A Generalized Discrete Scale-Space Formulation for 2-D and 3-D Signals

Ji-Young Lim and H. Siegfried Stiehl

Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Kognitive Systeme
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
{lim, stiehl}@informatik.uni-hamburg.de
http://kogs-www.informatik.uni-hamburg.de/~lim

Abstract. This paper addresses the issue of a higher dimensional discrete scale-space (DSS) formulation. The continuous linear scale-space theory provides a unique framework for visual front-end processes. In practice, a higher dimensional DSS formulation is necessary since higher dimensional discrete signals must be dealt with. In this paper, first we examine the approximation fidelity of the commonly used sampled Gaussian. Second, we propose a generalized DSS formulation for 2-D and 3-D signals. The DSS theory has been presented at first by Lindeberg. While his 1-D DSS formulation is complete, the formulation as related to the extension to higher dimensions has not been fully derived. Furthermore, we investigate the properties of our derived DSS kernels and present the results of a validation study with respect to both smoothing and differentiation performance.

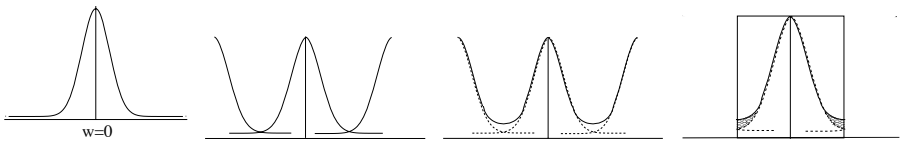
1 Introduction

It is theoretically proven ([1], [15]) that the isotropic Gaussian kernel is the unique kernel to generate the linear scale-space for continuous signals. Furthermore, the Gaussian is the only real-valued convolution kernel which gives the minimum uncertainty of the bandwidth-duration product ([2]) and it satisfies the necessary conditions required for being a lowpass filter. However, given the fact that the Gaussian kernel is defined in the continuous and infinite spatial domain, in practice we have to cope with bounded discrete signals and consequently a discrete Gaussian with compact support is required. A sampled Gaussian (SG) kernel is commonly used in practice, where the problem lies in the accuracy and validity of a SG kernel approximating the continuous Gaussian kernel. Two limitations of the Gaussian kernel were remarked in [13]; i) information loss caused by the unavoidable Gaussian truncation and ii) the prohibitive processing time due to the mask size. Also, it was shown in [5] that there exists a trade-off scale of the Gaussian kernel below which frequency filtering in the Fourier domain yields more accurate results than spatial filtering at the accompanying cost of computational load. In Sect. 2, we analyze the problems behind the SG kernel used as a convolution kernel, where we consider how to measure the fidelity of the approximation of the SG kernel with respect to the continuous Gaussian.

The DSS theory presented at first by Lindeberg [11] is closely linked to the continuous scale-space theory through the discretization of the linear diffusion equation. In his work, the 1-D DSS formulation is well derived and complete, whereas the proposed higher dimensional DSS formulation has left open important questions. Motivated by this, in Sect. 3 we propose a generalized higher dimensional DSS formulation through a clear theoretical derivation that improves upon Lindeberg's higher dimensional DSS formulation. We investigate the properties of our derived DSS kernels in Sect. 4. Moreover, Sect. 5 presents the results of a validation study of the DSS kernel through which we analyze its performance with respect to both smoothing and differentiation.

2 Analysis of the Sampled Gaussian Kernel

According to the sampling theorem (see e.g. [2], [3]), it is theoretically possible to recover the full range of original function values with full accuracy given the condition that the function is "band-limited". Fig. 1 illustrates the sampling process of the Gaussian in the frequency domain: The spectrum of the Fourier transformed continuous Gaussian kernel ($F_G(\omega;t)$; t is the scale parameter) becomes replicated by sampling in the spatial domain through the Shah function, which corresponds to $F_G^{\#}(\omega;t)$. Since the Gaussian is not perfectly band-limited, when its spectrum is repeated, high-frequency components are overlapping. This effect is the so-called *aliasing*. Owing to the aliasing effect, the contribution of high-frequency components is superimposed on low-frequency components. In Fig. 1(c), the solid line results from the aliasing effect. Using the rectangle function for windowing, one aperiodic spectrum ($\hat{F}_G(\omega;t)$) can be cut off, and we call the filled area of both lobes of $\hat{F}_G(\omega;t)$ in Fig. 1(d) the *high-frequency tail*.



(a) $F_G(\omega;t)$ (b) $F_G^{\#}(\omega;t)$ (c) Aliasing (d) $\hat{F}_G(\omega;t)$

Fig. 1. Sampling in the frequency domain: (a) The Fourier transformed continuous Gaussian, (b) replication occurs in the frequency domain by the sampling in the spatial domain, (c) high-frequency components influence on low frequencies around $|\omega| = \frac{\omega_0}{2}$, and (d) the cutoff spectrum contains the high-frequency tail caused by the aliasing.

For a given sampling period, we can derive numerically $\hat{F}_G(\omega;t)$, from which the amount of the high-frequency tail can be calculated. We fix here the sampling period to one (i.e. $T = 1$, or $\omega_0 = 2\pi$) for the reason that the input signal with

which the Gaussian kernel is convolved is in general the intensity function of a digital image with an inter-pixel distance of 1. Since

$$\int_{-\infty}^{\infty} \hat{F}_G(\omega;t)\, d\omega = \int_{-\infty}^{\infty} F_G(\omega;t)\, d\omega, \tag{1}$$

holds (see [8, Eq. 3, Sect. 4] for the detailed derivation), which means that the amount of the high-frequency tail corresponds to the total amount of the contribution of high-frequency components influenced on low-frequency components owing to the aliasing effect, one can conclude that the smaller the amount of the high frequency tail is the better the sampling result is. Provided that the amount of the high-frequency tail is zero (i.e. no aliasing occurs), for example, one can fully reconstruct the continuous Gaussian from the sampled Gaussian. Accordingly, based on the fact derived in (1), the amount of the high-frequency tail denoted by $F_{G_{\mathrm{HFT}}}$ is calculated as

$$F_{G_{\mathrm{HFT}}} = 2\int_{\pi}^{\infty} F_G(\omega;t)\, d\omega = \sqrt{\frac{2\pi}{t}}\left(1 - \mathrm{erf}\left(\pi\sqrt{\frac{t}{2}}\right)\right),$$

where $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{\xi^2} d\xi$. $F_{G_{\mathrm{HFT}}}$ is a monotonously decreasing function of the scale parameter t. Since $F_{G_{\mathrm{HFT}}}$ is expressed by $\mathrm{erf}(x)$, an approximated value can be given only.

Sampling the higher dimensional isotropic Gaussian kernel is analogous to that of the 1-D Gaussian based on the separability property. Therefore, we refrain from describing here in detail the sampling process of higher dimensional Gaussian kernels (see [8, Sect. 4.3] for details). For a given sampling period $T = 1$, the amount of the high-frequency tail in both 2-D and 3-D can be calculated as

$${}^2F_{G_{\mathrm{HFT}}} = (F_{G_{\mathrm{HFT}}})^2 \qquad \text{and} \qquad {}^3F_{G_{\mathrm{HFT}}} = (F_{G_{\mathrm{HFT}}})^3,$$

which can be further generalized for an N-D Gaussian kernel to $(F_{G_{\mathrm{HFT}}})^N$. Fig. 2 depicts the high-frequency tail of the sampled Gaussian kernel in 1-D, 2-D, and 3-D in dependence on the scale parameter, from which one can easily recognize that as the scale parameter decreases the amount of high-frequency tail increases in each dimension.

Consequently, it can be generally said that a sampled Gaussian with a small scale is not appropriate for approximating the continuous Gaussian.

3 DSS Formulation

Several open problems of Lindeberg's higher dimensional DSS formulation can be identified: i) The rationale behind the setting of coefficients for the point operator (i.e. $\frac{1}{2}$ for $\nabla^2_{\times 2}$ in 2-D as well as $\frac{1}{4}$ for both ∇^2_{+3} and $\nabla^2_{\times 3}$ in 3-D) has not been explained, ii) in the 2-D DSS formulation the answer to the question of how to determine the definite parameter γ for solving the semi-discretized diffusion equation was left open, and iii) the 3-D DSS formulation was not considered at all, which, however, is necessary e.g. for 3-D medical image analysis. Therefore, it is indispensable to contribute a generalized higher dimensional DSS formulation.

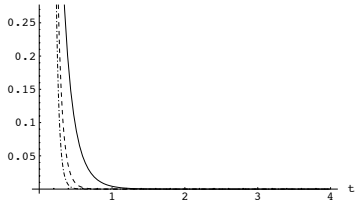


Fig. 2. The high-frequency tail of the sampled Gaussian kernel in 1-D, 2-D, and 3-D ($T = 1$). The solid curve, the dashed curve, and the dotted curve represent $F_{G_{\mathrm{HFT}}}$, ${}^2F_{G_{\mathrm{HFT}}}$, and ${}^3F_{G_{\mathrm{HFT}}}$, respectively.

3.1 Preliminaries

The Neighborhood Connectivity. For a given point $p \in \mathbb{Z}^k$, we define its neighborhood as

$$ {}^kN(p) = \{\xi \in \mathbb{Z}^k : (\|p - \xi\| \leq \sqrt{k}) \wedge (\xi \neq p)\}, \tag{2}$$

for $k \geq 1$. The neighborhood is classified according to the distance between a given point and its neighbors. That is, ${}^2N(p)$ consists of ${}^2N_1(p)$ and ${}^2N_{\sqrt{2}}(p)$, ${}^3N(p)$ comprises ${}^3N_1(p)$, ${}^3N_{\sqrt{2}}(p)$, and ${}^3N_{\sqrt{3}}(p)$, and ${}^NN(p)$ has ${}^NN_1(p)$, ${}^NN_{\sqrt{2}}(p)$, $\cdots$, ${}^NN_{\sqrt{N}}(p)$ (the number of elements of ${}^NN_{\sqrt{k}}(p)$ is equivalent to $\binom{N}{k}2^k$).

The Laplacian of the Higher Dimensional DSS Kernel. On the basis of numerical differentiation, the second-order derivative of $f(x)$ is approximated by

$$f(x+1) - 2f(x) + f(x-1) \approx \frac{\partial^2 f(x)}{\partial x^2}.$$

According to (2), for a given discrete signal $f : \mathbb{Z}^N \to \mathbb{R}$ its scale-space representation generated by the convolution with $T : \mathbb{Z}^N \times \mathbb{R}_+ \to \mathbb{R}$ (which we call the *higher dimensional DSS kernel*) satisfies in 2-D

$$\partial_t L(x, y; t) = a_1 \nabla^2_{{}^2N_1} L(x, y; t) + a_2 \nabla^2_{{}^2N_{\sqrt{2}}} L(x, y; t), \tag{3}$$

and in 3-D

$$\partial_t L(x, y, z; t) = a_1 \nabla^2_{{}^3N_1} L(x, y, z; t) + a_2 \nabla^2_{{}^3N_{\sqrt{2}}} L(x, y, z; t) + a_3 \nabla^2_{{}^3N_{\sqrt{3}}} L(x, y, z; t), \tag{4}$$

for some constants $a_1 \geq 0$, $a_2 \geq 0$, and $a_3 \geq 0$.

3.2 2-D DSS Formulation

Equation (3) can be expressed as a normalized form given by

$$\partial_t L(x, y; t) = \frac{1}{2}\nabla^2 L(x, y; t) = \frac{1}{2}\left((1-\gamma)\nabla^2_{{}^2N_1} L(x, y; t) + \lambda\gamma \nabla^2_{{}^2N_{\sqrt{2}}} L(x, y; t)\right) \tag{5}$$

for $\gamma \in [0,1]$ and $\lambda \in (0,1)$. Note that in [9, Sect. 3.2], for the definition of the Laplacian of ${}^2N_{\sqrt{2}}$, we intended to steer the ratio of rotational symmetry between the Laplacians of ${}^2N_1$ and ${}^2N_{\sqrt{2}}$ only through parameter γ, and thus we avoided setting an additional ambiguous (or unexplained) coefficient. However, this does not give rise to a correct result, i.e. the 2-D DSS kernel does not satisfy the semi-group property. On the other hand, one cannot find any proof that the unexplained coefficient in the definition of $\nabla^2_{\times 2}$ given by Lindeberg is proper. Therefore, in a more generalized way, we set this coefficient as a variable λ and determine its proper value through the following theoretical derivation.

Equation (5) can be further discretized with the scale step $\triangle t$ as

$$L^{k+1}_{x,y} = L^k_{x,y} + \triangle t \left(\partial_t L^k_{x,y}\right) = L^k_{x,y} + \triangle t \frac{1}{2}\left((1-\gamma)\nabla^2_{{}^2N_1} L + \lambda\gamma \nabla^2_{{}^2N_{\sqrt{2}}} L\right),$$

where subscripts x and y denote the spatial coordinates and superscript k represents the iteration index. This discretization corresponds to the iteration with the 2-D discrete iteration kernel given by

$$T_{\triangle t} = \begin{pmatrix} \frac{\lambda\triangle t}{2}\gamma & \frac{\triangle t}{2}(1-\gamma) & \frac{\lambda\triangle t}{2}\gamma \\ \frac{\triangle t}{2}(1-\gamma) & 1-2\triangle t(1-\gamma+\lambda\gamma) & \frac{\triangle t}{2}(1-\gamma) \\ \frac{\lambda\triangle t}{2}\gamma & \frac{\triangle t}{2}(1-\gamma) & \frac{\lambda\triangle t}{2}\gamma \end{pmatrix}. \tag{6}$$

The generating function describing one iteration given in (6) corresponds to

$${}^2\varphi_{step}(z,\chi) = (1-2\triangle t(1-\gamma+\lambda\gamma)) + \frac{\triangle t(1-\gamma)}{2}A + \frac{\lambda\triangle t\gamma}{2}B,$$

where

$$A = z^{-1} + z + \chi^{-1} + \chi \quad \text{and} \quad B = z^{-1}\chi^{-1} + z^{-1}\chi + z\chi^{-1} + z\chi,$$

and we obtain the generating function describing the composed transformation ($\triangle t = \frac{t}{n}$) as

$${}^2\varphi_{composed,n}(z,\chi) = \left(1 + \frac{t}{n}\left(-2(1-\gamma+\lambda\gamma) + \frac{(1-\gamma)}{2}A + \frac{\lambda\gamma}{2}B\right)\right)^n.$$

Based on the fact that $\lim_{n\to\infty}(1+\frac{\alpha_n}{n})^n = e^\alpha$ if $\lim_{n\to\infty}\alpha_n = \alpha$, the generating function of the kernel describing the transformation from the original signal to the representation at a certain scale t is given by

$${}^2\varphi_T(z,\chi) = \sum_{(m,n)\in\mathbb{Z}^2} T(m,n;t) z^m \chi^n = e^{t\left(-2(1-\gamma+\lambda\gamma)+\frac{(1-\gamma)}{2}A+\frac{\lambda\gamma}{2}B\right)}.$$

Its Fourier transform is derived by replacing the complex variables z and χ with e^{-iu} and e^{-iv} as

$$\begin{aligned}\mathcal{F}\left({}^2\varphi_T(z,\chi)\right) &= {}^2\psi_T(e^{-iu}, e^{-iv}) = {}^2\psi_T(\cos u - i\sin u, \cos v - i\sin v) \\ &= e^{t(-2(1-\gamma+\lambda\gamma)+(1-\gamma)(\cos u+\cos v)+\lambda\gamma 2\cos u\cos v)},\end{aligned}$$

which can be transformed into polar coordinates given a fixed value of radius r and an angular variable ϕ such that $u = r\cos\phi$ and $v = r\sin\phi$. It follows

$$^2\psi_T(r,\phi) = e^{(t\cdot k(r,\phi))},$$

where

$$\begin{aligned} k(r,\phi) = -2(1-\gamma+\lambda\gamma) + (1-\gamma)(\cos(r\cos\phi) + \cos(r\sin\phi)) \\ + \lambda\gamma 2\cos(r\cos\phi)\cos(r\sin\phi). \end{aligned} \tag{7}$$

Now, we determine the value of γ of $k(r,\phi)$ in (7) which gives the smallest angular variation of ϕ for a fixed value r. For examining the ϕ-dependency of γ from $k(r,\phi)$, we expand the MacLaurin series of $k(r,\phi)$ with respect to r

$$k(r,\phi) = -\frac{1-\gamma+2\lambda\gamma}{2}r^2 + \frac{3+3\gamma(4\lambda-1)+(1-\gamma-4\lambda\gamma)\cos 4\phi}{96}r^4 + O(r^6),$$

where the smallest angular variation is achieved when $\gamma = \frac{1}{1+4\lambda}$. That is to say, $\gamma = \frac{1}{1+4\lambda}$ yields the least possible rotational asymmetry for the 2-D DSS kernel. In other words, maximal isotropy of T is guaranteed even for the discrete case. Substituting $\gamma = \frac{1}{1+4\lambda}$ for the 2-D iteration kernel of (6) yields

$$T_{\triangle t} = \begin{pmatrix} \frac{\lambda}{2(1+4\lambda)}\triangle t & \frac{2\lambda}{1+4\lambda}\triangle t & \frac{\lambda}{2(1+4\lambda)}\triangle t \\ \frac{2\lambda}{1+4\lambda}\triangle t & 1-\frac{10\lambda}{1+4\lambda}\triangle t & \frac{2\lambda}{1+4\lambda}\triangle t \\ \frac{\lambda}{2(1+4\lambda)}\triangle t & \frac{2\lambda}{1+4\lambda}\triangle t & \frac{\lambda}{2(1+4\lambda)}\triangle t \end{pmatrix}, \tag{8}$$

where $\triangle t > 0$. This iteration kernel is symmetric and normalized.

In order to constrain computational cost when the number of iterations increases, it would be favorable to apply a separable iteration kernel since the higher the dimension is the more efficient separable filters are ([6]). Therefore, we assume the 2-D iteration kernel given in (8) to be separable such that it should be constructed by convolution of the 1-D kernel with itself given by

$$\begin{pmatrix} a & 1-2a & a \end{pmatrix} * \begin{pmatrix} a \\ 1-2a \\ a \end{pmatrix} = \begin{pmatrix} \frac{\lambda}{2(1+4\lambda)}\triangle t & \frac{2\lambda}{1+4\lambda}\triangle t & \frac{\lambda}{2(1+4\lambda)}\triangle t \\ \frac{2\lambda}{1+4\lambda}\triangle t & 1-\frac{10\lambda}{1+4\lambda}\triangle t & \frac{2\lambda}{1+4\lambda}\triangle t \\ \frac{\lambda}{2(1+4\lambda)}\triangle t & \frac{2\lambda}{1+4\lambda}\triangle t & \frac{\lambda}{2(1+4\lambda)}\triangle t \end{pmatrix},$$

for $0 < a \leq \frac{1}{4}$, from which we obtain

$$a = \frac{1}{6} \quad \text{and} \quad \triangle t = \frac{2}{9} + \frac{1}{18\lambda}.$$

Besides, in order to satisfy the semi-group property, $\triangle t$ should correspond to the variance of the 2-D discrete iteration kernel. This means that $\triangle t = 2a$ must hold. Finally, we have

$$a = \frac{1}{6}, \quad \lambda = \frac{1}{2}, \quad \gamma = \frac{1}{3}, \quad \text{and} \quad \triangle t = \frac{1}{3}.$$

It is noticeable that our theoretically derived value of λ is equal to that defined by Lindeberg [11], where, however, no formal explanation was given.

Consequently, the 2-D separable iteration kernel for the rotationally least asymmetric 2-D DSS kernel satisfying the semi-group property is given by

$$T_{\triangle t} = \left(\tfrac{1}{6}\ \tfrac{2}{3}\ \tfrac{1}{6}\right) * \begin{pmatrix} \frac{1}{6} \\ \frac{2}{3} \\ \frac{1}{6} \end{pmatrix} = \begin{pmatrix} \frac{1}{36} & \frac{1}{9} & \frac{1}{36} \\ \frac{1}{9} & \frac{4}{9} & \frac{1}{9} \\ \frac{1}{36} & \frac{1}{9} & \frac{1}{36} \end{pmatrix}, \tag{9}$$

where $\triangle t = \frac{1}{3}$.

3.3 3-D DSS Formulation

Equation (4) can be expressed as a normalized form given by

$$\partial_t L = \frac{1}{2}\left((1-\gamma_1-\gamma_2)\nabla^2_{^3N_1} L + \lambda_1\gamma_1\nabla^2_{^3N_{\sqrt{2}}} L + \lambda_2\gamma_2\nabla^2_{^3N_{\sqrt{3}}} L\right) \tag{10}$$

for $\gamma_1, \gamma_2 \in [0,1]$ and $\lambda_1, \lambda_2 \in (0,1)$. Following the line of thought of the 2-D case, in defining the Laplacians of ${}^3N_{\sqrt{2}}$ and of ${}^3N_{\sqrt{3}}$ in [9, Sect. 3.3], we intended to steer the ratio of rotational symmetry between the Laplacians of ${}^3N_1$, ${}^3N_{\sqrt{2}}$, and ${}^3N_{\sqrt{3}}$ only through parameters γ_1 and γ_2. Therefore, we did not define any a priori coefficients for the Laplacians of ${}^3N_{\sqrt{2}}$ and of ${}^3N_{\sqrt{3}}$, whereas Lindeberg set them both to $\frac{1}{4}$. However, our first account in [9, Sect. 3.3] did not give rise to a proper result (the semi-group property is not satisfied). On the other hand, it is neither evident nor proven why those parameters were set to $\frac{1}{4}$ by Lindeberg. As a consequence, we again approach this problem in a more generalized way by defining the coefficients of the Laplacians of ${}^3N_{\sqrt{2}}$ and of ${}^3N_{\sqrt{3}}$ as λ_1 and λ_2.

Equation (10) can be discretized with the scale step $\triangle t$ given by

$$\begin{aligned} L^{k+1}_{x,y,z} &= L^k_{x,y,z} + \triangle t\left(\partial_t L^k_{x,y,z}\right) \\ &= L^k_{x,y,z} + \triangle t\frac{1}{2}\left((1-\gamma_1-\gamma_2)\nabla^2_{^3N_1} L + \lambda_1\gamma_1\nabla^2_{^3N_{\sqrt{2}}} L + \lambda_2\gamma_2\nabla^2_{^3N_{\sqrt{3}}} L\right), \end{aligned}$$

where the parameters γ_1 and γ_2 play the role of preserving the rotational symmetry of the 3-D DSS kernel. Similarly to the 2-D case, based on the assumptions that i) the 3-D DSS kernel is rotationally least asymmetric, ii) the 3-D iteration kernel is separable, and iii) the 3-D DSS kernel satisfies the semi-group property, we determine the parameter values as follows (see [10, Sect. 3.2] for the detailed derivation):

$$\gamma_1 = \frac{4}{9}, \quad \gamma_2 = \frac{1}{9}, \quad \lambda_1 = \frac{1}{4}, \quad \lambda_2 = \frac{1}{4}, \quad a = \frac{1}{6}, \quad \text{and} \quad \triangle t = \frac{1}{3}.$$

As a consequence, the 3-D separable iteration kernel for the rotationally least asymmetric 3-D DSS kernel satisfying the semi-group property is given by

$$T_{\triangle t} = \left(\tfrac{1}{6}\ \tfrac{2}{3}\ \tfrac{1}{6}\right)_x * \left(\tfrac{1}{6}\ \tfrac{2}{3}\ \tfrac{1}{6}\right)_y * \left(\tfrac{1}{6}\ \tfrac{2}{3}\ \tfrac{1}{6}\right)_z = \begin{pmatrix} \frac{1}{216} & \frac{1}{54} & \frac{1}{216} \\ \frac{1}{54} & \frac{2}{27} & \frac{1}{54} \\ \frac{1}{216} & \frac{1}{54} & \frac{1}{216} \end{pmatrix}_{z\pm 1}, \begin{pmatrix} \frac{1}{54} & \frac{2}{27} & \frac{1}{54} \\ \frac{2}{27} & \frac{8}{27} & \frac{2}{27} \\ \frac{1}{54} & \frac{2}{27} & \frac{1}{54} \end{pmatrix}_z,$$

where $\triangle t = \frac{1}{3}$.

4 Properties of the DSS Kernels

4.1 Smoothing Kernel

The 1-D DSS kernel is given by

$$T\left(x;\frac{k}{3}\right) = *^k \left(\frac{1}{6} \quad \frac{2}{3} \quad \frac{1}{6}\right), \tag{11}$$

where $*^k$ is denoted as k-times self-convolution and $\frac{k}{3}$ corresponds to the variance. The coefficients of the 1-D DSS kernel generated by self-convolution given in (11) can be easily calculated using the z-transform of the given DSS kernel based on the property that convolution of sequences corresponds to multiplication of their z-transforms, from which

$$T\left(x;\frac{k}{3}\right) \circ\!\!-\!\!\bullet \left(\frac{1}{6}z^{-1} + \frac{2}{3} + \frac{1}{6}z\right)^k$$

follows. The 1-D DSS kernels are normalized to 1 for any k, and their implementation is simple and fast.

The higher dimensional DSS kernel is separable. For example, the smallest 2-D DSS kernel is given by

$$T\left(x,y;\frac{1}{3}\right) = T\left(x;\frac{1}{3}\right) * T\left(y;\frac{1}{3}\right) = \left(\frac{1}{6}\ \frac{2}{3}\ \frac{1}{6}\right)_x * \left(\frac{1}{6}\ \frac{2}{3}\ \frac{1}{6}\right)_y,$$

the coefficients of which can be easily calculated using its z-transform

$$T\left(x;\frac{1}{3}\right) * T\left(y;\frac{1}{3}\right) \circ\!\!-\!\!\bullet \left(\frac{1}{6}z^{-1} + \frac{2}{3} + \frac{1}{6}z\right) \cdot \left(\frac{1}{6}\omega^{-1} + \frac{2}{3} + \frac{1}{6}\omega\right) \bullet\!\!-\!\!\circ \begin{pmatrix} \frac{1}{36} & \frac{1}{9} & \frac{1}{36} \\ \frac{1}{9} & \frac{4}{9} & \frac{1}{9} \\ \frac{1}{36} & \frac{1}{9} & \frac{1}{36} \end{pmatrix}.$$

The higher dimensional DSS kernel with larger variance, analogously to 1-D, can be derived through self-convolution given by

$$T\left(x,y;\frac{k}{3}\right) = *^k\, T\left(x,y;\frac{1}{3}\right) \circ\!\!-\!\!\bullet \left(\frac{1}{6}z^{-1} + \frac{2}{3} + \frac{1}{6}z\right)^k \cdot \left(\frac{1}{6}\omega^{-1} + \frac{2}{3} + \frac{1}{6}\omega\right)^k,$$

where k denotes the number of self-convolution.

4.2 Differencing Kernel

In order to apply the DSS kernel to images for the purpose of feature extraction, it is necessary to derive derivative operators. In contrast to the continuous case in which any nth-order derivatives of the Gaussian can be defined at any scale, it is not as simple to define derivative operators in the discrete case. By introducing the terminology "differencing operator" denoted by $\triangle$, we here discriminate the discrete derivative from the continuous derivative.

Based on the principles of numerical differentiation, one can approximate the first-order derivative by the difference quotient, for which there exist two formulae according to the number of points involved in the differencing. One is the two-point difference formula denoted by $\triangle_{\text{even}}$

$$f_{\triangle_{\text{even}}}(x) = \frac{f(x) - f(x-h)}{h} = f(x) - f(x-1) \quad (h = 1),$$

while the other is the three-point difference formula denoted by $\triangle_{\text{odd}}$

$$f_{\triangle_{\text{odd}}}(x) = \frac{f(x+h) - f(x-h)}{2h} = \frac{f(x+1) - f(x-1)}{2} \quad (h = 1).$$

Based on these two formulae, we thoroughly derive two types of the 1-D DSS first-order differencing operator using the z-transform. The z-transform of $\triangle_{\text{even}}$ is given by

$$f_{\triangle_{\text{even}}}(x) = f(x) - f(x-1) \circ\!\!-\!\!\bullet\, F(z) \cdot (1 - z),$$

where $F(z)$ corresponds to the z-transform of $f(x)$. The DSS first-order differencing kernel follows through application of $\triangle_{\text{even}}$ as

$$T_{\triangle_{\text{even}}}\left(x; \frac{k}{3}\right) \circ\!\!-\!\!\bullet \left(\frac{1}{6}z^{-1} + \frac{2}{3} + \frac{1}{6}z\right)^k (1 - z) \bullet\!\!-\!\!\circ\, *^k \left\{\frac{1}{6}\ \frac{2}{3}\ \frac{1}{6}\right\} * \{1\ -1\}.$$

Analogously to $\triangle_{\text{even}}$, the z-transform of $\triangle_{\text{odd}}$ is given by

$$f_{\triangle_{\text{odd}}}(x) = \frac{f(x+1) - f(x-1)}{2} \circ\!\!-\!\!\bullet\, F(z) \cdot \left(\frac{1}{2}z^{-1} - \frac{1}{2}z\right),$$

and the DSS first-order differencing kernel follows from application of $\triangle_{\text{odd}}$

$$T_{\triangle_{\text{odd}}}\left(x; \frac{k}{3}\right) \circ\!\!-\!\!\bullet \left(\frac{1}{6}z^{-1} + \frac{2}{3} + \frac{1}{6}z\right)^k \left(\frac{1}{2}z^{-1} - \frac{1}{2}z\right) \bullet\!\!-\!\!\circ\, *^k \left\{\frac{1}{6}\ \frac{2}{3}\ \frac{1}{6}\right\} * \left\{\frac{1}{2}\ 0\ \frac{-1}{2}\right\}.$$

For a given higher dimensional DSS kernel, the differencing kernel through application of $\triangle_{x_\alpha}$ is derived from

$$T_{\triangle_{x_\alpha}}(\overrightarrow{x}; \cdot) = T(x_1; \cdot) * T(x_2; \cdot) * \cdots * T_{\triangle}(x_\alpha; \cdot) * \cdots * T(x_N; \cdot).$$

For example, applying $\triangle_{\text{even},x}$ to $T(x, y; \frac{1}{3})$ results in

$$T_{\triangle_{\text{even},x}}\left(x, y; \frac{1}{3}\right) = \left(\tfrac{1}{6}\ \tfrac{1}{2}\ \tfrac{-1}{2}\ \tfrac{-1}{6}\right)_x * \left(\tfrac{1}{6}\ \tfrac{2}{3}\ \tfrac{1}{6}\right)_y = \begin{pmatrix} \frac{1}{36} & \frac{1}{12} & \frac{-1}{12} & \frac{-1}{36} \\ \frac{1}{9} & \frac{1}{3} & \frac{-1}{3} & \frac{-1}{9} \\ \frac{1}{36} & \frac{1}{12} & \frac{-1}{12} & \frac{-1}{36} \end{pmatrix},$$

while applying $\triangle_{\text{odd},y}$ to $T(x, y; \frac{1}{3})$ yields

$$T_{\triangle_{\text{odd},y}}\left(x, y; \frac{1}{3}\right) = \left(\tfrac{1}{6}\ \tfrac{2}{3}\ \tfrac{1}{6}\right)_x * \left(\tfrac{1}{12}\ \tfrac{1}{3}\ 0\ \tfrac{-1}{3}\ \tfrac{-1}{12}\right)_y = \begin{pmatrix} \frac{1}{72} & \frac{1}{18} & \frac{1}{72} \\ \frac{1}{18} & \frac{2}{9} & \frac{1}{18} \\ 0 & 0 & 0 \\ \frac{-1}{18} & \frac{-2}{9} & \frac{-1}{18} \\ \frac{-1}{72} & \frac{-1}{18} & \frac{-1}{72} \end{pmatrix}.$$

Normalization. Let us assume that i) $f(x)$ is a scale-space kernel, ii) $f(x)$ should be sufficiently smooth such that n-th order derivatives can be taken, iii) $f(x)$ is normalized such that $\int f(x)\,dx = 1$, and additionally iv) $f(x)$ is essentially compact, meaning that the kernel and all of its derivatives vanish sufficiently fast when $|x|$ goes to infinity. Provided that these assumptions are satisfied, $f(x)$ simply follows by partial integration, i.e.

$$\int \frac{(-1)^n}{n!} x^n f^{(n)}(x)\,dx = 1,$$

where n corresponds to the order of the derivative (see [4]). According to this rule, in the case of the first-order derivative (i.e. $n = 1$),

$$\int -x f'(x)\,dx = 1 \tag{12}$$

must hold. It is evident that the $T_{\triangle_{\text{even}}}$ and $T_{\triangle_{\text{odd}}}$ satisfy the normalization requirement given in (12), and thus these two kernels are normalized DSS differencing kernels.

Variance. For a given $f : \mathbb{R} \to \mathbb{R}$, its variance derived by the second central moment (by assuming its existence) can be written as

$$\text{Var}(f(x)) = \int_{-\infty}^{\infty} x^2 |f(x)|\,dx, \tag{13}$$

in order to measure a dispersion of $f(x)$. According to (13), the variance of (even- and odd-number-sized) $T_{\triangle}(x;t)$ can be calculated, from which one can find that $\text{Var}(T_{\triangle_{\text{even}}}(x;t))$ nicely equals $\text{Var}(T_{\triangle_{\text{odd}}}(x;t))$ for any t even though the two kernels are different in local support and shape.

Integration. For a given z-transformed DSS kernel $T(z)$ (i.e. $T(z) \bullet\!\!-\!\!\circ\, T(x)$), $T_{\triangle}(z)$ is derived from multiplying $T(z)$ with $\triangle_z$, i.e.

$$T_{\triangle}(z) \cdot \triangle_z^{-1} = T(z) \cdot \triangle_z \cdot \triangle_z^{-1} = T(z).$$

Introducing the symbol $\blacktriangle$ that stands for $\triangle^{-1}$, we denote $\blacktriangle$ as the "discrete integration operator" (due to the duality of differentiation and integration). Multiplication of $\blacktriangle_z$ corresponds to convolution of $\blacktriangle$ in the spatial domain. Since there exist two types of $\triangle_z$, there are correspondingly two types of $\blacktriangle_z$, i.e. $\blacktriangle_{\text{even},z}$ and $\blacktriangle_{\text{odd},z}$.

$\blacktriangle_{\text{even},z}$ is given by

$$\blacktriangle_{\text{even},z} = (\triangle_{\text{even},z})^{-1} = (1-z)^{-1},$$

where the inverse z-transform of $\blacktriangle_{\text{even},z}$ is derived as

$$\blacktriangle_{\text{even},z} = \frac{1}{1-z} \bullet\!\!-\!\!\circ\, \mathcal{H}(x) = \{\cdots 0\ \ 0\ \ 0\ \underset{x=0}{1}\ 1\ \ 1 \cdots\}.$$

Integration of the DSS kernel through application of $\blacktriangle_{\text{even}}$ yields

$$T_{\blacktriangle_{\text{even}}}\left(x;\frac{k}{3}\right) \circ\!\!-\!\!\bullet\, T\left(z;\frac{k}{3}\right)\cdot \blacktriangle_{\text{even},z},$$

where, e.g. in the case of $k=1$,

$$T_{\blacktriangle_{\text{even}}}\left(x;\frac{1}{3}\right) \circ\!\!-\!\!\bullet\, \frac{1}{6}\left(-1+\frac{1}{z}+\frac{6}{1-z}\right) \bullet\!\!-\!\!\circ\, -\frac{1}{6}\delta(x)+\frac{1}{6}\delta(x+1)+\mathcal{H}(x),$$

i.e. $T_{\blacktriangle_{\text{even}}}(x;\frac{1}{3}) = \{\cdots\ 0\ \frac{1}{6}\ \frac{5}{6}\ 1\ \cdots\}$. Analogously, $\blacktriangle_{\text{odd},z}$ is given by

$$\blacktriangle_{\text{odd},z} = (\triangle_{\text{odd},z})^{-1} = \left(\frac{1}{2}z^{-1}-\frac{1}{2}z\right)^{-1} = \frac{2z}{1-z^2},$$

where the inverse z-transform of $\blacktriangle_{\text{odd},z}$ is derived as

$$\blacktriangle_{\text{odd},z} = \left(\frac{1}{1-z}-\frac{1}{1+z}\right) \bullet\!\!-\!\!\circ\, \mathcal{H}(x)-(-1)^x\mathcal{H}(x) = \{\cdots 0\ \ 0\ \ 0\ \ \underset{x=0}{0}\ \ 2\ \ 0\ \ 2\cdots\}.$$

Integration of the DSS kernel through application of $\blacktriangle_{\text{odd}}$ is given by

$$T_{\blacktriangle_{\text{odd}}}\left(x;\frac{k}{3}\right) \circ\!\!-\!\!\bullet\, T\left(z;\frac{k}{3}\right)\cdot \blacktriangle_{\text{odd},z},$$

where, e.g. in the case of $k=1$,

$$T_{\blacktriangle_{\text{odd},z}}\left(x;\frac{1}{3}\right) \circ\!\!-\!\!\bullet\, \frac{1}{3}\left(-1+\frac{3}{1-z}+\frac{-1}{1+z}\right) \bullet\!\!-\!\!\circ\, -\frac{1}{3}\delta(x)+\mathcal{H}(x)-\frac{(-1)^x}{3}\mathcal{H}(x),$$

i.e. $T_{\blacktriangle_{\text{odd}}}(x;\frac{1}{3}) = \{\cdots\ 0\ \frac{1}{3}\ \{\frac{4}{3}\ \frac{2}{3}\}\ \cdots\}$, where $\{\frac{4}{3}\ \frac{2}{3}\}$ denotes $\{\cdots\ \frac{4}{3}\ \frac{2}{3}\ \frac{4}{3}\ \frac{2}{3}\ \cdots\}$.

As a consequence, when one executes discrete integration and discrete differentiation simultaneously, it must be considered that $\blacktriangle_{\text{even}}$ ($\blacktriangle_{\text{odd}}$) is necessarily paired with $\triangle_{\text{even}}$ ($\triangle_{\text{odd}}$). Otherwise, one cannot expect a correct result since

$$T_{\blacktriangle_{\text{even}}}(z;\cdot)\cdot\triangle_{\text{odd},z} \neq T_{\blacktriangle_{\text{odd}}}(z;\cdot)\cdot\triangle_{\text{even},z} \neq T(z;\cdot).$$

5 Validation of the DSS Kernels

In this section, we validate the derived DSS kernel in comparison to the SG kernel in order to characterize its performance with respect to both smoothing and differentiation. As important performance criteria, we consider the accuracy of approximation, the fulfillment of the non-enhancement requirement (see [11]), and the accuracy of edge extraction. The criterion of accuracy of edge extraction is further divided into rotational invariance and steadiness from adjacency.

5.1 Accuracy of Approximation

We intend to measure how accurately a discrete convolution approximates a continuous convolution. To this end, we consider a continuous constant function $f(x) = c$ with $c \in \mathbb{R}$. Theoretically, convolution of a constant function with the normalized Gaussian kernel should result in the constant function again:

$$L(x;t) = f(x) * G(x;t) = c * G(x;t) = c. \tag{14}$$

In practice, (14) is implemented by

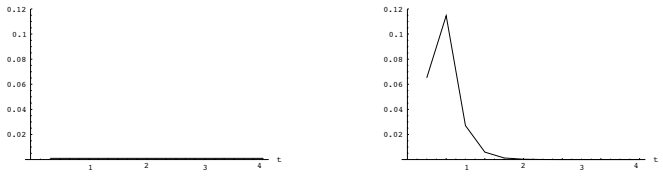
$$L_d(x;t) = f_d(x) * G_d(x;t) = \{\cdots\ c\ \ c\ \ c\ \cdots\} * G_d(x;t),$$

where $f_d(x)$ is the discrete constant signal ($c = 100$ in the experiment), $G_d(x;t)$ is a discrete kernel. The approximation error of discrete convolution is given by

$$\overline{\xi}(t) = \frac{1}{(2n_l+1)} \sum_{l=-n_l}^{n_l} \xi_l(t),$$

where $2n_l + 1$ corresponds to the number of coefficients of $L_d(x;\cdot)$ and $\xi_l(t) = |L_d(x_l;t) - L(x_l;t)|$.

The results of approximation accuracy are given in Fig. 3: $\overline{\xi}_T(t)$ consistently gives zero as t gradually increases, which implies that the discrete convolution with $T(x;t)$ accurately approximates the continuous convolution for any t, whereas $\overline{\xi}_{SG}(t)$ inconsistently varies over t and even attains a maximum at a small t. This unsatisfactory experimental result of the SG kernel is connected to the analytical result of the SG kernel given in Sect. 2. Consequently, it can be said that $T(x;t)$ is superior to $SG(x;t)$ with respect to approximation of discrete convolution when t gets smaller.



(a) $\overline{\xi}_T(t)$ (b) $\overline{\xi}_{SG}(t)$

Fig. 3. Accuracy of approximation of the DSS kernel compared with the SG kernel.

5.2 Fulfillment of the Non-Enhancement Requirement

According to the prerequisites for the DSS formulation proposed by Lindeberg [11], a higher dimensional DSS kernel is assumed to obey the non-enhancement

requirement. In order to examine the fulfillment of the non-enhancement requirement of the DSS kernel as well as of the SG kernel, we provide a synthetic image that has two local maxima and two local minima (one local extremum has a high intensity contrast, whereas the other has a low intensity contrast). For a given synthetic image, using additive Gaussian noise we control the level of noise. We generate the scale-space representation through convolution with both the DSS kernel and the SG kernel. Then, we observe whether the local extrema of each scale-space representation are not enhanced, i.e. whether the intensity value of the local maxima (minima) does not increase (decrease) as the scale parameter gradually increases.

The experimental results show that the intensity values of the local maxima (minima) in the scale-space representation generated by both DSS kernel and the SG kernel do not increase (decrease) as the scale parameter increases. The level of noise and the intensity contrast can influence the shape of convergence of local extrema, however, they do not affect the principal non-enhancement behavior of the local extrema. Based on this result, as a consequence, it can be said that the DSS kernel as well as the SG kernel fulfill the non-enhancement requirement.

5.3 Accuracy of Edge Extraction

We attempt to examine the accuracy of edge extraction (based on the non-maximum suppression method subsequent to the gradient magnitude) using the DSS differencing kernels in comparison to using the SG differencing kernel. For evaluating the result of edge extraction, we use a synthetic image in order to identify easily its edge image (we call it the *edge atlas*). The accuracy of $T_{\triangle_{\text{even}}}$, $T_{\triangle_{\text{odd}}}$, and $SG_{\triangle}$ for edge extraction is assessed by measuring the error of extracted edges based on the edge atlas (see e.g. [12]). In concrete terms, we denote $P_{l,\text{edge-atlas}}$ and $P_{l,\text{extracted-edge}}$ as the edge loci of the edge atlas and those of the extracted edge image, respectively, where n_l corresponds to the total number of edge loci. We measure the error of edge extraction in global terms by

$$\overline{\psi} = \frac{1}{n_l} \sum_{l=1}^{n_l} \psi_l, \tag{15}$$

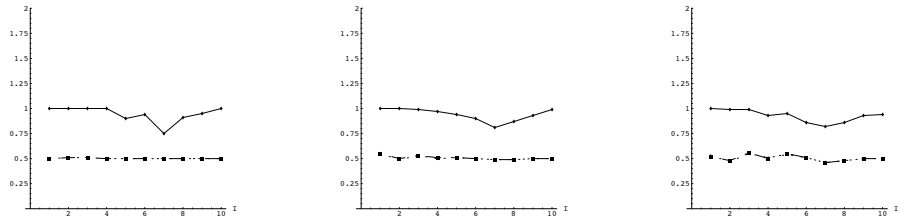
where $\psi_l = |P_{l,\text{edge-atlas}} - P_{l,\text{extracted-edge}}|$.

Rotational Invariance. The derived 2-D and 3-D DSS kernels are proven to be rotationally least asymmetric. We are now interested in the question whether $T_{\triangle_{\text{even}}}$ and $T_{\triangle_{\text{odd}}}$ as well as $SG_{\triangle}$ used for edge extraction are rotationally invariant. We examine how consistent the edge extraction result of each discrete differencing kernel is under gradual rotation of an edge line. For this, we provide a series of ten synthetic images as shown in Fig. 4, where a straight edge line gradually rotates.

The experimental results are given in Fig. 5, from which one can notice that i) for a given image edge extraction using $T_{\triangle_{\text{even}}}$ is less accurate than using $T_{\triangle_{\text{odd}}}$

Fig. 4. A series of ten synthetic images $\mathrm{I}_1 \cdots \mathrm{I}_{10}$: A straight edge line gradually rotates.

and $SG_{\triangle}$, ii) the edge extraction error using $T_{\triangle_{\mathrm{odd}}}$ is almost identical with that using $SG_{\triangle}$, and iii) the accuracy result of edge extraction using $T_{\triangle_{\mathrm{even}}}$ from I_1 to I_{10} is rather inconsistent compared with that using $T_{\triangle_{\mathrm{odd}}}$ and $SG_{\triangle}$, which definitely appears in the case of the noiseless images. This shows that $T_{\triangle_{\mathrm{even}}}$ is inferior to both $T_{\triangle_{\mathrm{odd}}}$ and $SG_{\triangle}$ with respect to rotational invariance for edge extraction.



(a) $\overline{\psi}_{n_0}$ (b) $\overline{\psi}_{n_5}$ (c) $\overline{\psi}_{n_{10}}$

Fig. 5. Error of edge extraction applied to the images given in Fig. 4. n_0, n_5, and n_{10} denote that the images are noiseless, weakly noisy, and strongly noisy, respectively. ($\overline{\psi}_{T_{\triangle_{\mathrm{even}}}}$, $\overline{\psi}_{T_{\triangle_{\mathrm{odd}}}}$, $\overline{\psi}_{SG_{\triangle}}$). Note that ★ and ■ almost overlap.

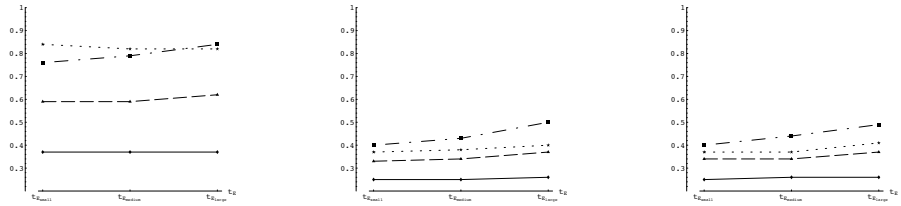
Steadiness from Adjacency. For the purpose of examining how steadily each discrete differencing kernel applied to an image (that contains closely adjacent edge structures) performs for edge extraction, we provide four synthetic images as shown in Fig. 6. Similarly to the case of rotational invariance, we apply $T_{\triangle_{\mathrm{even}}}$, $T_{\triangle_{\mathrm{odd}}}$, and $SG_{\triangle}$ to a given synthetic image and follow the procedure for evaluation of the edge extraction.

Fig. 7 illustrates the edge extraction results of three different types of the first-order differencing kernel applied to the weakly noisy images. From Fig. 7, one can observe that i) for a given image $\overline{\psi}_{T_{\triangle_{\mathrm{even}}}}$ is much larger than $\overline{\psi}_{T_{\triangle_{\mathrm{odd}}}}$ and $\overline{\psi}_{SG_{\triangle}}$, ii) the values of $\overline{\psi}_{T_{\triangle_{\mathrm{odd}}}}$ are similar to those of $\overline{\psi}_{SG_{\triangle}}$ on the whole, and iii) regardless of the type of the used discrete differencing kernel, SE1 is the steadiest image type of the adjacent edge structure in edge extraction (i.e. $\overline{\psi}_{SE1}$ is the smallest) and CE4 is the second steadiest one. Consequently, regarding

(a) SE1 (b) SE4 (c) CE1 (d) CE4

Fig. 6. Four synthetic images having closely adjacent edge structures.

the accuracy of edge extraction with respect to steadiness from adjacency, $T_{\triangle_{\mathrm{odd}}}$ and $SG_{\triangle}$ are superior to $T_{\triangle_{\mathrm{even}}}$ in general.



(a) $\overline{\psi}_{n_5,T_{\triangle_{\mathrm{even}}}}$ (b) $\overline{\psi}_{n_5,T_{\triangle_{\mathrm{odd}}}}$ (c) $\overline{\psi}_{n_5,SG_{\triangle}}$

Fig. 7. Error of edge extraction applied to the images given in Fig. 6. n_5 denotes that the images are weakly noisy (—◆— : SE1, ···★··· : SE4, ·–■–· : CE1, ––▲–– : CE4).

6 Conclusion

In this paper, we addressed the issue of a higher dimensional DSS formulation. For the purpose of analyzing the problem associated with the commonly used sampled Gaussian for approximating the continuous Gaussian, we derived the approximation error caused by the sampling as a function of the scale parameter. This analysis explicates that in general a sampled Gaussian with a small scale is not appropriate for approximating the continuous Gaussian. By developing the generalized 2-D and 3-D DSS formulation through a theoretical derivation, we made a step forward in investigating the problem of how to correctly approach the higher dimensional DSS theory. Furthermore, we investigated the properties of the derived DSS kernels and carried out a validation study with respect to

both smoothing and differentiation performance. Our investigation as well as the experimental results of the validation study show that the derived DSS kernel does not only match the performance of the SG kernel but also clearly exhibits superior performance with respect to both smoothing and differentiation.

Future work will include the investigation of the performance of higher-order differencing operators for feature extraction in higher dimensions. Also, a validation study with respect to the derived 3-D DSS kernel has to be carried out. Furthermore, it would be interesting to compare the derived DSS kernels with the SG kernel as well as e.g. Florack's scheme shown in [5] with respect to the cost of computational load.

Acknowledgement

The financial support by DAAD(German Academic Exchange Service) to the first author is greatly acknowledged.

References

1. J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the Gaussian Kernel for Scale-Space Filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):26-33, 1986.
2. R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, 3rd edition, 2000.
3. N. Fliege. *Systemtheorie*. Teubner, 1991.
4. L. M. J. Florack, *Image Structure*, Kluwer Academic Publishers, 1997.
5. L. M. J. Florack. A Spatio-Frequency Trade-Off Scale for Scale-Space Filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(9):1050-1055, 2000.
6. B. Jähne. *Digitale Bildverarbeitung*. Springer, 4. völlig neubearbeitete Auflage, 1997.
7. J. J. Koenderink. The Structure of Images. *Biological Cybernetics*, 50:363-370, 1984.
8. J. Y. Lim. On the Role of the Gaussian Kernel in Edge Detection and Scale-Space Methods. Technical Report FBI-HH-B-230/01, Fachbereich Informatik, Universität Hamburg, Germany, 2001.
9. J. Y. Lim. On the Discrete Scale-Space Formulation. Technical Report FBI-HH-B-231/01, Fachbereich Informatik, Universität Hamburg, Germany, 2001.
10. J. Y. Lim. The Supplemented Discrete Scale-Space Formulation. Technical Report FBI-HH-B-312/02, Fachbereich Informatik, Universität Hamburg, Germany, 2002.
11. T. Lindeberg. Scale-Space for Discrete Signals. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(3):234-264, 1990.
12. D. W. Paglieroni. A Unified Distance Transform Algorithm and Architecture. *Machine Vision and Applications*, 5(1):47-55, 1992.
13. L. Remaki and M. Cheriet. KCS-New Kernel Family with Compact Support in Scale-Space: Formulation and Impact. *IEEE Trans. on Image Processing*, 9(6):970-981, 2000.
14. J. Sporring, M. Nielsen, L. M. J. Florack, and P. Johansen. *Gaussian Scale-Space Theory*. Kluwer Academic Publishers, 1997.
15. L. Wu and Z. Xie. Scaling Theorems for Zero-Crossings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):46-54, 1990.
16. A. P. Witkin. Scale-Space Filtering. In *Proc. of 8th Int. Joint Conf. Artificial Intelligence, Karlsruhe*, 1019-1021, 1983.