J Math Imaging Vis (2018) 60:929–940
https://doi.org/10.1007/s10851-017-0775-8

Approximative Coding Methods for Channel Representations

Kristoffer Öfjäll[1] · **Michael Felsberg**[2]

Received: 19 June 2017 / Accepted: 10 November 2017 / Published online: 21 November 2017

Abstract Most methods that address computer vision problems require powerful visual features. Many successful approaches apply techniques motivated from nonparametric statistics. The channel representation provides a framework for nonparametric distribution representation. Although early work has focused on a signal processing view of the representation, the channel representation can be interpreted in probabilistic terms, e.g., representing the distribution of local image orientation. In this paper, a variety of approximative channel-based algorithms for probabilistic problems are presented: a novel efficient algorithm for density reconstruction, a novel and efficient scheme for nonlinear gridding of densities, and finally a novel method for estimating Copula densities. The experimental results provide evidence that by relaxing the requirements for exact solutions, efficient algorithms are obtained.

Keywords Visual features · Channel representations · Approximative density estimation · Maximum entropy

1 Introduction

Visual feature descriptors are essential to solve computer vision problems with state-of-the-art methods. Although deep learning [18] eliminates the need to design feature descriptors by hand, approximative algorithms for probabilistic processing of feature layers are useful, e.g., for visualization [20,31]. Furthermore, certain problems require more light-weight solutions and cannot make use of deep learning. Instead, combinations of designed feature descriptors with shallow networks or other machine learning approaches are more appropriate and produce good results, e.g., for real-time online learning of path following [21,23]. A demonstration video of such a system is available online (https://goo.gl/JcvqHz). The system requires obtaining a full reconstruction of represented probability densities. Furthermore, the representation should be adapted to nonlinear domains, such as depth. In cases where there are dependencies between signals, statistical approaches are expected to improve if the dependency structure can be properly handled and separated from the marginal distributions.

Besides for machine learning, feature descriptors such as HOG [1], SIFT [19], and distribution fields (DFs) [27] are also used in multi-view geometry (point matching) and visual tracking. Thus, they are of central importance to visual computing. All these approaches have in common that they compute local histograms, e.g., of local orientation, and are thus related to nonparametric density estimation. Consider the case of DFs: the image is *exploded* into several layers representing different ranges of intensity; see Fig. 1.

Whereas DFs make an ordinary bin assignment and apply post-smoothing, *channel representations* apply a soft-assignment, i.e., pre-smoothing. This has shown to be more efficient [4]. Similarly, SIFT descriptors can be considered as a particular variant of channel representation of local orientation and the latter framework allows generalizing to color images [7]. HOG descriptors are a specific variant of channel coded feature maps CCFMs [16], but in contrast to the former no additional visualization [30] is required. CCFMs

Both authors share a joint first authorship.

✉ Michael Felsberg
michael.felsberg@liu.se

Kristoffer Öfjäll
kristoffer.ofjall@visionists.se

[1] Visionists AB, Gothenburg, Sweden

[2] Computer Vision Laboratory, Linköping University, Linköping, Sweden

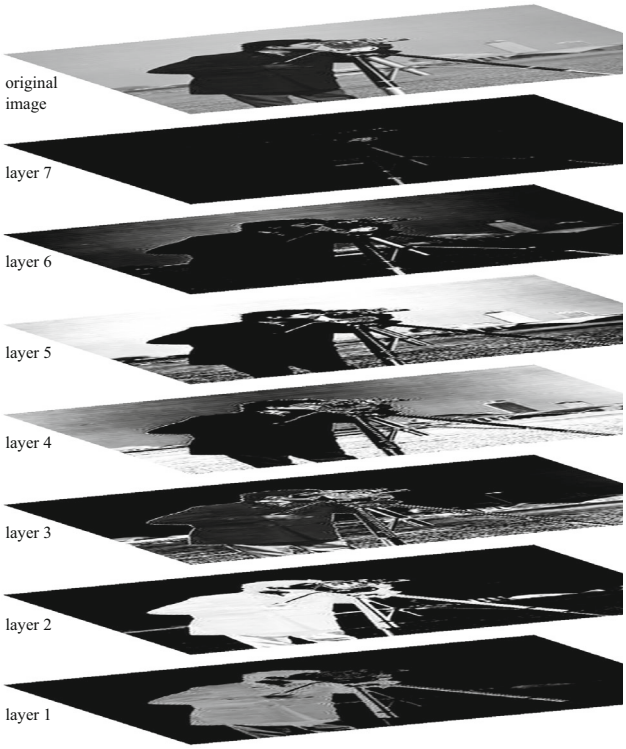

Fig. 1 Illustration of distribution fields: the image (top) is *exploded* into several layers (here: 7), each covering a different interval of the grayscale range. In these layers, intensity represents activation, where dark is no activation and white is full activation. Each layer represents a range of intensity values of the original image. The bottom layer represents dark intensities, i.e., the high activations in the bottom layer are at pixels with low image intensity. Each new layer above the bottom one represents respectively higher intensities. In the seventh layer, the high image intensity pixels appear active

are based on frame theory, which comes with a decoding methodology that also covers visual reconstruction [3].

Thus, channel representations is a general framework for building feature descriptors, and the goal of this article is to formulate efficient algorithms for three different tasks:

- From the measured coefficients in the nonparametric density representation, a continuous density is to be estimated under the assumption of minimum information (maximum entropy) [15].
- Whereas histogram bins are often equally distributed, i.e., the bin centers sample the input space regularly, highly varying densities require a nonlinear transformation of the input space before gridding. The resulting non-constant measure is to be compensated during the non-regular gridding of the input space.
- A joint density can be turned into a Copula distribution by transforming its marginals into uniform distributions. Similar to the second problem, the induced measure is to be taken care of during the calculation of the Copula distribution.

The remainder of the article is structured as follows. Section 2 reviews relevant methods and properties of channel representations. Section 3 addresses the first problem of efficient maximum entropy reconstruction. Section 4 addresses the second problem of non-regular gridding of the input space. Section 5 addresses the transformation of densities to uniform distribution for the estimation of the Copula distribution. The article is concluded with Sect. 6.

2 The Channel Representation

The channel representation has been proposed by Granlund [11]. It shares similarities to population codes [25,29] and similar to their probabilistic interpretation [32] they approximate a kernel density estimator [5]. The mathematical proof has basically already been given in context of averaged shifted histograms [26]. A further related representation, orientation scores, is based on generalized wavelet theory [2].

An intuitive understanding of channel representations, including their encoding and decoding, is obtained by considering the example of channel smoothing [5,6], which is sometimes also considered as an efficient version of bilateral filtering [17,24]; see Fig. 2.

Bilateral filtering allows to denoise a signal or an image without blurring edges because the different intensity/color levels on the two sides of the edge are represented in different parts of the model after the encoding. Thus, the two levels are not confused during spatial averaging. Instead, close to the edge a metamery region is formed, i.e., two different modes occur. The task during decoding is then to pick the stronger mode and to determine its maximum.

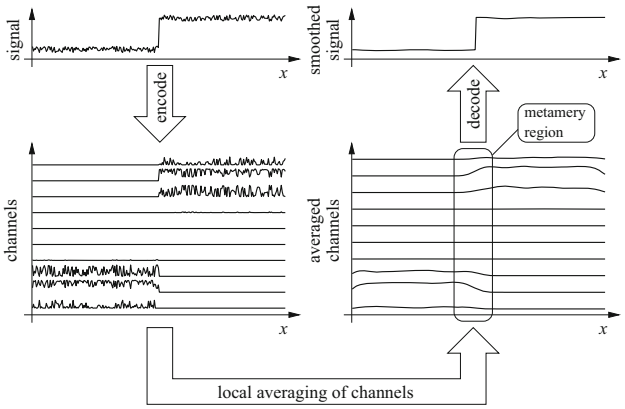

Fig. 2 Illustration of channel smoothing [6]: The noisy signal (left) is smoothed without blurring the edge (right). This is achieved by encoding, spatial averaging of the channels, and decoding

2.1 Encoding

This section makes use of notation and derivations according to [15]. The channel representation is built by channel encoding samples $x^{(m)}$ from a distribution with density p, resulting in the channel vector

$$\mathbf{c}^{(m)} = [c_1^{(m)}, \ldots, c_N^{(m)}]^t \tag{1}$$
$$= [K(x^{(m)} - \xi_1), \ldots, K(x^{(m)} - \xi_N)]^t, \tag{2}$$

where m denotes the sample index, c_n the channel coefficients, $K()$ the encoding kernel, and ξ_n the channel centers.

In contrast to previous work on maximum entropy reconstruction [15], we will use $\cos^2$-kernels instead of quadratic B-splines

$$K(x) = \begin{cases} \frac{2}{3}\cos^2(\pi x/3) & |x| \le 3/2 \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The reason for this choice is the uniqueness of $\cos^2$-kernels as minimal overlap kernels on the regular grid with constant l_2-norm, see Theorems 2.1 and 2.2 in [8].

Consider the sample set $\{x^{(m)}\}$ of size M. Summing over it, the nth channel coefficient becomes

$$c_n = \frac{1}{M}\sum_{m=1}^{M} c_n^{(m)} = \frac{1}{M}\sum_{m=1}^{M} K(x^{(m)} - \xi_n). \tag{4}$$

Since we draw the samples $x^{(m)}$ from the density p, the expectation of c_n is

$$\mathrm{E}[c_n] = \int_{-\infty}^{\infty} p(x) K(x - \xi_n)\,\mathrm{d}x. \tag{5}$$

2.2 Decoding/Reconstruction

Various ways to decode channel representations for different kernels have been suggested in the past [5,10]. For the $\cos^2$-kernel, different degrees of overlap and confidence measures have been considered [10]. In this short review, we describe the recently suggested maximum likelihood decoding [8].

The first step is to select an index n of $\mathbf{c}$, which will be the center of the decoding window of width three

$$\mathbf{c} = [\ldots, c_{n-2}, \underbrace{c_{n-1}, c_n, c_{n+1}}_{\text{decoding window}}, c_{n+2}, \ldots]^t. \tag{6}$$

How to select this index will be explained below.

By rotating the 3-vector in the decoding window $\mathbf{c}_n = [c_{n-1}, c_n, c_{n+1}]^t$, we obtain the $\mathbf{p}_n$ vector, which is parametrized in (r_n, s_n, α_n)

$$\mathbf{p}_n := \begin{bmatrix} r_n \cos(2\pi\alpha_n/3) \\ r_n \sin(2\pi\alpha_n/3) \\ s_n \end{bmatrix}$$
$$= \frac{1}{\sqrt{3}} \begin{bmatrix} \sqrt{2} & \sqrt{2}\cos(2\pi/3) & \sqrt{2}\cos(4\pi/3) \\ 0 & \sqrt{2}\sin(2\pi/3) & \sqrt{2}\sin(4\pi/3) \\ 1 & 1 & 1 \end{bmatrix} \mathbf{c}_n.$$

Usually,[1] $\alpha_n \in [\pi/3; \pi]$, and we select the decoding window according to

$$\hat{n} = \arg\max_n r_n + \sqrt{2} s_n. \tag{7}$$

The corresponding decoded value $\hat{x} = \max(\min(\frac{3}{2\pi}(\alpha_{\hat{n}} - 2\pi/3), \frac{1}{2}), -\frac{1}{2}) + \hat{n}$ is the maximum likelihood estimate of $\mathbf{c}$ assuming independent noise [8]

$$\hat{x} = \arg\max_x p(x|\mathbf{c}) \tag{8}$$
$$= \arg\min_x \|[K(x - \xi_1), \ldots, K(x - \xi_N)]^t - \mathbf{c}\|_2^2.$$

2.3 Maximum Entropy Reconstruction

In contrast to the decoding as suggested above, which just estimates the mode of the distribution, maximum entropy decoding [15] attempts to extract the whole distribution. The idea is to find the simplest, i.e., the least informative, distribution, which fits the channel coefficients, by maximizing its differential entropy

$$H(p) = -\int_{-\infty}^{\infty} p(x) \log p(x)\,\mathrm{d}x. \tag{9}$$

Fitting the channel coefficients is guaranteed by the constraints

$$\int_{-\infty}^{\infty} p(x) K(x - \xi_n)\,\mathrm{d}x = c_n, \quad 1 \le n \le N \tag{10}$$
$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1. \tag{11}$$

Using a variational approach with Lagrange multipliers λ_n, $0 \le n \le N$, we obtain

$$p(x) = \exp\lambda_0 \exp\left(\sum_{n=1}^{N} \lambda_n K(x - \xi_n)\right). \tag{12}$$

To the best of our knowledge, the explicit solution of λ_n cannot be calculated, and it has been suggested to apply a

[1] If α_n is outside that range, r_n needs to be modified [8].

Newton method using numerical evaluations of the integrals on a very fine grid [15]. Obviously, this comes with an enormous efficiency penalty and is thus only interesting for single simulations.

3 Maximum Approximative Entropy Reconstruction

In order to improve efficiency, the differential entropy as used in previous work [15] is approximated using the linear Taylor expansion of the logarithm in (9)

$$H_2(p) = \int_{-\infty}^{\infty} \frac{3}{2} p(x)(1 - p(x))\,\mathrm{d}x. \tag{13}$$

This objective is maximized under the same constraints (10) and (11). Using a variational approach with Lagrange multipliers λ_n, $0 \le n \le N$, we obtain:

$$p(x) = \frac{\lambda_0}{3} + \frac{1}{2} + \frac{1}{3}\sum_{n=1}^{N} \lambda_n K(x - \xi_n) \tag{14}$$

Note the finite support of K and the infinite integration in (11) imply $\lambda_0 = -\frac{3}{2}$. Thus, the first two terms in (14) cancel out and we will skip λ_0 in what follows.

The approximation is limited to a linear expansion in (13) to simplify subsequent equations. Higher orders might lead to better accuracy, but at the cost of significantly more complicated solution than (14).

3.1 Direct Solution

In contrast to previous work [15], (14) can be directly inserted into (10), resulting in:

$$\begin{aligned}
c_n &= \int_{-\infty}^{\infty} \left(\frac{1}{3} \sum_{n'=1}^{N} \lambda_{n'} K(x - \xi_{n'}) \right) K(x - \xi_n)\,\mathrm{d}x \\
&= \frac{1}{3} \sum_{n'=1}^{N} \lambda_{n'} \int_{-\infty}^{\infty} K(x - \xi_{n'}) K(x - \xi_n)\,\mathrm{d}x \\
&= \frac{1}{3} \sum_{n'=1}^{N} \lambda_{n'} \begin{cases} \frac{1}{2} & n = n' \\ \frac{1}{6} + \frac{\sqrt{3}}{8\pi} & n = n' \pm 1 \\ \frac{1}{12} - \frac{\sqrt{3}}{8\pi} & n = n' \pm 2 \end{cases} \\
&= \frac{1}{3} \left(\left(\frac{1}{12} - \frac{\sqrt{3}}{8\pi} \right) (\lambda_{n+2} + \lambda_{n-2}) + \right. \\
&\quad \left. + \left(\frac{1}{6} + \frac{\sqrt{3}}{8\pi} \right) (\lambda_{n+1} + \lambda_{n-1}) + \frac{\lambda_n}{2} \right)
\end{aligned}$$

where $\lambda_n = 0$ if $n < 1$ or $n > N$. Note that $\mathbf{c}$ is obtained from $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]^t$ by a discrete linear filter such that the sums of components behave as $\sum_{n=1}^{N} c_n = \frac{1}{3}\sum_{n=1}^{N} \lambda_n$. Thus, a normalized $\mathbf{c}$ implies that the sum of Lagrange multipliers $\boldsymbol{\lambda}$ is 3 and we obtain the linear system

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{c} \tag{15}$$

where

$$\mathbf{A} = \frac{1}{3} \begin{bmatrix} a_0 & a_1 & a_2 & 0 & \ldots & 0 \\ a_1 & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_2 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_1 \\ 0 & \ldots & 0 & a_2 & a_1 & a_0 \end{bmatrix} \tag{16}$$

with

$$a_0 = \frac{1}{2} \quad a_1 = \frac{1}{6} + \frac{\sqrt{3}}{8\pi} \quad a_2 = \frac{1}{12} - \frac{\sqrt{3}}{8\pi}. \tag{17}$$

Once the coefficients λ_n are determined from (15), we can exploit (14) to compute necessary conditions for local maxima x_0 by requiring a vanishing first derivative and a negative second derivative, i.e.,

$$p'(x_0) = \frac{1}{3}\sum_{n=1}^{N} \lambda_n K'(x_0 - \xi_n) = 0 \tag{18}$$

$$p''(x_0) = \frac{1}{3}\sum_{n=1}^{N} \lambda_n K''(x_0 - \xi_n) < 0. \tag{19}$$

From (3) we determine

$$K'(x) = \begin{cases} -\frac{4\pi}{9} \sin(2\pi x/3) & |x| \le 3/2 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

$$K''(x) = \begin{cases} -\frac{8\pi^2}{27} \cos(2\pi x/3) & |x| \le 3/2 \\ 0 & \text{otherwise} \end{cases}. \tag{21}$$

Instead of inverting the matrix (16), we derive a recursive filter that traverses the channel vector $\mathbf{c}$ forth and back, similar to the decoding method for B-spline kernels [5]. We start looking at the z-transform of the filter realized by (16) (defining $a = \frac{1}{3} - \frac{\sqrt{3}}{2\pi}$)

$$H(z) = \frac{az^{-2} + (1-a)z^{-1} + 2 + (1-a)z + az^2}{12} \tag{22}$$

and thus we obtain

$$H^{-1} = \frac{12z^{-2}}{a + (1-a)z^{-1} + 2z^{-2} + (1-a)z^{-3} + az^{-4}} = \frac{12}{a}\frac{1}{z^{-2} - z_1 z^{-1} + 1}\frac{z^{-2}}{z^{-2} - z_2 z^{-1} + 1} \tag{23}$$

where

$$z_{1/2} = \frac{1}{2} - \frac{1}{2a} \pm \frac{\sqrt{a^{-2} - 10a^{-1} + 9}}{2}. \tag{24}$$

Hence, we get the following recursions

$$\begin{aligned} c_n^+ &= c_n + z_1 c_{n-1}^+ - c_{n-2}^+ \qquad (n = 3, \ldots, N) \\ c_n^- &= c_n^+ + z_2 c_{n+1}^- - c_{n+2}^- \qquad (n = N-2, \ldots, 1) \\ \lambda_n &= \frac{12}{a} c_n^- \qquad (n = 1, \ldots, N). \end{aligned} \tag{25}$$

It has been assumed that $c_n = 0$ for $n < 1$ or $n > N$. Therefore, the initial conditions of the filters are[2]

$$c_1^+ = c_1 \qquad c_2^+ = c_2 + z_1 c_1 \tag{26}$$

$$c_N^- = c_N^+ \qquad c_{N-1}^- = c_{N-1}^+ + z_2 c_N^+. \tag{27}$$

In contrast to (12), which is nonnegative by design, negative λ_n might lead to (14) violating the nonnegativity property of density functions and a separate consideration of this property is required.

3.2 Nonnegativity Constraint

Conjecture 1 According to (14) let

$$p(x) = \frac{1}{3}\sum_{n=1}^{N} \lambda_n K(x - \xi_n) \tag{28}$$

then $p(x) \geq 0$ iff for all $n = 1, 2, \ldots, N$

$$\lambda_n < 0 \rightarrow \sqrt{\sum_{k=n-1}^{n+1} \lambda_k^2} \leq \sum_{k=n-1}^{n+1} \lambda_k \quad , \tag{29}$$

where coefficients outside the valid range are taken to be $\lambda_0 = \lambda_{N+1} = 0$.

This conjecture is motivated by simulation results where the nonnegativity of $p(x)$ has been studied for increasingly finer grids in the space of reconstruction coefficients λ. For any negative coefficient λ_n, all valid solutions, and no invalid solutions are within the cone with twice the radius of the cone of valid channel representations $\mathbf{c}$. This can be expressed as the relation of the l_2 norm and the sum over three coefficient windows. Since the overlap is three, it is necessary and sufficient for the condition to be satisfied for all such windows. The condition on one such window is illustrated in Fig. 3, where the coefficients in the window have been normalized to unit sum, allowing presentation in a plane. The symmetry axis of the cone is perpendicular to the plane and passes through the origin of the figure coordinate system. The general geometry of the channel representation is further explored in [8].

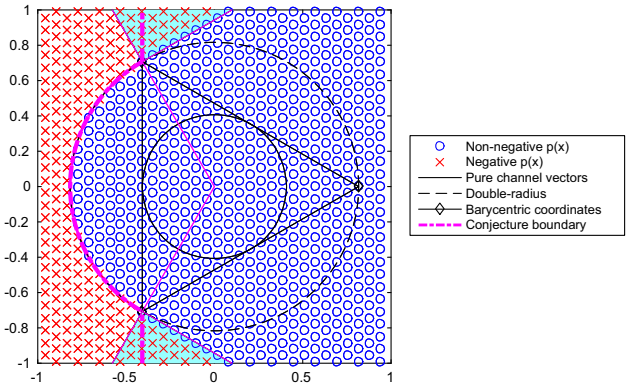

Fig. 3 Numerical verification of the conjecture. Red crosses indicate reconstruction coefficients generating function values below zero within the current decoding interval. Blue circles indicate nonnegative reconstructions. The boundary of the conjecture is indicated by a thick magenta line. The solid circle shows pure channel vectors, i.e., encodings of single values. The dashed circle, passing through $(1\ 0\ 0)^t$ and $(0\ 0\ 1)^t$, has precisely twice the radius of the solid line circle. Coefficient vectors are normalized such that $\sum_{k=n-1}^{n+1} \lambda_k = 1$. The solid line circle is a section from the cone of valid channel representations. Due to overlapping decoding intervals, the continuation of the conjecture boundary outside the dashed line circle can be chosen anywhere between the radial line and the tangential line at the transition point (the cyan area) (Color figure online)

These constraints can be enforced either in the channel space or the reconstruction space because they are connected by the linear operator $\mathbf{A}$. Enforcing the constraint in Conjecture 1 should not change the corresponding channel coefficients c_n by an arbitrary amount. From a statistical point of view, small coefficients build on fewer observations than large ones. The penalty for changing coefficients should thus scale with their value. This is fulfilled by the weighted quadratic error, and we thus aim to minimize

$$\varepsilon(\boldsymbol{\lambda}) = \|\mathbf{CA}\boldsymbol{\lambda} - \mathbf{Cc}\|_2 \qquad \text{s.t. } \lambda_n < 0 \implies \sqrt{\sum_{k=n-1}^{n+1} \lambda_k^2} \leq \sum_{k=n-1}^{n+1} \lambda_k \ , \ n = 1, \ldots, N \tag{30}$$

[2] Note that the boundary conditions (27) are nontrivial: Due to the instability of the filters, numerical results might differ. However, we know from our assumptions that all $\lambda_n = 0$ for $n < 1$ or $n > N$ and thus $c_n = 0$ for $n > N$.

where the diagonal weight matrix $\mathbf{C} = \mathrm{diag}(\mathbf{c})w + \mathbf{I}(1-w)$, with the parameter w controlling the influence of the weighting. The quadratic norm is a special case $w = 0$. The conditional constraint makes this problem hard to solve. We choose an iterative heuristic approach starting from $\boldsymbol{\lambda}$ according to (25). This initial $\boldsymbol{\lambda}$ results in two index sets, $\mathcal{C}^+$ and $\mathcal{C}^-$, such that $\mathcal{C}^+ \cap \mathcal{C}^- = \emptyset$, $\mathcal{C}^+ \cup \mathcal{C}^- = \{1, \ldots, N\}$, $\lambda_n \geq 0$ for $n \in \mathcal{C}^+$, and $\lambda_n < 0$ for $n \in \mathcal{C}^-$. We assume that coefficients λ_n will not change sign and thus $\mathcal{C}^-$ remains static.

Introducing Lagrange multipliers γ_n, $n \in \mathcal{C}^-$, we reformulate the optimization (30) as

$$\varepsilon(\boldsymbol{\lambda}) = \|\mathbf{C}\mathbf{A}\boldsymbol{\lambda} - \mathbf{C}\mathbf{c}\|_2 + \gamma_0 r_0 + \sum_{n \in \mathcal{C}^-} \gamma_n r_n \tag{31}$$

with

$$r_n = \sqrt{\sum_{k=n-1}^{n+1} \lambda_k^2} - \sum_{k=n-1}^{n+1} \lambda_k, \quad r_0 = \left(\sum_{n=1}^{N} \lambda_n - 3\right)^2 \tag{32}$$

the latter keeping the total weight constant.

Let $\mathbf{0}$ be a zero vector of suitable size,

$$\mathbf{r}_n = \frac{\mathrm{d}r_n}{\mathrm{d}\lambda} = \frac{1}{\sqrt{\lambda_{n-1}^2 + \lambda_n^2 + \lambda_{n+1}^2}} \begin{pmatrix} \mathbf{0} \\ \lambda_{n-1} \\ \lambda_n \\ \lambda_{n+1} \\ \mathbf{0} \end{pmatrix} - 1 \tag{33}$$

and

$$\mathbf{r}_0 = \frac{\mathrm{d}r_0}{\mathrm{d}\lambda} = 2\left(\sum_{n=1}^{N} \lambda_n - 3\right). \tag{34}$$

Furthermore, the gradient of the weighted quadratic norm is

$$\Delta_\lambda = \frac{1}{\|\mathbf{C}\mathbf{A}\boldsymbol{\lambda} - \mathbf{C}\mathbf{c}\|_2} \mathbf{A}^t \mathbf{C}^2 (\mathbf{A}\boldsymbol{\lambda} - \mathbf{c}). \tag{35}$$

A valid solution to (30) is thus found by iterating

$$\lambda := \lambda - a \left[\Delta_\lambda - \Delta_\lambda \parallel \mathrm{span}\{\mathbf{r}_n\}_{n \in \{0\} \cup \mathcal{C}^-} + \mathrm{r}_0 + \sum_{n \in \mathcal{C}^-} \mathrm{r}_n \right], \tag{36}$$

where $\Delta_\lambda \parallel \mathrm{span}\{\mathbf{r}_n\}_{n \in \{0\} \cup \mathcal{C}^-}$ is the part of Δ_λ in the subspace spanned by $\mathbf{r}_0$ and $\mathbf{r}_n$, $n \in \mathcal{C}^-$. The step length a is set to 0.1.

A faster convergence to valid solutions (however, not necessarily minimal) can be obtained by a Newton approach, replacing the last part of (36) with a solution $\mathbf{q}$ to $\mathbf{r} + \mathbf{q}^t \mathbf{R} = \mathbf{0}$, with $\mathbf{R} = (\mathbf{r}_0, \ldots, \mathbf{r}_n, \ldots)$ and $\mathbf{r} = (r_0, \ldots, r_n, \ldots)^t$ where $n \in \mathcal{C}^-$. Note that the equation system is underdetermined in general.

3.3 Simulation Experiments

The reconstruction procedures are evaluated on samples drawn from known distributions. The $N = 10$ channel coefficients $\mathbf{c}$ are set to their expected values, corresponding to infinitely many samples. From the channel coefficients, the maximum entropy and approximate entropy estimates of the distributions are calculated using the methods of Sects. 2.3 and 3, respectively.

The results are shown in Fig. 4, after six iterations and after convergence. The maximum entropy approach uses Newton iterations as suggested [15]. Note that each element of the Jacobian requires numerical evaluation of an integral. The maximum approximative entropy approach uses Newton iterations for fulfilling the nonnegativity constraints and gradient descent for minimizing (30). The Jacobian is obtained by matrix computations, with the number of elements related to the number of channels used. Using Matlab implementations and gridding the integrals at 100 points, each iteration of the maximum entropy approach requires 3–5 ms of computation time. For the maximum approximative entropy approach using Newton iterations, each iteration takes 0.5–0.6 ms.

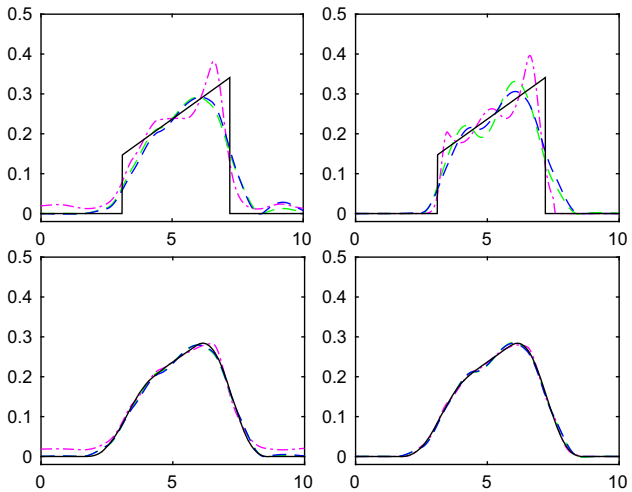

Fig. 4 Reconstruction experiment from known distributions. Left: six iterations. Right: iteration until convergence. Top: non-smooth distribution. Bottom: smooth distribution. Black: original distribution. Magenta: Max Entropy reconstruction [15]. Green: Max Approximative Entropy reconstruction ($w = 0$). Blue: Max Approximative Entropy reconstruction ($w = 0.9$) (Color figure online)

For samples drawn from distributions with smooth density functions, the initial solution using the approximate entropy is close to the final solution. For density functions with discontinuities (upper row), the initial solution obtains negative values. However, less than six iterations are required to obtain a valid density function. The use of a weighted norm (30) has a small impact on the final result, generating a solution slightly closer to the true distribution function in the high-density areas in Fig. 4, top right.

3.4 Regression Learning Experiments

The results from the simulation experiment above are confirmed by regression learning experiments. In these experiments, the head yaw angle for a set of people, taken from the Multi-PIE dataset [13], has to be estimated. The experiment is described in detail in [14] and the channel-based regression method has been described in [21]. Channel-based regression clearly outperforms robust regression as introduced in [14], which is why we use the former as baseline below.

We have repeated the same evaluation as in [21], but changed the decoding for calculating the yaw angle to the proposed approximative maximum entropy reconstruction; see Sect. 3.1, and subsequent detection of maxima. The results are displayed in Fig. 5 and show that the regression performance is significantly improved using the new decoding mechanism.

The experiment is providing a successively growing amount of training data to the regression method, which is evaluated on the respectively subsequent batch of data before using it for training. When comparing the performance of the new decoding method and the original method [21], we observe an increase in error after about 50 training samples, before both methods coincide after about 500 training samples. This intermediate decay of performance is presumably caused by secondary modes in the density function originating from the regression.

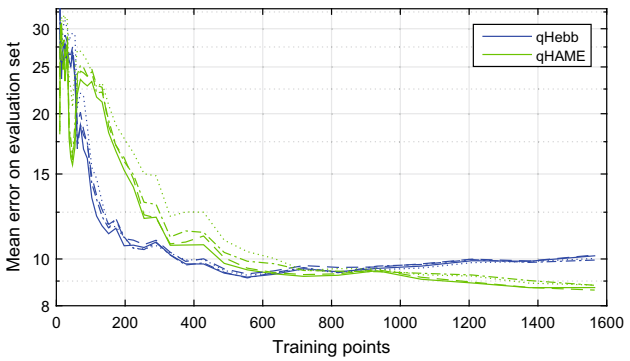

Fig. 5 Experiment from [21], Fig. 4. The solid, dashed, dash-dotted, and dotted lines correspond to 0, 20, 40, and 80% corrupted images, respectively. qHebb is the method from [21] and qHAME is the proposed method (Color figure online)

Beyond 1000 training samples, the baseline method with standard decoding [21] does not further improve, it even decays slightly, presumably caused by bias effects from the maximum operation on the channels. The proposed method, however, further improves performance until the end of the experiment and is likely to further improve if more data had been available. The final improvement of performance is larger than 15%.

4 Non-regular Channel Placement

In most applications, the channel centers are distributed evenly in the space to be represented. In certain applications, however, other channel placements are beneficial. In this section, logarithmic and log-polar placements are presented along with some results and pointers to suitable applications.

4.1 Logarithmic Channels

Using logarithmic channels, the ability to resolve nearby encoded values varies over the domain. One typical application would be encoding events in time, where high resolution is required for recent events and low resolution suffices for older events. Referring to an event "about an hour ago," the precision is some tens of minutes, while referring to an event "about 3 months ago," the precision is some tens of days.

Using logarithmic channel placement, the support of each channel is a constant factor wider than the support of the previous channel. The basis functions used are

$$K_n(x) = \cos^2\left((\log_d(x) - n)\frac{\pi}{3}\right)\frac{1}{x}, \quad d^{n-1.5} \le x \le d^{n+1.5} \tag{37}$$

and zero everywhere else. Using the base d logarithm, the parameter d determines the rate of expansion of the channels. The factor $\frac{1}{x}$ normalizes the weight of the basis functions, compare with the functional determinant of the logarithm. Recreating a continuous function from channel coefficients uses unscaled basis functions. The scaling can be moved from the analysis to the synthesis side. See Fig. 6.

Letting $d = 2$, each channel will be twice as wide as the previous channel. Instead letting $d = 2^{\frac{1}{3}}$, the basis function support will be doubled every third channel, i.e., when a channel support ends, the new channel will be twice as wide; see Fig. 7.

The major advantage of logarithmic transformations is that scaling of the encoded values will lead to a shift of the channel coefficients. In the example above, scaling values by a factor of two will lead to a shift of coefficient by three

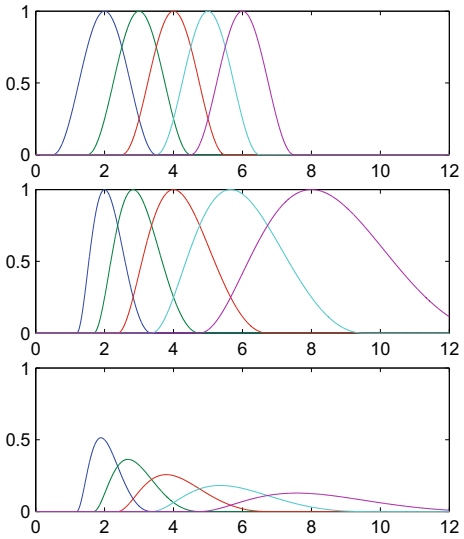

Fig. 6 From top to bottom: five regular channels, five logarithmic channels, and five scaled logarithmic channels with constant area (Color figure online)

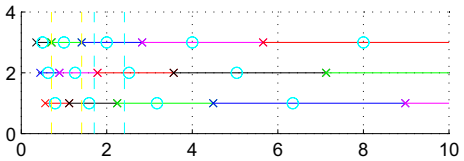

Fig. 7 Layout of basis function supports using logarithmic channels and expansion parameter $d = 2^{\frac{1}{3}}$. Crosses indicate the channel support bounds and overlapping channel supports are distributed on the three lines (Color figure online)

channels. Since humans often perceive entities in relative terms, see the example regarding temporal precision above or pitch spaces in music, the logarithmic mapping is biologically well-motivated. Also in projective geometry, relative changes are of interest, e.g., in depth estimation.

4.2 Log-Polar Channels

A polar coordinate system can be employed to extend the logarithmic channels to images. Log-polar coordinate systems have been applied to images before, e.g., for filter design in the Fourier domain [12] and similitude group invariant transforms, both globally [9] and locally [28].

In the log-polar channel arrangement, channels are regularly placed around concentric circles (representing orientation) with logarithmically increasing distance from the center. The setup stems from foveal vision, with higher resolution in the central parts; see Fig. 8.

The primary efficiency gain here stems from the resolution reduction further out in the visual field. This allows wider fields of view while avoiding the quadratic growth of the number of pixels in a regularly sampled image. Certainly,

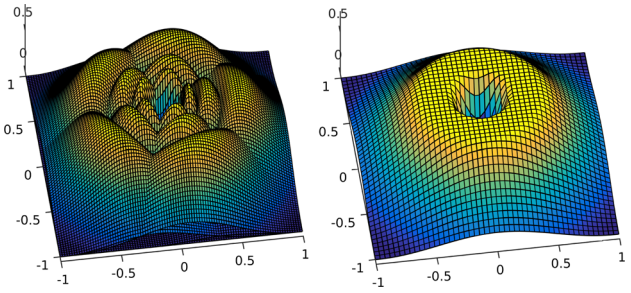

Fig. 8 Left: example of basis functions using three radial and five angular channels. For clarity of presentation, the normalization factor is removed and thus the amplitude of all basis functions are the same. Right: the sum of all normalized basis functions, generating a flat surface on the disk-shaped representable range

this is only applicable if the objects of interest can be moved to the central area of the image, e.g., pan-tilt cameras.

A Cartesian image position $(x,\ y)$ is mapped to the log-polar grid $(r,\ \theta)$ by the complex logarithm $r + i\theta = \mathrm{Log}(x + iy)$. The logarithmic radial position r and the angular position θ are encoded in an outer product channel representation.

The angular channels are modular, mending the branch cut of the logarithm function. This domain is represented by a periodic kernel

$$K_{\mathrm{mod}} = \sum_{k=-\infty}^{\infty} K\left(\frac{N\theta}{2\pi} - kN\right) \tag{38}$$

when using N channels to represent the angle θ. In the example of Fig. 8, using $N = 5$ channels with channel centers $\xi_n = n - 1/2$, the modular channel coefficients representing an angle θ are

$$c_n = \sum_{k=-\infty}^{\infty} K\left(\frac{5\theta}{2\pi} - (n - 1/2 + 5k)\right) \tag{39}$$

for $n = 1, \ldots, 5$. In practice the usual non-periodic kernel (3) is used. Since the kernel has compact support and assuming θ in the range 0–2π, the summation is limited to $k \in \{-1, 0, 1\}$. Note further that for $k = 1$ and the maximum $\theta = 2\pi$, $K(-n + 1/2) \neq 0$ implies $n = 0$. Similarly, for $k = -1$ and the minimum $\theta = 0$, $K(5 - n + 1/2) \neq 0$ implies $n = 6$. Thus, the periodicity is solved by calculating two extra coefficients, $c_n = K\left(\frac{5\theta}{2\pi} - n + 1/2\right)$ for $n \in \{0, 6\}$ and forming the modular channel vector $[c_1 + c_6,\ c_2,\ c_3,\ c_4,\ c_5 + c_0]^t$.

Channel coefficients are scaled with a factor $1/(x^2 + y^2)$ to maintain a constant weight of all basis functions, compensating for the polar coordinate system and the logarithm of the radial position. Note that the supported radial range is limited at both ends, avoiding an infinite channel density at the origin.

Fig. 9 Left: the log-polar channel encoded and decoded cameraman image. Right: the original image

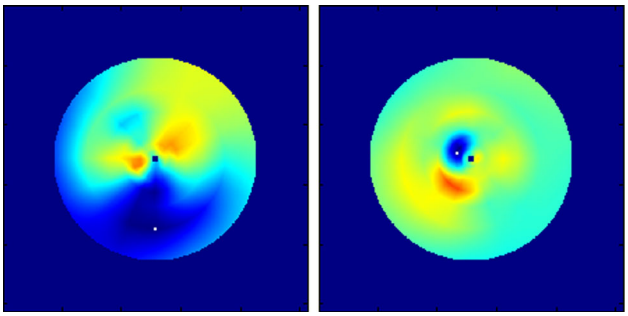

Fig. 10 Estimated difference between translated representations of one frame compared to the representation of the next frame, sampled on a log-polar grid and interpolated using log-polar channel basis functions. In the left case, the precise translation is uncertain; however, there is a strong indication that the tracked object has moved downwards in the image. In the right case, the precise translation is more certain. The white markers indicate global minima of the error function with respect to translations. Blue indicate the smallest differences and red the largest (Color figure online)

The channel arrangement is illustrated in Fig. 9, where the pixel coordinate system is centered in the middle of the image. The image is channel encoded, using log-polar channels for spatial position and regular channels for intensity. The encoded information is illustrated by a decoded image to the left. Note that the spatial resolution is reduced radially, however, intensity resolution and sharp edges are preserved. Since pixel positions are constant, position-dependent coefficients can be pre-calculated.

4.3 Visual Tracking

One application for the log-polar channel layout is visual tracking. The operation of moving the central position of the log-polar grid followed by re-encoding the image is approximated by a linear operation directly on the previous channel coefficients. Since the high-resolution area will be at a different part of the image after translation, where only lower resolution information is available in the previous representation, only an approximation of the representation is obtained.

For rotations of the image in increments of the channel spacing in the angular direction, the corresponding new channel coefficients are obtained through a circular shift of the old coefficients. Combining the rotation operation with a single translation operation, translations in all directions can be generated through combined rotation-shift-inverse-rotation operations.

With shift operations of different lengths, effects of operations in the 2D translation space can be sampled. By comparison with the representation of the next frame, translation information between the frames is obtained. This is illustrated in Fig. 10, where the errors after operations in the translation space are sampled in a log-polar grid and illustrated using log-polar channels. In this manner, more information regarding the local error surface is obtained.

5 Uniformization and Copula Estimation

Extending the idea of non-regular channel placement, channels should be placed depending on the data to be encoded, with high channel density where samples are likely. This can be obtained by mapping samples using the cumulative density function of the distribution from which the samples are drawn. Usually this function is not available, but using the ideas of density reconstruction from Sect. 3, a useful representation of the cumulative density function can be obtained and maintained online.

From Sect. 3 it is clear that for a set of reconstruction channel coefficients λ fulfilling the conjecture,

$$p(x) = \frac{1}{3}\sum_{n=1}^{N} \lambda_n K(x - \xi_n) \tag{40}$$

is a valid density function. Furthermore, the corresponding cumulative density function is

$$\begin{aligned} P(x) &= \int_{-\infty}^{x} \frac{1}{3}\sum_{n=1}^{N} \lambda_n K(y - \xi_n)\,\mathrm{d}y = \\ &= \frac{1}{3}\sum_{n=1}^{N} \lambda_n \int_{-\infty}^{x} K(y - \xi_n)\,\mathrm{d}y = \\ &= \frac{1}{3}\sum_{n=1}^{N} \lambda_n \hat{K}(x - \xi_n) \end{aligned} \tag{41}$$

with the (cumulative) basis functions $\hat{K}(x) = \int_{-\infty}^{x} K(y)\,\mathrm{d}y$. Only three (for three overlapping channels) cumulative basis functions are in the transition region for any given x, (41) can thus be calculated in constant time (independent of channel count N) as

$$P(x) = 0 + \frac{1}{3} \sum_{n=j-1}^{j+1} \lambda_n \hat{K}(x - \xi_n) + \frac{N - (j+1)}{N} \quad , \quad (42)$$

where j is the central channel activated by x. The function P maps values x to the range [0, 1].

The mapped values will be close to uniformly distributed (using the true cumulative density functions, the mapped values will be uniformly distributed). Placing a new set of regularly spaced channels in this transformed space, their distribution in the original space will be sample density dependent.

For multi-dimensional distributions, this can be used to estimate the Copula which clearly indicates dependencies between dimensions by removing the effect of the marginal distributions. This is obtained by estimating marginal densities using the approach of Sect. 3, where the estimation of $\mathbf{c}$ can be done incrementally. The reconstruction coefficients λ are updated by iterating (36) once after every new data point. The (density) Copula representation is obtained by encoding the mapped points using an outer product channel representation on the space $[0, 1] \times [0, 1]$. For independent random variables, the Copula is constant (one).

5.1 Experiments

A simple simulation result for the Copula density of correlated Gaussian distributions is given in Fig. 11.

Figure 12 illustrates the representation of one of the marginal distributions and the Copula estimation basis functions mapped through the inverse estimated marginal cumulative density function. Since the marginal distribution is smooth, the estimated densities follow the true densities closely. Figure 13 indicates the state of the estimate of the marginal distribution after 20 samples have been observed.

Copulas estimated from samples drawn from two different multivariate Gaussian distributions are shown in Fig. 14. The covariance matrices of these distributions are

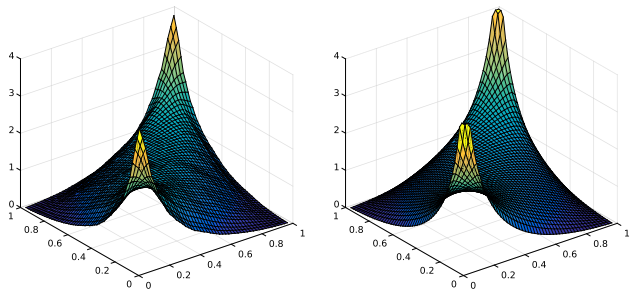

Fig. 11 Copula distribution for correlated Gaussian distributions. Left: simulation result using the known marginals. Right: simulation result using the estimated marginals

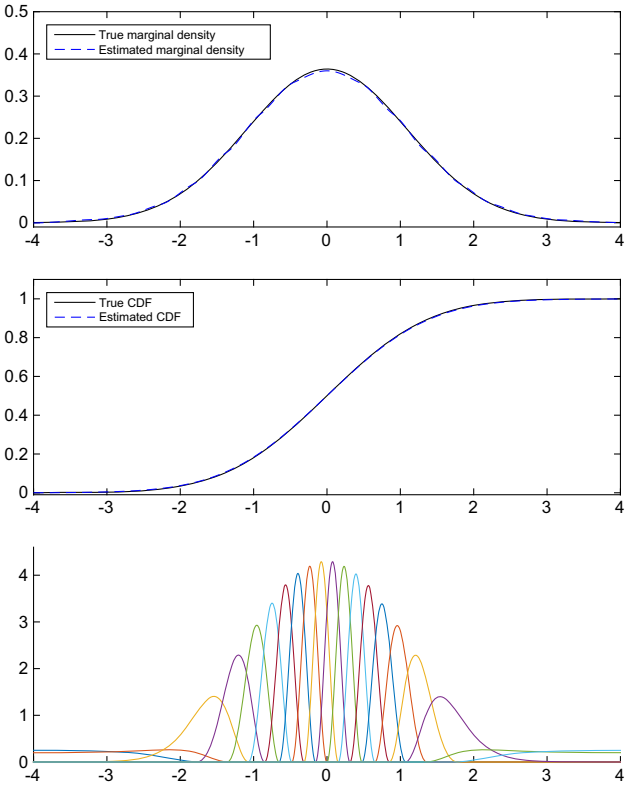

Fig. 12 Top and middle: marginal density functions estimated using incremental channel representations and maximum approximative entropy reconstruction, compared with the true marginal densities. Bottom: basis functions for Copula estimation. The basis functions are regularly spaced on [0, 1] and mapped through the inverse estimated CDF. When estimating the Copula, samples are instead mapped by the estimated CDF (Color figure online)

$$\Sigma_1 = \begin{pmatrix} 0.3 & 0.3 \\ 0.3 & 1.2 \end{pmatrix} \quad \text{and} \quad \Sigma_2 = \begin{pmatrix} 0.3 & 0 \\ 0 & 1.2 \end{pmatrix} \quad (43)$$

respectively. In these estimated Copulas, the first 100 samples were only used for estimating the marginals. The following samples were used both for updating the estimate of the marginals and for generating the Copula estimate. As apparent in the figures, the estimated Copula captures the dependency structure, and the independency in the latter case is clear.

6 Conclusion

Channel representations are descriptors for visual features, motivated from nonparametric statistics. Powerful visual features are fundamental requirements for applying machine learning techniques to computer vision problems, e.g., for learning path following [23] and visual tracking [22].

This work extends previous work on channel representations that often only addressed orientation estimation or

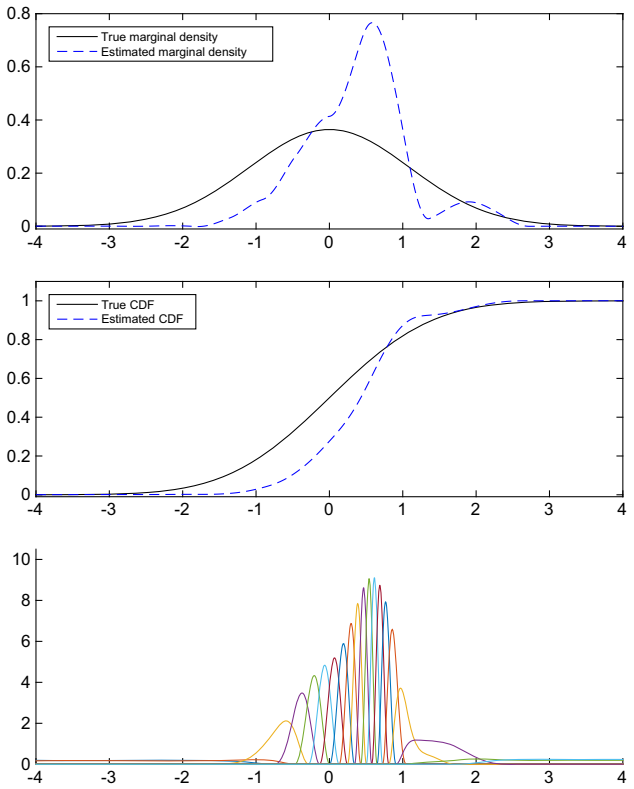

Fig. 13 Estimate of marginal density functions after observing 20 samples, compared with true functions. Bottom: basis functions for Copula estimation seen through the current estimate of the cumulative density function (Color figure online)

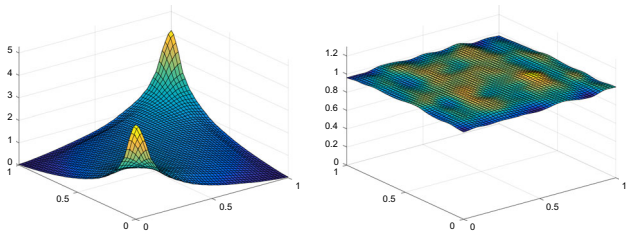

Fig. 14 Copulas estimated from multivariate Gaussian distributions. Left: covariance Σ_1 (dependent). Right: covariance Σ_1 (independent). See (43)

smoothing problems. We have presented a variety of approximative channel-based algorithms for probabilistic problems: a novel efficient algorithm for density reconstruction, a novel and efficient scheme for nonlinear gridding of densities, and finally a novel method for estimating Copula densities.

The proposed algorithms have been evaluated, and the experimental results provide evidence that by relaxing the requirements for exact solutions, efficient algorithms are obtained while retaining low approximation errors.

The incorporation of the proposed methods into existing learning systems, such as [21], and into new systems remains future work. With the novel algorithms at hand, possibly new problems can be approached or at least known problems can be approached in novel ways.

Acknowledgements The authors express their gratitude to Ulrich Köthe for discussions on the topic and in particular for proposing the use of the Copula. This research was partly funded by the Swedish Research Council through a framework grant for the project Energy Minimization for Computational Cameras (2014-6227), by SSF grant RIT 15-0097 SymbiCloud, and by the excellence center ELLIIT.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2005, vol. 1, pp. 886–893 (2005). https://doi.org/10.1109/CVPR.2005.177
2. Duits, R., Franken, E.: Left-invariant parabolic evolutions on SE(2) and contour enhancement via invertible orientation scores part II: nonlinear left-invariant diffusions on invertible orientation scores. Q. Appl. Math. **68**(2), 293–331 (2010). https://doi.org/10.1090/S0033-569X-10-01173-3
3. Felsberg, M.: Incremental computation of feature hierarchies. In: Pattern Recognition, *Lecture Notes in Computer Science*, vol. 6376, pp. 523–532. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-15986-2_53
4. Felsberg, M.: Enhanced distribution field tracking using channel representations. In: IEEE ICCV Workshop On Visual Object Tracking Challenge (2013)
5. Felsberg, M., Forssén, P.E., Scharr, H.: Channel smoothing: efficient robust smoothing of low-level signal features. IEEE Trans. Pattern Anal. Mach. Intell. **28**(2), 209–222 (2006)
6. Felsberg, M., Granlund, G.: Anisotropic channel filtering. In: Proceedings of 13th Scandinavian Conference on Image Analysis, LNCS 2749, pp. 755–762 (2003)
7. Felsberg, M., Hedborg, J.: Real-time view-based pose recognition and interpolation for tracking initialization. J. Real Time Image Process. **2**(2–3), 103–116 (2007)
8. Felsberg, M., Öfjäll, K., Lenz, R.: Unbiased decoding of biologically motivated visual feature descriptors. Front. Robot. AI **2**, 20 (2015). https://doi.org/10.3389/frobt.2015.00020
9. Ferraro, M., Caelli, T.M.: Lie transformation groups, integral transforms, and invariant pattern recognition. Spat. Vis. **8**(4), 33–44 (1994)
10. Forssén, P.E.: Low and medium level vision using channel representations. Ph.D. Thesis, Linköping University, Sweden (2004)
11. 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 (2000)
12. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers, Dordrecht (1995)
13. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image Vision Comput. **28**(5), 807–813 (2010). https://doi.org/10.1016/j.imavis.2009.08.002. **(Best of Automatic Face and Gesture Recognition 2008)**

14. Huang, D., Cabral, R.S., De la Torre, F.: Robust regression. In: European Conference on Computer Vision (ECCV) (2012)
15. Jonsson, E., Felsberg, M.: Reconstruction of probability density functions from channel representations. In: Proceedings of 14th Scandinavian Conference on Image Analysis (2005)
16. Jonsson, E., Felsberg, M.: Efficient computation of channel-coded feature maps through piecewise polynomials. Image Vis. Comput. **27**(11), 1688–1694 (2009). https://doi.org/10.1016/j.imavis.2008.11.002
17. Kass, M., Solomon, J.: Smoothed local histogram filters. In: ACM SIGGRAPH 2010 Papers, SIGGRAPH '10, pp. 100:1–100:10. ACM, New York, NY, USA (2010). https://doi.org/10.1145/1833349.1778837
18. LeCun,Y., Bengio,Y.: Convolutional networks for images, speech, and time series. Handb. Brain Theory Neural Netw., pp 255–258 (1995)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
20. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
21. Öfjäll, K., Felsberg, M.: Biologically inspired online learning of visual autonomous driving. In: Proceedings of the British Machine Vision Conference. BMVA Press (2014)
22. Öfjäll, K., Felsberg, M.: Weighted update and comparison for channel-based distribution field tracking. In: ECCV 2014 Workshops, *Lecture Notes in Computer Science*, vol. 8926, pp. 218–231. Springer (2015). https://doi.org/10.1007/978-3-319-16181-5_15
23. Öfjäll, K., Felsberg, M., Robinson, A.: Visual autonomous road following by symbiotic online learning. In: 2016 IEEE Intelligent Vehicles Symposium Proceedings (2016)
24. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: European Conference on Computer Vision (2006)
25. Pouget, A., Dayan, P., Zemel, R.: Information processing with population codes. Nat. Rev. Neurosci. **1**, 125–132 (2000)
26. Scott, D.W.: Averaged shifted histograms: effective nonparametric density estimators in several dimensions. Ann. Stat. **13**(3), 1024–1040 (1985)
27. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: IEEE Computer Vision and Pattern Recognition (2012)
28. Sharma, U., Duits, R.: Left-invariant evolutions of wavelet transforms on the similitude group. Appl. Comput. Harmonic Anal. **39**(1), 110–137 (2015). https://doi.org/10.1016/j.acha.2014.09.001
29. Snippe, H.P., Koenderink, J.J.: Discrimination thresholds for channel-coded systems. Biol. Cybern. **66**, 543–551 (1992)
30. Vondrick, C., Khosla, A., Malisiewicz, T., Torralba, A.: HOGgles: visualizing object detection features. In: ICCV (2013)
31. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV 2014, *Lecture Notes in Computer Science*, vol. 8689, pp. 818–833. Springer (2014). https://doi.org/10.1007/978-3-319-10590-1_53
32. Zemel, R.S., Dayan, P., Pouget, A.: Probabilistic interpretation of population codes. Neural Comput. **10**(2), 403–430 (1998)

Kristoffer Öfjäll received the Ph.D. degree in engineering from Linköping University, Sweden, in 2016. He is now at Visionists AB, Gothenburg, Sweden. His current research interests include computer vision, machine learning, and robotics.

Michael Felsberg received the Ph.D. degree in engineering from the University of Kiel, Kiel, Germany, in 2002. Since 2008, he has been a Full Professor and the Head of the Computer Vision Laboratory, Linköping University, Sweden. His current research interests include signal processing methods for image analysis, computer and robot vision, and machine learning. He has published more than 150 reviewed conference papers, journal articles, and book contributions. He was a recipient of awards from the German Pattern Recognition Society in 2000, 2004, and 2005, from the Swedish Society for Automated Image Analysis in 2007 and 2010, from Conference on Information Fusion in 2011 (Honorable Mention), from the CVPR Workshop on Mobile Vision 2014, and from the ICPR 2016 track on Computer Vision (best paper). He has achieved top ranks on various challenges (VOT: 3rd 2013, 1st 2014, 2nd 2015, 1st 2016, 1st 2017 (sequestered test); VOT-TIR: 1st 2015, 1st 2016, 3rd 2017; OpenCV Tracking: 1st 2015; KITTI Stereo Odometry: 1st 2015, March). He has coordinated the EU projects COSPAL and DIPLECS, he is an Associate Editor of the Journal of Mathematical Imaging and Vision and the Journal of Image and Vision Computing. He was Publication Chair of the International Conference on Pattern Recognition 2014 and Track Chair 2016, he is VOT-committee member since 2015, he was the General Co-Chair of the DAGM symposium in 2011, General Chair of CAIP 2017, and will be General Co-Chair of SCIA 2019.