EDFT region_noise

	accuracy	robustness	speed (fps)
bicycle	0.44	0.00	18.45
bolt	0.72	0.93	16.28
car	0.42	0.53	17.91
cup	0.73	0.00	14.17
david	0.66	0.40	11.04
diving	0.36	4.07	12.17
face	0.77	0.00	10.58
gymnastics	0.54	2.60	9.58
hand	0.48	1.47	13.45
iceskater	0.40	3.80	8.78
juice	0.58	0.00	13.60
jump	0.60	0.00	12.90
singer	0.36	0.33	6.04
sunshade	0.62	2.40	11.27
torus	0.76	0.00	14.56
woman	0.60	1.87	14.81

Table 5. Experiment DFTc region_noise

	accuracy	robustness	speed (fps)
bicycle	0.44	0.07	12.14
bolt	0.71	1.00	10.75
car	0.42	0.67	11.39
cup	0.73	0.00	9.70
david	0.62	0.67	7.08
diving	0.37	3.40	9.09
face	0.77	0.00	6.39
gymnastics	0.56	3.13	6.83
hand	0.51	1.73	9.64
iceskater	0.41	3.87	5.27
juice	0.58	0.00	8.93
jump	0.60	0.00	9.10
singer	0.36	0.27	3.81
sunshade	0.63	2.27	8.77
torus	0.76	0.07	10.29
woman	0.59	1.13	8.81

We perform the analysis on the noisy initialization results with respect to accuracy, robustness, and speed, see tables 1, 2, 3, 4, 5, and 6. The baseline/grayscale results are not further considered because the region noise results are more significant, see [21] for a detailed argumentation.

The results of the comparison to the state-of-the-art methods are summarized in table 7 , where mean and median values for all three measures and the four methods DFT, MST, MIL, and EDFT are listed for the noisy initialization experiment.

In order to analyze what exactly caused the improved performance of EDFT compared to the original DFT method, the intermediate results of each step from section 2.4 are summarized in table 8.

3.4. Discussion of Results

As stated above, due to the higher significance of results, we only consider the results from the noisy initializations (see also [21]). Whenever a result is characterized as significant, the p-value of the t-test is below 1%. Strictly speaking, the p-value larger than 1% only implies that the null-hypothesis (the results stem from the same distribution) cannot be rejected at 1% significance level, but we use the common notion of *significant difference* if $p < 1\%$. The p-values have been computed on the basis of single tracking runs, *i.e.* $16 \times 15 = 240$ runs in total.

As it can be seen from table 7, DFT and EDFT perform significantly better than MST and MIL concerning accu-

Table 6. Experiment CBDF region_noise

	accuracy	robustness	speed (fps)
bicycle	0.44	0.00	17.92
bolt	0.68	1.73	15.86
car	0.42	0.60	17.69
cup	0.73	0.00	14.17
david	0.66	0.40	11.28
diving	0.36	3.73	12.63
face	0.77	0.07	10.67
gymnastics	0.54	2.73	9.18
hand	0.49	1.20	13.40
iceskater	0.40	4.13	8.80
juice	0.58	0.00	13.65
jump	0.60	0.00	12.69
singer	0.38	0.40	6.03
sunshade	0.62	2.80	11.57
torus	0.76	0.07	14.32
woman	0.60	1.87	14.41

Table 7. Summarized results for the region noise experiment, comparison to state-of-the-art (best scores in boldface)

method	DFT	EDFT	MST	MIL
mean accuracy	0.55	**0.57**	0.50	0.51
median accuracy	**0.59**	**0.59**	0.49	0.53
mean robustness	1.55	**1.15**	2.50	1.64
median robustness	1.17	0.47	1.54	**0.40**
mean speed	6.64	12.85	**20.72**	8.23
median speed	7.05	13.18	**18.79**	7.90

Table 8. Summarized results for the region noise experiment, analysis of the DFT improvements (best scores in boldface)

method	DFT	DFTc	CBDF	EDFT
mean accurracy	0.55	**0.57**	0.56	**0.57**
median accurracy	**0.59**	**0.59**	**0.59**	**0.59**
mean robustness	1.55	**1.14**	1.23	1.15
median robustness	1.17	0.67	0.50	**0.47**
mean speed	6.64	8.62	12.77	**12.85**
median speed	7.05	9.01	13.05	**13.18**

racy. The accuracy of DFT and EDFT does not differ significantly. Similarly, MST and MIL show insignificant difference of accuracy.

For robustness, EDFT obviously outperforms DFT and MST significantly. EDFT has a better mean robustness score than MIL, but MIL has a better median robustness score than EDFT. In general, a high discrepancy between mean and median indicates the presence of outliers. Thus, MIL shows fewer tracking failures than EDFT on the majority of cases, but in the remaining cases, MIL fails more often than EDFT. This is confirmed by comparing the numbers for the individual experiments in tables 3 and 4. Applying the t-test shows that EDFT is *significantly better* than MIL, despite MIL having the better median. Quite surprisingly, the robustness of DFT and MIL do not differ significantly.

The observation from table 7 that EDFT has up to the significance level the same accuracy as DFT, but has significantly better robustness can be explained from the evaluation setup. As argued in section 3.1, the accuracy measure might even increase in case of growing number of tracking failures, because the failure cases are not considered in the accuracy calculation. Thus, EDFT can be considered as at least as accurate as DFT if both succeed to track, but EDFT is significantly less likely to fail than DFT.

For the computational speed, all differences are significant and MST is fastest, EDFT second, MIL third, and DFT slowest. Thus, to summarize, EDFT produces the best results in accuracy and robustness for the second best speed. DFT produces the best result in terms of accuracy and second best in terms of robustness, but is also slowest. MIL produces the least accurate results and second best robustness for the second highest computational cost. Finally MST produces the least accurate and robust results at the lowest computational cost.

While accuracy remains at the same level by going from DFT to EDFT, speed and robustness improve significantly. Using table 8, we analyze which steps lead to this significant improvement. Going from DFT to DFTc, *i.e.* by using a symmetric search window and the constant motion prediction, both robustness and speed improve significantly. Closer analysis shows that the constant motion prediction is often more accurate and thus fewer local search steps are required. Also the risk to end up in a local minimum is reduced.

Replacing DFs with CRs, *i.e.*, comparing DFTc and CBDF, does not influence robustness significantly, which is to be expected from the asymptotic equivalence of the two methods. However, the CR-based approach is significantly (50%) faster, which is also expected from the theory. Thus, the observed results can be considered as a positive verification of the theoretic findings.

The smoothed motion prediction (18) in EDFT improves the robustness and speed significantly compared to CBDF. The improvement can be explained with the same argument as the improvement when going from DFT to DFTc: The more accurate the prediction, the fewer iterations are required (better speed) and the lower is the risk for ending up in a local minimum (robustness).

Thus, EDFT is superior or equal to all compared methods in all measures except for speed in the case of MST. Note that the parameters of EDFT have not been tuned to the dataset, but have been derived directly from the proposed DFT parameters. Therefore, further performance improvement by parameter tuning is possible.

We have also compared other modifications of the algorithm according to [12], using downsampling of the spatial domain (channel-coded feature maps – CCFMs), Hellinger

and Euclidean distance, as well as extending to color images using RGB channels and using $\cos^2$ basis functions. Without giving the details here, all results were inferior to or at least not significantly better than using dense CRs, the L_1-distances, and B-spline channels.

However, we assume that cross-validation optimization of parameters might change the relative ranking. The parameters as obtained from section 2.4 are well suited for the L_1-distance and full resolution, but might be suboptimal for CCFMs and other distance measures. The poorer performance of the color-based variant can also be explained by the suboptimal parameters and possibly by the color characteristics of the dataset.

4. Conclusion

Visual tracking of objects under varying lighting conditions and changes of the object appearance, such as articulation and change of aspect, remains a challenging problem with room for improving the state-of-the-art. In the present paper we have used the theoretic connection between distribution fields, averaged shifted histograms, and channel representations to derive an enhanced computational scheme for distribution field tracking. This enhanced distribution field tracking method outperforms state-of-the-art methods in the VOT2013 challenge. It achieves highest accuracy and robustness at the second highest speed. Future extensions will address adaptive windows sizes, embedding of color information, and more adaptive model update schemes.

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 ELLIIT, and the Linnaeus research environment CADICS.

References

[1] The VOT 2013 evaluation kit. http://www.votchallenge.net.

[2] B. Babenko and M.-H. Y. S. Belongie. Robust object tracking with online multiple instance learning. 2011.

[3] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[4] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1158–1163. IEEE, 2005.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, pages 2142–. IEEE Computer Society, 2000.

[6] M. Felsberg. Incremental computation of feature hierarchies. In *Pattern Recognition 2010, Proceedings of the 32nd DAGM*, 2010.

[7] M. Felsberg. *Mathematical Methods for Signal and Image Analysis and Representation*, volume 41 of *Computational Imaging and Vision*, chapter Adaptive Filtering using Channel Representations, pages 31–48. Springer, 2012.

[8] M. Felsberg, P.-E. Forssén, and H. Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):209–222, 2006.

[9] P.-E. Forssén. *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, Sweden, 2004.

[10] G. H. Granlund. An associative perception-action structure using a localized space variant information representation. In *Proc. Int. Workshop on Algebraic Frames for the Perception-Action Cycle*. Springer, Heidelberg, 2000.

[11] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with ssd. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–790. IEEE, 2004.

[12] E. Jonsson. *Channel-Coded Feature Maps for Computer Vision and Machine Learning*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, February 2008. Dissertation No. 1160, ISBN 978-91-7393-988-1.

[13] E. Jonsson and M. Felsberg. Efficient computation of channel-coded feature maps through piecewise polynomials. *Image and Vision Computing*, 27(11):1688–1694, 2009.

[14] Y. Ma, H. Mobahi, and C. L. Zitnick. Seeing through the blur. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1736–1743, 2012.

[15] H. T. Nguyen, M. Worring, and R. van den Boomgaard. Occlusion robust adaptive template tracking. In *ICCV*, pages 678–683, 2001.

[16] A. Pagani, D. Stricker, and M. Felsberg. Integral P-channels for fast and robust region matching. In *ICIP*, 2009.

[17] A. Pouget, P. Dayan, and R. Zemel. Information processing with population codes. *Nature Reviews – Neuroscience*, 1:125–132, 2000.

[18] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, 1992.

[19] D. W. Scott. Averaged shifted histograms: Effective nonparametric density estimators in several dimensions. *Annals of Statistics*, 13(3):1024–1040, 1985.

[20] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *IEEE Computer Vision and Pattern Recognition*, 2012.

[21] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[22] R. S. Zemel, P. Dayan, and A. Pouget. Probabilistic interpretation of population codes. *Neural Computation*, 10(2):403–430, 1998.