Incremental computation of feature hierarchies

Michael Felsberg

Original Publication:

Michael Felsberg, Incremental computation of feature hierarchies, 2010, Pattern Recognition, Lecture Notes in Computer Science.

Incremental Computation of Feature Hierarchies

Michael Felsberg

Linköping University, Department of Electrical Engineering, Sweden
mfe@isy.liu.se

Abstract. Feature hierarchies are essential to many visual object recognition systems and are well motivated by observations in biological systems. The present paper proposes an algorithm to incrementally compute feature hierarchies. The features are represented as estimated densities, using a variant of local soft histograms. The kernel functions used for this estimation in conjunction with their unitary extension establish a tight frame and results from framelet theory apply. Traversing the feature hierarchy requires resampling of the spatial and the feature bins. For the resampling, we derive a multi-resolution scheme for quadratic spline kernels and we derive an optimization algorithm for the upsampling. We complement the theoretic results by some illustrative experiments, consideration of convergence rate and computational efficiency.

1 Introduction

The computation of feature hierarchies is an essential step in many state-of-the-art visual object recognition systems. The hierarchical processing is well motivated by observations in biological systems [1]. The present paper proposes an algorithm for incremental computation of feature hierarchies.

Feature hierarchies can be build with respect to abstraction levels or resolution [2], p. 8–9, or a combination of both [3, 4]. Here, we focus on the resolution of soft histograms as used in, e.g., [5], where matching is performed on histograms with increasing resolution, i.e., coarse to fine. In this work, increasing spatial resolution goes in hand with decreasing resolution in feature space. This is plausible from a practical (computational effort) and statistical (significance) point of view, and the reciprocal relation of resolution in the spatial and the feature domain has an theoretical upper bound [6].

Computations in the joint spatio-featural space require a common framework for spatial positions and generic features. Biological systems have been observed to use a representation called population codes [7], in earlier work also called channel codes [8, 9]. Channel representations as a computational framework, e.g. for object recognition, have been introduced in [10], and are directly related to kernel density estimation [11]. Channel representations of features are basically soft-histograms or Parzen estimators with a smooth kernel function. They are beneficial in many tasks due to their robustness [11, 23].

Applying channel representations to spatial coordinates results in low-pass filters or point-spread functions. Subsampled low-pass filters give rise to resolution

pyramids [12, 13] and multi-resolution analysis. The complementing high-pass filters became very popular in terms of wavelets [14], where signals are decomposed based on a orthonormal basis. The main drawback of discrete wavelets is that if the scaling function (low-pass filter) is smooth, the wavelet becomes highly non-smooth [15]. In contrast to wavelets, framelets can be selected to be smooth, which is beneficial for stability, and contain redundancy, which improves robustness. Similar to wavelets, framelets are compact, which results in limited interaction and thus sparseness and efficiency in computations, e.g. exploited for super-resolution still image extraction using C^0 framelets [16].
In this paper, we extend the existing work in two ways: a) We derive a multi-resolution scheme for quadratic spline channel representations (C^1 kernels) using frame theory. b) We derive an algorithm for incrementally compute spatio-featural hierarchies. Since we consider linear and periodic features, any feature represented as a combination of linear and periodic parameters is covered.

The paper is structured as follows. The second section on methods gives the required background and explains the proposed novel methods. In the subsequent section on experiments and results we explain the performed tests, discuss the results, and analyze the computational complexity. We conclude the paper with a summary of achieved results.

2 Methods

2.1 Channel Representations and CCFMs

The channel representation as a computational framework goes back to [10]. In channel representations features are represented by weights assigned to ranges of feature values, similar to histograms but exploiting smooth bins. The closer the current feature value ξ to the respective feature interval center n, the higher the channel weight f_n (for an example, the reader might refer to Section 3.3):

$$f_n(\xi) = k(\xi - n) \qquad n \in \mathbb{N} \ , \tag{1}$$

where $k(\cdot)$ is a symmetric, unimodal kernel function and where ξ has been scaled such that it has a suitable range (note that the channel centers are integers).

In what follows, we have been using quadratic B-splines as kernel function, since they are smooth and easy to formulate in the z-domain [11]:

$$B_2(\xi) = \begin{cases} 3/4 - \xi^2 & |\xi| \leq 1/2 \\ (|\xi| - 3/2)^2/2 & 1/2 < |\xi| < 3/2 \\ 0 & |\xi| \geq 3/2 \end{cases} \tag{2}$$

Comparing (1) with a kernel density estimator, the only difference is that the kernel function is placed at equidistant positions and not on the samples drawn from the distribution. Since the kernel is symmetric, the estimated coefficient at the discrete position is the same in both cases and the distribution of the stochastic variable ξ, p_ξ, is approximated by f_n in expectation sense [11]:

$$E_\xi\{f_n(\xi)\} = (p_\xi * k)(n) = (p_\xi * B_2)(n) \ . \tag{3}$$

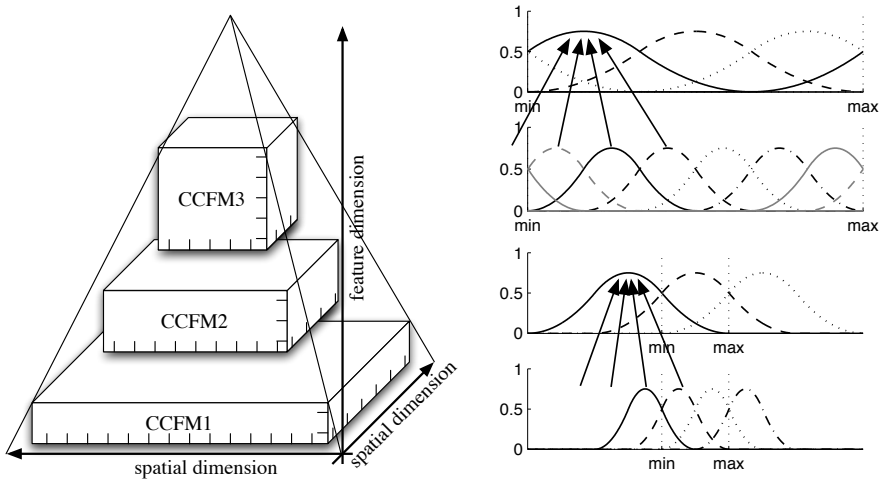

Fig. 1. Left: Going upwards in the spatio-featural pyramid reduces spatial resolution and increases feature resolution. Top right: Downsampling for periodic boundary conditions. The kernel functions are re-entering the domain from the respective opposite side, i.e., the left most connection refers to the solid grey kernel. Bottom right: Downsampling for the zero padded case. The connections outside the existing channels get zero weight. The four connections have the weights 1/8, 3/8, 3/8, 1/8 in both cases.

A multi-dimensional channel representation for a set of features is formed by taking the Cartesian product of the respective one-dimensional representations. If the spatial coordinates are contained in the feature set, a channel-coded feature map (CCFM) [17] is generated by local averaging, representing the spatio-featural density. For instance one might consider local orientation, hue, and saturation as local image features, resulting in a 5D CCFM. For practical (computational effort) and statistical (significance) reasons, the number of spatial and featural channels is not independent, but should be chosen reciprocally [6].

As motivated in the introduction, many object recognition approaches require a hierarchical spatio-featural representation, e.g., a pyramid of CCFMs. As an example, let us consider an orientation-based CCFM scale-space. At the finest spatial resolution, we have a minimum number of four orientation channels, representing e.g. positive and negative responses of a Gaussian derivative. On the second level, we obtain eight orientation channels, which, if combined with four by four spatial channels, yields a structure similar to the SIFT descriptor [18], although with quadratic histogram bins instead of linear ones.

Single level descriptors like SIFT are directly computed from filter outputs, but if several different levels are to be considered, a staged process that builds the pyramid successively is beneficial. This means that higher-resolution feature density estimates are estimated from several lower-resolution estimates when traversing the pyramid towards lower spatial resolutions, c.f. Fig.1 left.

2.2 Multi-Resolution Analysis of Density Estimates

The first step to find a staged computation of the CCFM pyramid is to derive the scaling function for quadratic B-splines (2). According to [19] it is given as

$$m_0 = \frac{1}{8}[1,\, 3,\, 3,\, 1] \ . \tag{4}$$

The scaling function allows to compute the scaled quadratic B-spline channel directly from the unscaled B-spline channel coefficients. This is easily verified either in the z-domain or by elementary calculus using (2).

If we combine (4) with a downsampling scheme, we compute new channel values between existing channels and leave out every other channel. This process is illustrated in Fig. 1, right. For periodic features (periodic boundary conditions), the corresponding matrix operator (subscript p) is a circular matrix with every other row left out. For linear features, the channel vector is zero padded, cf. Fig. 1, bottom right. The corresponding matrix operator (subscript l) is a Toeplitz matrix with every other row left out.

$$\mathbf{T}_{p,0} = \frac{1}{8}\begin{pmatrix} 3 & 3 & 1 & 0 & \dots & 0 & 1 \\ 0 & 1 & 3 & 3 & 1 & 0 & \dots\, 0 \\ \vdots & & \ddots & & & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 3 & 3 & 1\ 0 \\ 1 & 0 & \dots & 0 & 1 & 3 & 3 \end{pmatrix} \quad \mathbf{T}_{l,0} = \frac{1}{8}\begin{pmatrix} 3 & 1 & 0 & \dots & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 & \dots\ 0 \\ \ddots & & \ddots & & \ddots & \ddots \\ 0 & \dots & 0 & 1 & 3 & 3 & 1 \\ 0 & 0 & \dots & 0 & 1 & 3 & 3 \end{pmatrix} \tag{5}$$

The subscript p respectively l is omitted if it is obvious from the context.

Let $\mathbf{f}$ denote a channel vector. Assume further that $\mathbf{g}$ is a low-pass filtered (with m_0) and downsampled copy of $\mathbf{f}$. In matrix notation we obtain

$$\mathbf{g} = 2\mathbf{T}_0\mathbf{f} \ , \tag{6}$$

where the factor 2 is required to keep the channel vector $\mathbf{g}$ normalized.

We form tight frame filters by applying the unitary extension principle [20] resulting in the two high-pass filters [21]

$$m_1 = \frac{1}{8}[1,\, 3,\, -3,\, -1] \qquad m_2 = -\frac{\sqrt{3}}{4}[1,\, -1,\, 0,\, 0] \ . \tag{7}$$

The matrix operators corresponding to m_1 read

$$\mathbf{T}_{p,1} = \frac{1}{8}\begin{pmatrix} -3 & 3 & 1 & 0 & \dots & \dots & 0 & -1 \\ 0 & -1 & -3 & 3 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & & \ddots & & \vdots \\ 0 & \dots & 0 & -1 & -3 & 3 & 1 & 0 \\ 1 & 0 & \dots & \dots & 0 & -1 & -3 & 3 \end{pmatrix} \quad \mathbf{T}_{l,1} = \frac{1}{8}\begin{pmatrix} 3 & 1 & 0 & \dots & \dots & 0 & 0 \\ -1 & -3 & 3 & 1 & 0 & \dots & 0 \\ \ddots & & \ddots & & \ddots & & \ddots \\ 0 & \dots & 0 & -1 & -3 & 3 & 1 \\ 0 & 0 & \dots & \dots & 0 & -1 & -3 \end{pmatrix}$$

and the matrix operators corresponding to m_2 are formed accordingly. By basic calculations, we verify the reconstruction formula

$$\frac{1}{2}\mathbf{I} = \sum_{j=0}^{2} \mathbf{T}_j^T\mathbf{T}_j \ , \tag{8}$$

where $\mathbf{I}$ is the identity and the factor $\frac{1}{2}$ originates from the downsampling.

2.3 Upsampling Channel Vectors

If we traverse the spatio-featural hierarchy upwards, we downsample spatially after[1] having upsampled the feature dimensions. The downsampling is fully covered in terms of the matrices (5). The upsampling of $\mathbf{g}$ is achieved by combining the reconstruction formula (4.13) in [21] with the iterative scheme in [20]. Plugging (6) into (8) results in

$$\frac{1}{2}\mathbf{f} = \sum_{j=0}^{2} \mathbf{T}_j^T \mathbf{T}_j \mathbf{f} = \frac{1}{2}\mathbf{T}_0^T \mathbf{g} + \mathbf{T}_1^T \mathbf{T}_1 \mathbf{f} + \mathbf{T}_2^T \mathbf{T}_2 \mathbf{f} \ . \tag{9}$$

Using an explicit approach, we obtain the iterative scheme

$$\mathbf{f}^{(k+1)} = \mathbf{T}_0^T \mathbf{g} + 2(\mathbf{T}_1^T \mathbf{T}_1 + \mathbf{T}_2^T \mathbf{T}_2)\mathbf{f}^{(k)} \ . \tag{10}$$

Iterating this equation results in the solution of the underdetermined problem

$$\min_{\mathbf{f}} \|\mathbf{g} - 2\mathbf{T}_0 \mathbf{f}\|_2^2 \ . \tag{11}$$

Unrolling the iteration (10), we obtain

$$\mathbf{f}^{(k+1)} = \Delta^{k+1}\mathbf{f}^{(0)} + (\Delta^k + \ldots + \Delta + \mathbf{I})\mathbf{T}_0^T \mathbf{g} \tag{12}$$

where $\Delta = 2(\mathbf{T}_1^T \mathbf{T}_1 + \mathbf{T}_2^T \mathbf{T}_2)$. The M rows of $\mathbf{T}_0$ are linearly independent and span an M-d space. Let $\mathbf{P}_M$ denote the $N \times N$ projection matrix onto this M-dimensional subspace (non-zero eigenvectors of $\mathbf{T}_0^T \mathbf{T}_0$). We obtain

$$\begin{aligned} \mathbf{f}^{(k+1)} &= \Delta^{k+1}\mathbf{f}^{(0)} + (\Delta^k \mathbf{P}_M + \ldots + \Delta \mathbf{P}_M + \mathbf{I})\mathbf{T}_0^T \mathbf{g} \\ &= \Delta^{k+1}\mathbf{f}^{(0)} + ((\Delta \mathbf{P}_M)^k + \ldots + \Delta \mathbf{P}_M + \mathbf{I})\mathbf{T}_0^T \mathbf{g} \end{aligned}$$

because $\mathbf{P}_M = \mathbf{P}_M^2$ and $\mathbf{P}_M$ commutes with $\Delta = \mathbf{I} - 2\mathbf{T}_0^T \mathbf{T}_0$. In the limit, we obtain

$$\mathbf{f}^{(\infty)} = (\mathbf{I} - \mathbf{P}_M)\mathbf{f}^{(0)} + (\mathbf{I} - \Delta \mathbf{P}_M)^{-1}\mathbf{T}_0^T \mathbf{g} \tag{13}$$

because $\Delta \mathbf{P}_M$ has a spectral radius smaller than one.

This means that the N-dimensional solution $\mathbf{f}$ of our iteration is determined by an M-dimensional constraint given in terms of $\mathbf{g}$. The remaining $N - M$ dimensions, i.e., the null-space of $\mathbf{P}_M$, is determined to have norm zero if we start from $\mathbf{f}^{(0)} = 0$, i.e., we obtain the minimum norm solution of (11). In our particular problem, however, we are not interested in the minimum norm solution, but we require to have a non-negative solution instead, since kernel density estimates are non-negative. Hence, we should choose $\mathbf{f}^{(0)}$ such that $\mathbf{f}^{(\infty)}$ is non-negative everywhere.

[1] If we downsampled first, we would lose information.

For the periodic case, we know that $N = 2M$ and for the linear case, we obtain $N = 2M - 2$ since all channels are doubled except for those at the boundaries. Hence, we need $2M - M = M$ respectively $2M - 2 - M = M - 2$ equations that are not linearly dependent on the rows of $\mathbf{T}_0$. These can be obtained in terms of $\mathbf{I} - \mathbf{P}_M$, but we can easily derive another set of vectors spanning the null-space of $\mathbf{T}_0$. Define the two matrices

$$\mathbf{S}_p = \frac{1}{8}\begin{pmatrix} -3 & 3 & -1 & 0 & \dots & \dots & 0 & 1 \\ 0 & 1 & -3 & 3 & -1 & 0 & \dots & 0 \\ \vdots & & \ddots & & & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & -3 & 3 & -1 & 0 \\ -1 & 0 & \dots & \dots & 0 & 1 & -3 & 3 \end{pmatrix} \quad \mathbf{S}_l = \frac{1}{8}\begin{pmatrix} 1 & -3 & 3 & -1 & 0 & \dots & 0 \\ \ddots & & \ddots & & \ddots & & \ddots \\ 0 & \dots & 0 & 1 & -3 & 3 & -1 \end{pmatrix} .$$

We verify that

$$\mathbf{S}_p \mathbf{T}_{p,0}^T = 0 \quad \text{and} \quad \mathbf{S}_l \mathbf{T}_{l,0}^T = 0 \tag{14}$$

i.e., $\mathbf{S}_p$ ($\mathbf{S}_l$) is in the null-space of $\mathbf{T}_{p,0}$ ($\mathbf{T}_{l,0}$). Furthermore, since $\mathbf{S}_p$ is of rank M and $\mathbf{S}_l$ is of rank $M - 2$, we conclude that they span the null-space of $\mathbf{T}_{p,0}$ respectively $\mathbf{T}_{l,0}$.

In order to obtain a solution according to (11) with non-negative coefficients, a straightforward idea is to simply set all negative coefficients to zero in each iteration of (10). However, this does not lead to stable results in our experiments. Instead, we compute the projection of the negative coefficients onto the null-space. For this purpose, we define the vector $\mathbf{f}_{\text{neg}}$ component-wise

$$f_{\text{neg},n} = \begin{cases} f_n & f_n < 0 \\ 0 & f_n \geq 0 \end{cases} \qquad n = 1, \dots, N \ . \tag{15}$$

This vector is then projected onto the null-space ($\cdot^\dagger$ denotes the pseudoinverse)

$$\mathbf{f}_{\text{null}} = \mathbf{S}^T {\mathbf{S}^T}^\dagger \mathbf{f}_{\text{neg}} \ . \tag{16}$$

Subtracting this vector from the current solution brings us closer to the non-negative solution without leaving our solution space, but we need to determine the step-length λ. A greedy approach is to use the ratio of the largest negative value of $\mathbf{f}_{\text{neg}}$ and the corresponding coefficient of $\mathbf{f}_{\text{null}}$:

$$n_0 = \arg\min_n f_{\text{neg},n} \qquad \lambda = f_{\text{neg},n_0} / f_{\text{null},n_0} \ . \tag{17}$$

To achieve numerical stability, the coefficient λ must be bounded in the positive and negative range, e.g., by requiring $|f_{\text{null},n_0}| > 10^{-5}$. Finally, we update

$$\mathbf{f}^{(k+1)} \Leftarrow \mathbf{f}^{(k+1)} - \lambda \mathbf{f}_{\text{null}} \ . \tag{18}$$

3 Experiments

We have applied our algorithm (10,15-18) in three experiments: Convergence tests, image reconstruction from upsampled densities, and orientation pyramids.

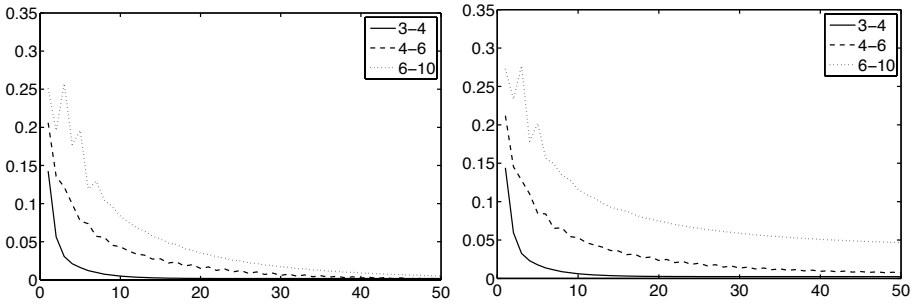

Fig. 2. Convergence of our algorithm: Hellinger distance plotted versus the number of iterations. Left: no noise. Right: Gaussian noise (5% standard deviation).

3.1 Convergence Behavior

For analyzing the convergence behavior of our upsampling algorithm, we have generated sets of samples from a linear ramp function. One data set consists of the function values without noise and the second contains 5% Gaussian noise. The samples have been collected in soft histograms (channels vectors) with 3, 4, and 6 bins respectively. These vectors have then been upsampled to 4, 6, and 10 bins and compared to the directly encoded noise-free data $\mathbf{h}$, using the Hellinger distance (which is often used in kernel-based matching [22]

$$d^2(\mathbf{f},\mathbf{h}) = \frac{1}{2}\sum_n (\sqrt{f_n} - \sqrt{h_n})^2 = 1 - \sum_n \sqrt{f_n h_n} \tag{19}$$

where the right hand-side is obtained since the coefficients of $\mathbf{f}$ and $\mathbf{h}$ sum to one. The right-most term is called Bhattacharyya coefficient [22].

The plots in Fig. 2 show the Hellinger distance as a function of the number of iterations. We can clearly see that the more iterations are required the more channels are to be reconstructed. On the other hand, achieving the same Hellinger distance for a larger number of bins means to achieve a higher accuracy per bin, and thus more information needs to be recovered. If the number of iterations is normalized with the number of bins, convergence speed is about the same.

One conclusion of this observation is that if the upsampling of feature distributions is combined with a downsampling in the 2D spatial domain, the algorithm has constant computational complexity, independent of the actual level in the pyramid. This complexity is linear in the number of pixels and the number of initial channels. This has been confirmed by the observed runtimes, which were all in the sub-second range (Matlab implementation).

Another observation we make in Fig. 2 is that convergence speed seems to be unaffected by noise, but the final error level depends on the noise. Since the reconstruction is compared against the noise-free density estimates, the fixed Gaussian noise has growing influence for decreasing kernel widths.

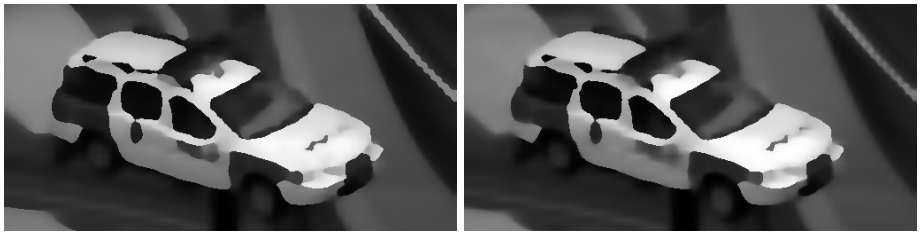

Fig. 3. A car image from [23] reconstructed from CCFMs, clipped to 512×256. Left: reconstruction from direct CCFM encoding. Right: reconstruction from successive encoding using the proposed algorithm.

3.2 Image Reconstruction

In this experiment we visualize the similarities and differences of direct encoding of CCFMs and successiv encoding using the proposed algorithm. The images are taken from [23], size 512×512. We have reproduced the experiment from [6], generating CCFMs with a resolution of $66 \times 66 \times 10$ channels, and reconstructing images with a resolution of 512×512 using the method from [6]. The encoding has been done in two different ways: a) by directly encoding the CCFM at the final resolution and b) by encoding into three channels point-wise and subsequent three-fold upsampling of the feature (greyscale) channel and downsampling of the spatial channels, resulting in the sequence $514 \times 514 \times 3 \rightarrow 258 \times 258 \times 4 \rightarrow 130 \times 130 \times 6 \rightarrow 66 \times 66 \times 10$. The two reconstructions for one example (an image of a police car) are depicted in Fig. 3. In the ideal case, the two images should be identical; the absolute reconstruction quality is not of relevance here.

Up to minor differences, the two reconstructions are identical. The main differences occur at some few edges of medium greyscale difference, which are more crispy in the direct encoding case. The reason for this minor difference is presumably that some information loss is caused by the fact that spatial downsampling is performed in 2D while the feature upsampling is only 1D. However, the level at which edges are blurred is far beyond the original scale of channels. Using three channels to encode the whole range is identical to linear smoothing of the image, i.e., the proposed algorithm has succeeded in recovering robust smoothing from initially uni-modal representations. In addition to that, the striking advantage of the new method is that all intermediate CCFMs are also available, whereas the direct method needs to encode from scratch for any change of resolution.

3.3 Illustration of Orientation Density Estimates

This experiment illustrates the spatio-featural hierarchy for orientation. We have chosen a siemens star as input since its orientation channels are easy to analyze visually. The orientation channels respond on respective antipodal sides and the bow length corresponds to the width of the kernel, see Fig. 4. The first orientation channel (out of four) at level $n = 1$ responds with a quite large variance.

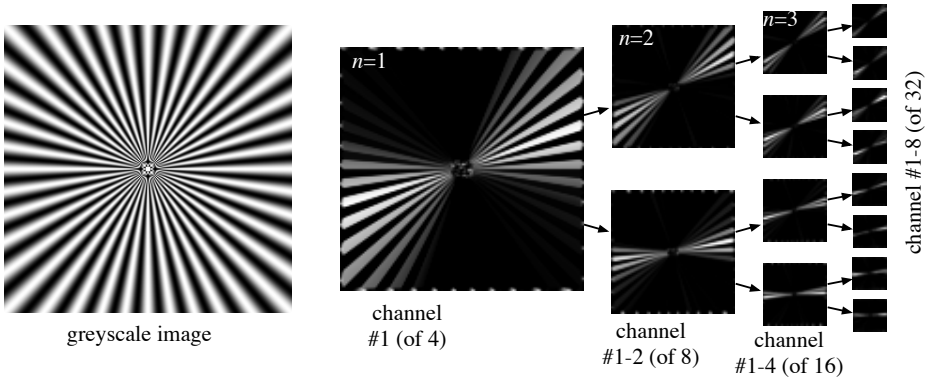

Fig. 4. Orientation pyramid. From left to right: image and hierarchy levels $n = 1, \ldots, 4$.

Running our upsampling algorithm results in eight channels and the first two are depicted, together covering about the same range as the previous one. After another run, we obtain 16 channels, of which we show the first four. Finally, we obtain 32 channels of which we show eight.

This illustrates how we obtain increasingly distinct estimates of the orientation by gathering information from a growing spatial support. We can repeat the procedure until we reach a global histogram. The features are not at all restricted to either greyscale or orientation. Any (combination) of features including color, texture, depth, motion, etc. is possible, but more difficult to illustrate.

4 Conclusion

We have presented a framelet-theory based framework for resampling channel-based density estimates up and down. We have further derived an optimization algorithm that produces the required high-pass components for upsampling density estimates with a non-negativity constraint. The algorithm is stable and converges rapidly towards the correct solution. The presented framework is well suited to generate feature hierarchies. Such hierarchies can be traversed up and down, allowing for bottom-up driven detection and top-down driven priming and adaptation of lower levels by simply multiplying prior distributions to the higher-level representations.

In some illustrative experiments, we have shown that the derived framework can be applied as intended. The optimization converges rapidly and is efficient to compute. The greyscale reconstruction results are as close to the direct encoding as it can be expected. A periodic feature (orientation) has been illustrated on four levels of the hierarchy. What remains for future work is to apply the new framework in an object recognition system and to verify its benefits on a more applied level.

Acknowledgements

This research has received funding from the EC's 7th Framework Programme (FP7/2007-2013), grant agreements 215078 (DIPLECS) and 247947 (GARNICS).

References

1. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience **2**(11) (November 1999) 1019–1025
2. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers, Dordrecht (1995)
3. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: Computer Vision and Pattern Recognition. (2005) 994–1000
4. Mutch, J., Lowe, D.G.: Multiclass object recognition with sparse, localized features. In: Computer Vision and Pattern Recognition. (2006) 11–18
5. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: Intern. Conf. Computer Vision. (2005) 1458–1465
6. Felsberg, M.: Spatio-featural scale-space. In: Intern. Conf. Scale Space Methods and Variational Methods in Computer Vision. Volume 5567 of LNCS. (2009)
7. Pouget, A., Dayan, P., Zemel, R.: Information processing with population codes. Nature Reviews – Neuroscience **1** (2000) 125–132
8. Snippe, H.P., Koenderink, J.J.: Discrimination thresholds for channel-coded systems. Biological Cybernetics **66** (1992) 543–551
9. Howard, I.P., Rogers, B.J.: Binocular Vision and Stereopsis. OUP, Oxford (1995)
10. Granlund, G.H.: An associative perception-action structure using a localized space variant information representation. In: Proc. AFPAC, Springer, Heidelberg (2000)
11. Felsberg, M., Forssén, P.E., Scharr, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. IEEE PAMI **28**(2) (2006) 209–222
12. Granlund, G.H.: In search of a general picture processing operator. Computer Graphics and Image Processing **8** (1978) 155–173
13. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Trans. Communications **31**(4) (1983) 532–540
14. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Analysis Machine Intelligence **11** (1989) 674–693
15. Daubechies, I.: Orthonormal bases of compactly supported wavelets. Communications on Pure and Applied Mathematics **41**(7) (1988) 909–996
16. Chan, R.H., Shen, Z., Xia, T.: A framelet algorithm for enhancing video stills. Applied and Computational Harmonic Analysis **23**(2) (2007) 153 – 170
17. Jonsson, E., Felsberg, M.: Efficient computation of channel-coded feature maps through piecewise polynomials. Image and Vision Comp. **27**(11) (2009) 1688–1694
18. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
19. Petukhov, A.: Symmetric framelets. Constr. Approx **19** (2000) 309–328
20. Chan, R.H., Riemenschneider, S.D., Shen, L., Shen, Z.: Tight frame: an efficient way for high-resolution image reconstruction. ACHA **17**(1) (2004) 91 – 115
21. Chui, C.K., He, W.: Compactly supported tight frames associated with refinable functions,. Applied and Computational Harmonic Analysis **8**(3) (2000) 293 – 319
22. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Thomson-Engineering (2007)
23. Vikstén, F., Forssén, P.E., Johansson, B., Moe, A.: Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In: IEEE ICRA. (2009)