Accurate and Efficient Approximation of the Continuous Gaussian Scale-Space

Ullrich Köthe

Cognitive Systems Group, University of Hamburg, Germany
koethe@informatik.uni-hamburg.de

Abstract. The Gaussian scale-space is a standard tool in image analysis. While continuous in theory, it is generally realized with fixed regular grids in practice. This prevents the use of algorithms which require continuous and differentiable data and adaptive step size control, such as numerical path following. We propose an efficient continuous approximation of the Gaussian scale-space that removes this restriction and opens up new ways to subpixel feature detection and scale adaptation.

1 Introduction

Smoothing with Gaussian functions and the Gaussian scale-space have become standard tools in low-level image analysis. They are routinely used for preprocessing, estimation of derivatives, and feature extraction. With few exceptions, theories about scale-space and scale-based feature detection are derived for continuous, differentiable functions, but are then realized on discrete grids, e.g. by sampling the Gaussian kernel or replacing it with a discrete approximation (e.g. binomial filters, Lindeberg's discrete analog [7], or recursive filters [4]). To save memory and time, images are often subsampled after a certain amount of smoothing as in a Gaussian pyramid [2] or hybrid pyramid [8]. These approaches always use grids whose sampling density is at most that of the original image. However, in [6] it was shown that a higher sampling density can be necessary in order to prevent information loss during image processing. Empirical evidence for improved feature detection on oversampled data was also reported by [10,9].

In this paper, we approach the sampling issue in a radical way: instead of working on a discrete representation, we propose an abstract data type that represents the Gaussian scale-space as a function over the reals, i.e. as a continuous, differentiable mapping from $\mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$, with given precision ε. Algorithms can access this data structure at arbitrary coordinates, and the requested function values or derivatives are computed *on demand*. Even for very irregular access patterns efficiency remains reasonable, as all calculations are based on splines and thus require only simple operations in relatively small neighborhoods.

By using a continuous approach, many difficult problems may find natural solutions. Consider, for example, edge following and linking: powerful path following algorithms exist in the field of numerical analysis, but they require continuously differentiable functions. Convergence statements come in the form of assymptotic theorems $(f - \hat{f})^2 = \mathcal{O}(h^n)$, where $\hat{f}$ is the approximation of f and h the sampling step. Thus, to guarantee a given accuracy, one must be able

Fig. 1. Line junctions drawn on a grid usually occupy more than a single pixel and have rather unpredictable shapes. This can only be prevented with a real-valued (vector) representation.

to *adapt the sampling step locally*. We have found indications that this may also be true in image analysis: in continuous image reconstructions single pixels are often intersected by more than one edge and may contain more than one critical point. In fact, some configurations, in particular junctions, are not in general correctly representable by any grid (fig. 1). The same applies to bifurcations of critical point trajectories encountered in scale selection [7] or edge focusing [1].

Up to now, attempts to access images in real-valued coordinate systems have been based on simple interpolation schemes such as linear interpolation, low order polynomial fits, or the facet model [5,8,3]. However, these methods lead to discontinuities of the function values or the first derivatives at pixel borders, and algorithms requiring differentiability are not applicable. In contrast, we are defining a reconstruction that is everywhere differentiable (up to some order) in both the spatial and the scale directions.

2 Continuity in the Spatial Coordinates

For an observed discrete 1D signal $\hat{f}_i$, the continuous Gaussian scale-space is defined as a family of continuous functions $f_\sigma(x)$ obtained by convolution:

$$f_\sigma(x) = g_\sigma \circledast \hat{f} = \sum_{i=-\infty}^{\infty} g_\sigma(x-i)\hat{f}_i \quad \text{with} \quad g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{1}$$

Unfortunately, this expression cannot directly be used on computers because Gaussian kernels have infinite support and must be clipped to a finite window. No matter how large a window is chosen, a *discontinuity* is introduced at the window borders, and this causes severe errors in the derivatives [12]. [12] recommends to remove the discontinuity of the windowed sampled Gaussian by interpolation with a spline. This is a special case of a more general strategy: first compute an intermediate discrete scale-space representation by means of some discrete prefilter, and then reconstruct a continuous scale-space from it by means of a spline. Splines are a natural choice for this task because they are easy to compute, achieve the highest order of differentiability for a given polynomial order, and have small support. The prefilter will be defined so that the net-result of the prefilter/spline combination approximates the true Gaussian as closely as possible. Ideally, we might require preservation of image structure (e.g. number and location of extrema), but this is very difficult to formalize. Instead we minimize the squared error between the approximation and the desired function:

$$\mathrm{E}[\tilde{f}_\sigma] = \int_{-\infty}^{\infty} (f_\sigma - \tilde{f}_\sigma)^2 dx = \int_{-\infty}^{\infty} (g_\sigma \circledast \hat{f} - s_n \circledast (\hat{p}_\sigma * \hat{f}))^2 dx \tag{2}$$

where $\tilde{f}_\sigma$ is the approximation for scale σ, $\hat{p}_\sigma$ the prefilter, s_n an n^{th}-order B-spline, and $*$ vs. $\circledast$ distinguish discrete from continuous convolution. This minimization problem is still intractable in the spatial domain, but due to Parseval's theorem it can also be formulated and solved (with minor simplifications) in the Fourier domain:

$$\mathrm{E}[\tilde{f}_\sigma] = \int_{-\infty}^{\infty} (G_\sigma - S_n \hat{P}_\sigma)^2 \hat{F}^2 du \tag{3}$$

where $G_\sigma = e^{-u^2\sigma^2/2}$, $S_n = \left(\frac{\sin(u/2)}{u/2}\right)^{n+1}$ and $\hat{P}_\sigma$ are the Fourier transforms of the Gaussian, the spline, and the prefilter. The spectrum $\hat{F}$ of the original image is of course unknown. We use the common choice $\hat{F} = 1$, i.e. a white noise spectrum, where no frequency is preferred. While other possibilities exist (e.g. natural image statistics), this doesn't significantly alter the optimal filter choice.

We have compared many different prefilters and report some of them below. To realize the suggestion of [12] the prefilter $\hat{P}_\sigma$ must be the combination of a sampled windowed Gaussian and the direct spline transform [11] which ensures that the subsequent continuous convolution with the B-spline S_n (indirect spline transform) indeed interpolates the Gaussian's sample values:

$$\hat{P}_\sigma^{(1)} = \frac{\hat{G}_\sigma}{\hat{S}_n} \quad \text{with} \quad \hat{S}_3 = \frac{4 + 2\cos(u)}{6}, \quad \hat{S}_5 = \frac{66 + 52\cos(u) + 2\cos(2u)}{120} \tag{4}$$

$\hat{G}_\sigma$ (the transfer function of a sampled and windowed Gaussian) can be derived by using well-known properties of the Fourier transform: Windowing with a box function of radius w in the spatial domain corresponds to convolution with a scaled sinc-function in the Fourier domain. Spatial sampling with step size $h = 1$ then leads to spectrum repetition at all multiples of 2π. Unfortunately, the resulting infinite sum is intractable. However, in the product $S_n \hat{P}_\sigma$ the B-spline transfer function effectively supresses the spectrum of the prefilter for $u > 2\pi$, so that only the first spectrum repetition at $\pm 2\pi$ needs to be considered, and the effect of windowing can be neglected if $w \geq 3\sigma$. Thus,

$$\hat{G}_\sigma \simeq e^{-(u+2\pi)^2\sigma^2/2} + e^{-u^2\sigma^2/2} + e^{-(u-2\pi)^2\sigma^2/2} \tag{5}$$

A simpler prefilter $\hat{P}_\sigma^{(2)}$ is obtained by noticing that $1/\hat{S}_n$ acts as a sharpening filter that exactly counters the smoothing effect of the indirect spline transform S_n at the sampling points. When we apply the sampled Gaussian $\hat{G}_\sigma$ at a smaller scale $\sigma' < \sigma$, we can drop this sharpening, i.e. $\hat{P}_\sigma^{(2)} = \hat{G}_{\sigma'}$. Further we replaced $\hat{G}_\sigma$ with approximate Gaussians: binomial filters, Deriche's recursive filters [4], Lindeberg's discrete analogue of the Gaussian [7], and the smoothing spline filter from [11]. Space doesn't allow to give all transfer functions here. An even simpler idea is to drop the prefilter altogether, and stretch the B-spline instead so that its variance matches that of the desired Gaussian: $S_{n,\sigma'}(u) = S_n(\sigma' u)$, $\hat{P}_\sigma^{(3)} = 1$. All possibilities mentioned so far perform poorly at small scales ($\sigma < 1$), so we also tested oversampled Gaussians as prefilters, i.e. sampled Gaussians with sampling step $h = 1/2$ whose transfer functions are (the up-arrow denotes oversampling):

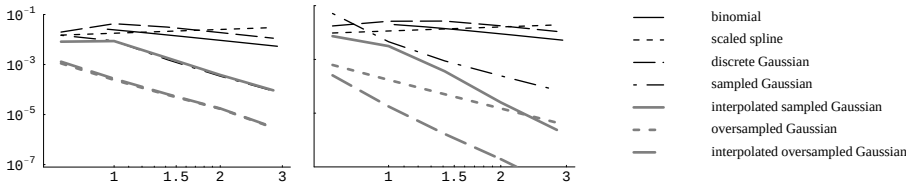

Fig. 2. Scale normalized RMS residuals for Gaussian scale-space approximation with 3^{rd}-order (left) and 5^{th}-order (right) splines for various prefilters and scales.

$$\hat{P}^{(1)}_{\sigma'\uparrow}(u) = \hat{P}^{(1)}_{\sigma}(u/2) \qquad \hat{P}^{(2)}_{\sigma'\uparrow}(u) = \hat{P}^{(2)}_{\sigma}(u/2) \tag{6}$$

and the B-spline transfer function must be accordingly stretched to $S_n(u/2)$.

Figure 2 presents the scale normalized root mean square residuals $\sigma\sqrt{\mathrm{E}}$ of the minimization problem for our prefilter variants at variuous scales. The RMS directly corresponds to the expected error in the spatial domain, and scale normalization is applied in order to make residuals comparable over scales. It can be seen that oversampled Gaussians give the best results, and interpolation (use of $\hat{P}^{(1)}_{\sigma}$ instead of $\hat{P}^{(2)}_{\sigma'}$) only improves 5^{th}-order spline results. At scales $\sigma > \sqrt{2}$, non-oversampling Gaussians also achieve errors below $\simeq 10^{-3}$, which can be considered as good enough for practical applications (it roughly equals the quantization noise for 256 gray levels). We also repeated this analysis with the first and second derivatives of the Gaussian, with essentially the same results.

3 Continuity in Space with Subsampling

So far the resolution of the intermediate images was fixed. Considering that neighboring sampling points become more and more redundant as scale increases, this is rather inefficient, especially for higher dimensional data. We now replace the intermediate representation with a pyramid and analyse the residuals as a function of the scale where subsampling is performed. Usually, subsampling in a pyramid scheme is done by simply dropping every other sampling point. However, in the context of splines we can do better: Since the function space of possible splines with a given sample distance is a strict superset of the function space at half that distance, one can define an orthogonal projection from one space to the other. This projection can be realized by applying a projection filter before dropping samples [11]. The projection filter can be derived analytically, and its transfer function for 3^{rd}-order splines is

$$\Pi_3(u) = \frac{12132 + 18482\cos(u) + 7904\cos(2u) + 1677\cos(3u) + 124\cos(4u) + \cos(5u)}{16(1208 + 1191\cos(2u) + 120\cos(4u) + \cos(6u))} \tag{7}$$

i.e. a combination of a 5^{th}-order FIR and a 3^{rd}-order IIR filter. It is important to note that this filter preserves the average gray value ($\Pi_3(0) = 1$), and fulfills the equal contribution condition, i.e. the even and odd samples have equal

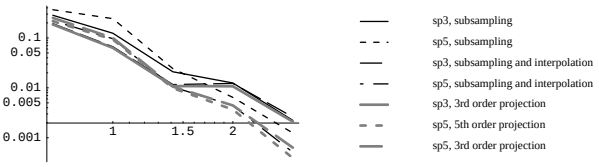

Fig. 3. Scale normalized RMS residuals for various prefilters with subsampling, as a function of the subsampling scale.

total weights ($\Pi_3(\pi) = 0$). If used alone, the projection approximates the ideal lowpass filter (the Fourier transform of the sinc interpolator) but this causes severe ringing artifacts in the reduced images. This is avoided when the projection filter is combined with one of the smoothing prefilters $\hat{P}_\sigma$. To derive their combined transfer functions, recall that $2-$fold subsampling in space corresponds to a spectrum repetition at π in the Fourier domain. The projection filter is optionally applied before subsampling. The subsampled prefilter transfer function is multiplied with the transfer function of a scaled B-spline $S_n(2u)$ (below, $\downarrow k$ means that the approximation resulted from 2^k-fold subsampling):

$$\hat{P}^{(i)}_{\sigma'\downarrow 1}(u) = \hat{P}^{(i)}_{\sigma'\downarrow 0}(u) + \hat{P}^{(i)}_{\sigma'\downarrow 0}(u-\pi) \qquad \text{(without projection)} \tag{8}$$

$$\hat{P}^{(i)+}_{\sigma'\downarrow 1}(u) = \Pi_n(u)\hat{P}^{(i)}_{\sigma'\downarrow 0}(u) + \Pi_n(u-\pi)\hat{P}^{(i)}_{\sigma'\downarrow 0}(u-\pi) \quad \text{(with proj.)} \tag{9}$$

$$\tilde{G}_{\sigma\downarrow 1}(u) = S_n(2u)\hat{P}^{(i)}_{\sigma'\downarrow 1}(u) \qquad \text{or} \qquad \tilde{G}_{\sigma\downarrow 1}(u) = S_n(2u)\hat{P}^{(i)+}_{\sigma'\downarrow 1}(u) \tag{10}$$

For higher levels k of the pyramid, this process is repeated recursively, with spectrum repetitions at $\pi/2^{k-1}$, and splines scaled to $S_n(2^k u)$. Figure 3 depicts the scale normalized RMS errors for a single downsampling step as a function of the scale where the downsampling occurs, for various prefilters (with optimized σ' and with or without the projection filter). It can be seen that an error of 0.01 is achieved for the 3^{rd}-order spline without projection at $\sigma \simeq 2$, and an error of 0.001 for the 5^{th}-order spline with projection at $\sigma \simeq 2.4$. Instead of the rather expensive 5^{th}-order projection filter, 3^{rd}-order projection has been used for 5^{th}-order splines as well, with only a marginal increase in error. Further analysis showed that roughly the same accuracy levels are maintained if subsampling is repeated in the same manner at octave intervals.

4 Continuity in the Scale Direction

If one wants to improve feature detection by means of scale selection or coarse-to-fine tracking, function values or derivatives at arbitrary scales rather than at precomputed ones are often needed. If one uses simple interpolation schemes such as rounding to the nearest scale, linear interpolation or parabola fitting, the true Gaussian scale-space is not approximated very well, and the resulting representation is not differentiable with respect to scale. A much better interpolation scheme can be derived by looking at the diffusion equation whose solution for a given initial image is precisely the Gaussian scale-space

$$\frac{\partial f}{\partial \tau} = \frac{1}{2}\frac{\partial^2 f}{\partial x^2}, \qquad (\tau = \sigma^2) \tag{11}$$

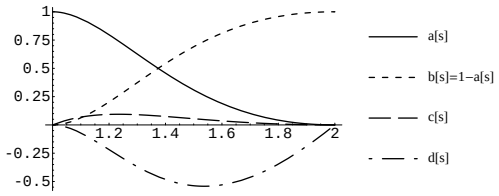

Fig. 4. The blending functions for scale interpolation.

According to this equation the smoothed image at some scale $\tau + \epsilon$ can be calculated from the image at scale τ and the corrsponding second derivative by $f_{\tau+\epsilon}(x) = f_\tau(x) + \epsilon\, f''_\tau(x)$ if ϵ is small. This suggests that a better interpolation scheme can be defined by a linear combination of smoothed images *and second derivatives* (Laplacians in higher dimensions) at two neighboring scales. In particular this means that a Gaussian at scale σ can be interpolated by:

$$\tilde{g}_\sigma(x) \simeq a(\sigma) g_{\sigma_1}(x) + b(\sigma) g_{\sigma_2}(x) + c(\sigma) \frac{\partial^2}{\partial x^2} g_{\sigma_1}(x) + d(\sigma) \frac{\partial^2}{\partial x^2} g_{\sigma_2}(x) \qquad (12)$$

with $\sigma_1 \le \sigma \le \sigma_2$. In order for the interpolation to preserve the average gray value, we must require $b(\sigma) = 1 - a(\sigma)$. Since the same relationship holds in the Fourier domain, we can again formulate a least squares minimization problem

$$\mathrm{E}[a, c, d] = \int_0^{2\pi} \int_{-\infty}^{\infty} (G_\sigma(u) - \tilde{G}_\sigma(u))^2 \, u \, du \, d\varphi \qquad (13)$$

Note that we defined the residual in 2D polar coordinates because this lead to a simpler functional form than the 1D formulation and to higher accuracy in 2D. Setting the derivatives with respect to a, c and d to zero leads to a linear system for the interpolation coefficients. If $\sigma_2 = 2\sigma_1$, the solution to this system is

$$\chi_1 = \sigma^2/\sigma_1^2, \qquad \chi_2 = \frac{1}{(1+\chi_1)(4+\chi_1)},$$

$$a = (62 + \chi_2(-10560 + \chi_2(32000 + 72800\chi_1)))/54 \qquad (14)$$

$$c = \sigma_1^2(15 + \chi_2(-2700 + \chi_2(6000 + 19500\chi_1)))/54 \qquad (15)$$

$$d = \sigma_1^2(240 + \chi_2(-28800 + \chi_2(96000 + 168000\chi_1)))/54 \qquad (16)$$

This is indeed a continuous, differentiable interpolation scheme, as the original Gaussians are recovered at the interpolation borders, and the diffusion equation is fulfilled there, i.e $\tilde{g}_\sigma(x)|_{\sigma=\sigma_{1,2}} = g_{\sigma_{1,2}}(x)$ and $\partial_\tau\, \tilde{g}_\sigma(x)|_{\sigma=\sigma_{1,2}} = \partial_{xx}\, g_{\sigma_{1,2}}(x)/2$. It is somewhat surprising that simple least squares error minimization results in blending formulas which fulfill these requirements, because this was not enforced during the derivation. Probably there is a (yet to be discovered) deeper reason behind this. The accuracy of the scale interpolation scheme is very high. The maximum scale normalized RMS error is 4.5×10^{-3} and is reached at $\sigma = 1.398\sigma_1$. If desired, the error can be reduced by an order of magnitude if $\sigma_2 = \sqrt{2}\sigma_1$ is chosen. Figure 4 depicts the blending functions a, b, c and d. Derivatives are interpolated likewise by replacing $g_{\sigma_{1,2}}$ with the derivative and using its Laplacian. Derivative interpolation thus requires splines of at least order 5.

5 Results and Conclusions

Our analysis suggests that an accurate continuous scale-space approximation can be obtained in two phases: First, an intermediate pyramid representation is computed by means of some optimized discrete filter. Second, function values and derivatives at arbitrary real-valued coordinates and scales are calculated on demand, using spline reconstruction and scale interpolation. These procedures can be encapsulated in an abstract data type, so that algorithms never see the complications behind the calculations. The scale-space starts at base scale σ_{base} which should be at least 0.5. The Gaussian should be windowed at $w \geq 3\sigma$.

Phase 1: Intermediate Pyramid Representation

1. Pyramid level "-1" (scale σ_{base}): Convolve original image with oversampled Gaussian $\hat{g}_{\sigma_{-1}\uparrow}$. Optionally apply the direct spline transform (interpolation prefilter).
2. Level "0" (scale $2\sigma_{\text{base}}$): Convolve original image with sampled Gaussian $\hat{g}_{\sigma_0}$. Optionally apply the direct spline transform.
3. Level "1" (scale $4\sigma_{\text{base}}$): Convolve original image with sampled Gaussian $\hat{g}_{\sigma_1}$. Optionally apply the projection filter. Drop odd samples.
4. Level "k" ($k > 1$): Convolve the intermediate image at level $k-1$ with sampled Gaussian $\hat{g}_{\sigma_2}$. Optionally apply the projection filter. Drop odd samples.

The optimal values for $\sigma_{-1}, ..., \sigma_2$ depend on the order of the spline used, on the value of σ_{base} and on whether or not the interpolation/projection prefilters are applied. Table 1 gives the values for some useful choices. They were calculated by minimizing the scale normalized RMS error between the approximation and the true Gaussian. It can be seen (last column) that these errors decrease for higher order splines, larger σ_{base} and use of interpolation/projection.

Phase 2: On-demand Calculation of Function Values or Derivatives at (x, y, σ)

1. If $\sigma = 2^{k+1}\sigma_{\text{base}}$ $(k \geq -1)$: Work on level k of the intermediate representation. Calculate spline coefficients for $(\delta x, \delta y) = (x/2^k, y/2^k) - (\lfloor x/2^k \rfloor, \lfloor y/2^k \rfloor)$ and convolve with the appropriate image window around $(\lfloor x/2^k \rfloor, \lfloor y/2^k \rfloor)$.
2. If $2^{k+1}\sigma_{\text{base}} < \sigma < 2^{k+2}\sigma_{\text{base}}$ $(k \geq -1)$: Use the algorithm from Phase 2.1 to calculate function values and corresponding Laplacians at levels k and $k+1$. Use the scale interpolation formula to interpolate to scale σ.

The computation time for a single point during phase 2 is independent of the image size. It involves only additions and multiplications (in roughly equal proportions). If σ coincides with one of the precalculated levels, we need 44 multiplications per point for a 3^{rd}-order spline and 102 for a 5^{th}-order one. When an intermediate scale must be interpolated, the numbers are 154 and 342 respectively. Derivative calculations are cheaper as the polynomial order of the splines reduces. When the data are accessed in a fixed order rather than randomly, the effort significantly decreases because intermediate results can be reused. On a modern machine (2.5 GHz Pentium), our implementation provides about a million random point accesses per second for the 5^{th}-order spline. While this is not suitable for real time processing, it is fast enough for practical applications.

algorithm variant	σ_{base}	σ_{-1}	σ_0	σ_1	σ_2	max. resid.
3^{rd}-order spline without	1/2	0.4076	0.8152	1.6304	1.4121	0.018
interpolation/projection	0.6	0.5249	1.0498	2.0995	1.8183	0.0070
	$\sqrt{2}/2$	0.6448	1.2896	2.5793	2.2337	0.0031
5^{th}-order spline without	1/2	0.3531	0.7062	1.4124	1.0586	0.017
interpolation/projection	0.6	0.4829	0.9658	1.9316	1.6728	0.0035
	$\sqrt{2}/2$	0.6113	1.2226	2.4451	2.1175	0.0018
5^{th}-order spline with	1/2	0.4994	0.9987	1.7265	1.5771	0.0062
interpolation/projection	0.6	0.5998	1.1996	2.1790	1.9525	0.0025
	$\sqrt{2}/2$	0.7070	1.4141	2.6442	2.3441	0.0009

Table 1. Optimal scales for sampled Gaussian prefilters for various algorithm variants. "Optional interpolation" refers to levels -1 and 0, "optional projection" (always with 3^{rd}-order projection filter) to levels 1 and higher.

In the future, we will apply the new method to design high-quality subpixel feature detectors. Preliminary results (which we cannot report here due to space) are very encouraging. We also believe that a continuous scale-space representation will open up new roads to scale selection and scale adaptation. For example, variable resolution as in the human eye can be achieved by simply using a position dependent scale instead of an irregular (e.g. log-polar) sampling grid.

Acknowledgement: This work was done during a visit at KTH Stockholm. I'd like to thank Tony Lindeberg for many helpful discussions and comments.

References

1. F. Bergholm: *Edge Focusing*, IEEE Trans. Pattern Analysis and Machine Intelligence, 9(6), pp. 726-741, 1987
2. P. Burt: *The Pyramid as a Structure for Efficient Computation*, in: A. Rosenfeld (Ed.): Multiresolution Image Processing and Analysis, pp. 6-35, Springer, 1984
3. J. Crowley, O. Riff: *Fast Computation of Scale Normalized Gaussian Receptive Fields*, in: L. Griffin, M. Lillholm (Eds.): Scale-Space Methods in Computer Vision, Proc. ScaleSpace '03, Springer LNCS 2695, pp. 584-598, 2003
4. R. Deriche: *Fast algorithms for low-level vision*, IEEE Trans. Pattern Analysis and Machine Intelligence, 1(12), pp. 78-88, 1990
5. R. Haralick, L. Shapiro: *Computer and Robot Vision*, vol. 1, Addison Wesley, 1992
6. U. Köthe: *Edge and Junction Detection with an Improved Structure Tensor*, in: B. Michaelis, G. Krell (Eds.): Pattern Recognition, Proc. 25th DAGM Symposium, Springer LNCS 2781, pp. 25-32, 2003
7. T. Lindeberg: *Scale-Space Theory in Computer Vision*, Kluwer, 1994
8. T. Lindeberg, L. Bretzner: *Real-time scale selection in hybrid multi-scale representations*, in: L. Griffin, M. Lillholm (Eds.): Scale-Space Methods in Computer Vision, Proc. ScaleSpace '03, Springer LNCS 2695, pp. 148-163, 2003
9. D. Lowe: *Object recognition from local scale-invariant features*, In: Proc. 7th Intl. Conf. on Computer Vision, pp. 1150-1157, 1999
10. I. Overington: *Computer Vision*, Elsevier, 1992
11. M. Unser, A. Aldroubi, M. Eden: *B-Spline Signal Processing*, IEEE Trans. Signal Processing, 41(2), pp. 821-833 (part I), 834-848 (part II), 1993
12. I. Weiss: *High-Order Differentiation Filters That Work*, IEEE Trans. Pattern Analysis and Machine Intelligence, 16(7), pp. 734-739, 1994