Theoretical Foundations of Gaussian Convolution by Extended Box Filtering

Pascal Gwosdek[1], Sven Grewenig[1], Andrés Bruhn[2], and Joachim Weickert[1]

[1] Mathematical Image Analysis Group, Dept. of Mathematics and Computer Science,
Campus E1.1, Saarland University, 66041 Saarbrücken, Germany.
{gwosdek,grewenig,weickert}@mia.uni-saarland.de

[2] Vision and Image Processing Group,
Cluster of Excellence Multimodal Computing and Interaction,
Saarland University, Campus E 1.1, 66041 Saarbrücken, Germany
bruhn@mmci.uni-saarland.de

Abstract. Gaussian convolution is of fundamental importance in linear scale-space theory and in numerous applications. We introduce iterated extended box filtering as an efficient and highly accurate way to compute Gaussian convolution. Extended box filtering approximates a continuous box filter of arbitrary non-integer standard deviation. It provides a much better approximation to Gaussian convolution than conventional iterated box filtering. Moreover, it retains the efficiency benefits of iterated box filtering where the runtime is a linear function of the image size and does not depend on the standard deviation of the Gaussian. In a detailed mathematical analysis, we establish the fundamental properties of our approach and deduce its error bounds. An experimental evaluation shows the advantages of our method over classical implementations of Gaussian convolution in the spatial and the Fourier domain.

Keywords: Gaussian scale-space, box filter, image processing, computer vision

1 Introduction

Convolution with a Gaussian is one of the most widely used linear filter operations in signal and image processing. It forms the backbone of Gaussian scale-space theory [4,9,13] which has been introduced in Japanese and English papers of Iijima [8] long before it became popular in the western world by Witkin's work [16]. The strong regularisation properties of Gaussian convolution render the filtered signal infinitely times differentiable and stabilise the numerical evaluation of higher order derivatives. Gaussian convolution is inevitable for the detection of edges [2,11] and interest points [6,10] that play a central role in computer vision. The rapid decay properties of the Gaussian both in the spatial and the Fourier domain and the fact that it is the only filter that is rotationally invariant and separable under convolution make Gaussian convolution a perfect low-pass filter in linear systems theory.

Many applications require an accurate and efficient implementation of Gaussian convolution in order to ensure the high quality of the results, to meet runtime requirements, or even to guarantee convergence. However, this can be challenging: It comes down to a convolution of the input signal with a kernel function with infinite support. The m-dimensional Gaussian kernel

$$K_\sigma(\boldsymbol{x}) \;=\; \frac{1}{(2\pi\sigma^2)^{\frac{m}{2}}}\;\exp\left(-\frac{|\boldsymbol{x}|^2}{2\cdot\sigma^2}\right) \tag{1}$$

of standard deviation σ has a characteristic 'bell curve' shape which drops off rapidly towards $\pm\infty$. This is why in practice one often applies a discrete convolution with a sampled and renormalised kernel that is cut off at $n\cdot\sigma$. However, this method becomes inefficient for large σ, as the number of operations grows linearly in the number of samples of both the signal and the kernel. A more efficient alternative for those cases is the computation as a point-wise multiplication in the frequency domain [1]. To this end, a Fourier transform is applied to both the kernel and the signal, the multiplication is performed, and the result is transformed back into the spatial domain. Since the Gaussian kernel in the frequency domain can immediately be evaluated, this method reduces to two fast Fourier transforms, and one point-wise multiplication.

Although these spatial and Fourier-based implementations are the most popular algorithms for Gaussian convolution, and their trade-offs are well investigated [5], there are also further alternatives: Approximations with recursive filters [3,17] offer a runtime behaviour that scales linearly in the number of pixels. However, these filters require a special boundary treatment and a higher implementational effort than other methods which poses additional challenges [14]. Since Gaussian scale-space is equivalent to evolving the image under a homogeneous diffusion problem, one can also implement Gaussian convolution with efficient numerical methods for partial differential equations, e.g. with implicit finite difference schemes [7]. Unfortunately, this requires the fast solution of linear systems of equations which is also a nontrivial task. Gaussian convolution can also be approximated by discrete convolution with binomial kernels. They have a finite support and offer some interesting properties from an implementational viewpoint, but do not allow to approximate Gaussians with arbitrary standard deviations. This can constitute a drawback in scale-space applications which aim at representations at arbitrary scales.

A simple but extremely fast discrete approximation of Gaussian smoothing can be achieved by convolution with iterated box filters [15]. A box filter uses a normalised kernel with identical coefficients within its finite support. By the central limit theorem, a sufficiently high number of iterations with a box filter approximates a Gaussian arbitrarily well. However, this has the same drawback as convolution with binomial kernels: It introduces a quantisation to the range of standard deviations that can be approximated.

In our paper we address this problem. We advocate a modification of the box filter that is based on a new discretisation of the continuous box kernel. In particular, we concentrate on establishing important properties of the resulting *extended box filter*: It combines the simplicity and algorithmic efficiency of the conventional box filter with a good approximation of theoretic properties of Gaussian filtering. In an experimental evaluation, we show that the extended box filter approximates the Gaussian filter significantly better than a classical box filter and offers advantages over spatial and Fourier-based approximations of Gaussian convolution. Moreover, our method introduces only marginal runtime overheads over classical box filtering.

Our paper is structured as follows. In Section 2, we first recapitulate the basic notations and definitions of conventional box filtering. Thereafter, we present our new method in Section 3. After an experimental evaluation in Section 4, we conclude with a summary in Section 5.

2 Conventional Box Filtering

Box filters are usually defined in a purely discrete context. However, in order to derive a new discretisation in this paper, we start with a short review of a continuous definition:

Definition 1. *A continuous* box filter B_Λ *with a real-valued length* $\Lambda \in \mathbb{R}^+ := \{a \in \mathbb{R} \ : \ a > 0\}$ *is a convolution*

$$(B_\Lambda * f)(x) \ := \int\limits_{-\infty}^{\infty} B_\Lambda(x-y) \cdot f(y)\, dy \tag{2}$$

of a signal f *with a* box kernel

$$B_\Lambda(x) \ := \begin{cases} \frac{1}{\Lambda}, & x \in (-\lambda, \lambda) \\ 0, & else \end{cases} \tag{3}$$

for $x \in \mathbb{R}$ *and* $\Lambda = 2\lambda$.

In the literature, one usually finds the continuous length Λ being rounded to the closest odd integer L [15]:

Definition 2. *A* discrete box filter B_L *of length* $L = h(2l+1)$, $l \in \mathbb{N}_0$, *and sampled at an equidistant grid of spacing* $h > 0$ *is a convolution*

$$(B_L * f)(hk) \ := \sum_{m \in \mathbb{Z}} B_L(h(k-m)) \cdot f(hm) \tag{4}$$

of a signal f *with a* discrete box kernel

$$B_L(hk) \ := \begin{cases} \frac{h}{L}, & -l \leqslant k \leqslant l \\ 0, & else \end{cases} \tag{5}$$

for $k \in \mathbb{Z}$.

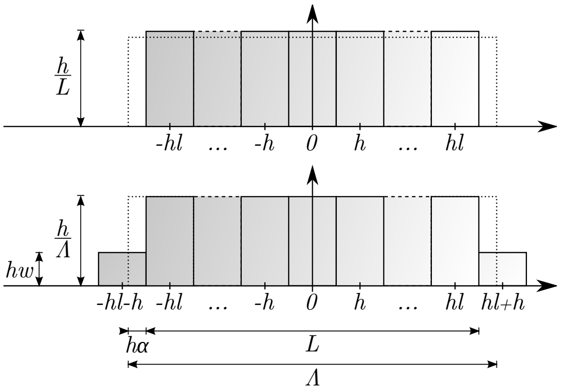

Fig. 1. Visualisation of box kernels. **Top**: Continuous box kernel B_Λ (**dotted**) and its conventional discrete approximation B_L. **Bottom**: Corresponding discrete extended box kernel E_Λ.

An illustration of this construction is depicted in Figure 1. Note that we introduce an arbitrary grid spacing h, and couple the length L to a multiple of this distance. For $h \to 0$, B_L thus approaches B_Λ (cf. Definition 1). If we set $h = 1$, we obtain the formulation in [15].

On discrete data, it can be implemented very efficiently in an iterative 'sliding window' manner, i.e.

$$(B_L * f)_i \;=\; (B_L * f)_{i-1} + \frac{h}{L}\left(f_{i+l} - f_{i-l-1}\right), \tag{6}$$

with $(\cdot)_k$ or f_k denoting the discrete value at sampling point hk. After the initialisation of the first sample, the method needs one multiplication and two additions per pixel and dimension, independent of the size of the kernel. Thus, it enjoys a linear complexity in time.

A d-fold convolution of the kernel with the signal approximates a Gaussian convolution. This removes artefacts that arise from the piecewise linearity of the box kernel, as well as from the lack of a rotational invariance property in the multi-dimensional case. The resulting operation is equivalent to the convolution with a C^{d-1}-continuous kernel B_L^d of variance $\sigma^2(B_L^d)$ [15]:

$$\sigma^2(B_L^d) \;=\; d\,\frac{L^2-1}{12}. \tag{7}$$

Note that this formula only allows a *discrete* set of standard deviations to be chosen. In the literature, it is suggested to handle this problem by a series of box filters of different length [15]. Unfortunately, this idea does not solve the problem: By practical considerations, d is typically chosen from the range $\{3, 4, 5\}$, such that the distance between admissible σ cannot be reduced arbitrarily. Moreover, the kernel resulting from a convolution of box kernels with different lengths does not fulfil the continuity properties mentioned above.

In contrast to this suggestion, we are now going to derive a better discretisation of the continuous formulation which does not have this problem by construction. Still, it possesses all advantages of the discrete box filter.

3 Extended Box Filter

Our goal is now to find a better discretisation E_Λ of the continuous box filter B_Λ than is given by the conventional discrete approximation B_L. In doing so, we focus in particular on the following criteria:

1. E_Λ must be continuous over Λ to allow kernels with arbitrary variance.
2. For $\Lambda = L \in \mathbb{N}_{\text{odd}}$, it must equal the discrete box filter B_L of length L.
3. For $h \to 0$, it must approach the continuous case, i.e. $\lim\limits_{h\to 0} \sigma^2(E_\Lambda) = \sigma^2(B_\Lambda)$.

To this end, we decompose Λ into an integer part and a real-valued remainder:

$$\Lambda \;=\; h(2l+1+2\alpha) \;=\; L + 2h\alpha \tag{8}$$

such that $0 \leqslant \alpha < 1$ and $l \in \mathbb{N}_0$. With this formalism, we are now able to set up an ‘extended’ variant of the discrete box filter:

Definition 3. *An* extended box filter E_Λ *with a real-valued length* $\Lambda \in \mathbb{R}^+$ *and discretised on a uniform grid of spacing* $h > 0$ *is a convolution*

$$(E_\Lambda * f)(hk) \;:=\; \sum_{m\in\mathbb{Z}} E_\Lambda(h(k-m)) \cdot f(hm) \tag{9}$$

of a signal f *with an* extended box kernel

$$E_\Lambda(hk) \;:=\; \begin{cases} \frac{h}{\Lambda}, & -l \leqslant k \leqslant l \\ hw, & k \in \{-(l+1), l+1\} \\ 0, & \textit{else} \end{cases} \tag{10}$$

with

$$l \;:=\; \left\lfloor \frac{\Lambda}{2h} - \frac{1}{2} \right\rfloor, \qquad w \;:=\; \frac{1}{2}\left(\frac{1}{h} - \frac{2l+1}{\Lambda}\right), \tag{11}$$

and $k \in \mathbb{Z}$. $\lfloor x \rfloor$ *denotes the so-called floor function, which computes the largest integer not greater than* $x \in \mathbb{R}$.

Both constraints in (11) are necessary in order to ensure that all weights sum up to 1. A visualisation of the extended box kernel (10) is depicted in Figure 1. It is immediately clear that our new filter preserves many advantages of the original box filter. It is separable in space, and an efficient ‘sliding-window’ implementation is still possible and beneficial: Apart from the first value in a row, only four additions and two multiplications are needed per pixel and dimension (since both weighting factors are constants):

$$\begin{aligned}(E_\Lambda * f)_i \;=\; (E_\Lambda * f)_{i-1} \;&+\; \left(\frac{h}{\Lambda} - hw\right)(f_{i+l} \quad - f_{i-l-1}) \\ &+ \qquad hw \;\; (f_{i+l+1} - f_{i-l-2})\,. \end{aligned} \tag{12}$$

This means, the computational complexity of a box filtering step is in $\mathcal{O}(n)$ in the number of pixels, and is thus in particular independent of the length of the chosen box kernel. Let us now discuss some mathematical properties of our construction. First of all, we immediately see that w depends proportionally on α (cf. Figure 1):

$$w \;=\; \frac{1}{2h}\left(1 - \frac{(2l+1)h}{\Lambda}\right) \;=\; \frac{1}{2h}\left(1 - \frac{\Lambda - 2\alpha h}{\Lambda}\right) \;=\; \frac{1}{2h}\cdot\frac{2\alpha h}{\Lambda} \;=\; \frac{\alpha}{\Lambda}. \tag{13}$$

Using this equivalence, we can formulate the variance of E_Λ by considering the components of Λ only. Like for the conventional box filter, we regard the more general case for a convolution kernel that corresponds to a d-fold application of a single extended box kernel E_Λ:

Theorem 1. *The variance $\sigma^2(E_\Lambda^d)$ of a d-fold iterated extended box kernel is given by*

$$\sigma^2(E_\Lambda^d) \;=\; \frac{dh^3}{3\Lambda}\left(2l^3 + 3l^2 + l + 6\alpha(l+1)^2\right)\ . \tag{14}$$

Proof. By symmetry considerations, we see that the expectation value of E_Λ is zero. For the variance $\sigma^2(E_\Lambda)$ of one (non-iterated) box kernel, it follows

$$\begin{aligned}\sigma^2(E_\Lambda) &= \sum_{k=-(l+1)}^{l+1} E_\Lambda(hk)\cdot(hk-0)^2 \\ &= \sum_{k=-l}^{l} \frac{h}{\Lambda}(hk)^2 + hw\left(-(hl+h)\right)^2 + hw(hl+h)^2 \\ &= \frac{2h^3}{\Lambda}\sum_{k=1}^{l} k^2 + 2h^3 w(l+1)^2 \\ &\overset{(13)}{=} \frac{h^3}{3\Lambda}\left(2l^3 + 3l^2 + l + 6\alpha(l+1)^2\right)\ .\end{aligned}$$

From probability theory, we obtain the variance $\sigma^2(E_\Lambda^d)$ for the iterated extended box kernel as the sum of single variances. This concludes the proof. □

For $h = 1$ and $\Lambda = 2l+1 \in \mathbb{N}_{\text{odd}}$, i.e. $\alpha = 0$, this is just a generalisation of Equation (7). This means that the extended box filter falls back to the notion of the conventional box filter in these cases:

Theorem 2. *The extended box kernel E_Λ constitutes a generalisation of the discrete box kernel B_L for the case $h = 1$, i.e. $E_L = B_L$ for $L \in \mathbb{N}_{odd}$ and*

$$\forall L \in \mathbb{N}_{odd} : \lim_{\Lambda \to L^+} \sigma^2(E^d_\Lambda) = \sigma^2(B^d_L) \quad \textit{and} \quad \lim_{\Lambda \to (L+2)^-} \sigma^2(E^d_\Lambda) = \sigma^2(B^d_{L+2}). \tag{15}$$

Proof. It is clear that $E_L = B_L$ for $L \in \mathbb{N}_{\text{odd}}$, because in this case we get $\alpha = 0$ and $w = 0$. Thus, it immediately follows that $\lim_{\Lambda \to L^+} \sigma^2(E^d_\Lambda) = \sigma^2(B^d_L)$. So, it remains to show the case $\Lambda \to (L+2)^-$, for which we first consider a single extended box kernel E_Λ:

$$\begin{aligned}
\lim_{\Lambda \to (L+2)^-} \sigma^2(E_\Lambda) &= \lim_{\Lambda \to (L+2)^-} \frac{1}{3\Lambda} \left(2l^3 + 3l^2 + l + 3(\Lambda - L)(l+1)^2\right) \\
&= \frac{1}{3(L+2)} (2l^3 + 9l^2 + 13l + 6) \\
&= \frac{1}{3(2l+3)} (2l+3)(l^2 + 3l + 2) \\
&= \frac{(L+2)^2 - 1}{12} \ ,
\end{aligned}$$

where we have used that $\alpha = \frac{\Lambda - L}{2}$ and $L = 2l + 1$. It follows immediately that

$$\lim_{\Lambda \to (L+2)^-} \sigma^2(E^d_\Lambda) \ = \ \sigma^2(B^d_\Lambda).$$

This shows that E^d_Λ is a consistent generalisation of B^d_L with respect to Λ. □

Now that we have shown that the extended box filter extends the previous discrete definition, we want to show that it is a good discretisation of the continuous box filter we are about to approximate:

Theorem 3. *The extended box kernel E_Λ is a suitable discretisation of a box kernel B_Λ in the continuous domain, i.e. for d-fold application,*

1. *its variance approximates the continuous analogue arbitrarily well:*

$$\lim_{h \to 0} \sigma^2(E^d_\Lambda) = \sigma^2(B^d_\Lambda), \qquad \textit{and} \tag{16}$$

2. *the order of consistency is $\mathcal{O}(h^2)$.*

Proof. We can deduce an approximation of the continuous setting by computing the limit of $\sigma^2(E^d_\Lambda)$ for the grid spacing $h \to 0$. Since we are interested in the order of consistency, we must consider the variance in (14) and rewrite it:

$$\begin{aligned}
\sigma^2(E^d_\Lambda) &= \frac{(2hl)^3}{12\Lambda} + dh\frac{(2hl)^2}{4\Lambda} + dh^2\frac{2hl}{6\Lambda} + 2dh^3\frac{\alpha}{\Lambda}(l^2 + 2l + 1) \\
&= d\frac{(2hl)^3}{12\Lambda} + dh(1+2\alpha)\frac{(2hl)^2}{4\Lambda} + dh^2(1+12\alpha)\frac{2hl}{6\Lambda} + dh^3\frac{2\alpha}{\Lambda}.
\end{aligned}$$

```
Input:  Signal $u^0$, standard deviation $\sigma$, iterations $d$.
Output: Signal $u^d := E^d_\Lambda * u^0$

$l$ ← largest integer such that $\sigma^2(B^d_L) \le \sigma^2$    (by (7))
$\alpha \leftarrow (2l+1)\frac{l(l+1) - \frac{3\sigma^2}{d}}{6(\frac{\sigma^2}{d} - (l+1)^2)}$    (by $\sigma^2(E^d_\Lambda) \le \sigma^2$, (8), and (14))
$w \leftarrow \frac{\alpha}{2l+1+2\alpha}$ , $\hat{w} \leftarrow \frac{1-\alpha}{2l+1+2\alpha}$    (by (8) and (13))

For all $j \in \{1,\dots,d\}$
  Compute $u^j$[0] (for the first pixel)
  For all i > 0
    Compute $u^j$[i] ← $u^j$[i-1] + $w$ · ($u^{j-1}$[i+$l$+1] - $u^{j-1}$[i-$l$-2])
                                  + $\hat{w}$ · ($u^{j-1}$[i+$l$  ] - $u^{j-1}$[i-$l$-1])
```

Fig. 2. Algorithm for 1-D extended box filtering. Boundaries can be handled on-the-fly.

Now we replace $2hl$ by $\Lambda - (1+2\alpha)h$ and get for the first three terms:

$$\begin{aligned}
d\,\frac{(\Lambda-(1+2\alpha)h)^3}{12\Lambda} &= \frac{d\Lambda^2}{12} - \frac{dh}{4}(1+2\alpha)\Lambda + \frac{dh^2}{4}(1+2\alpha)^2 + \mathcal{O}(h^3),\\
dh(1+2\alpha)\frac{(\Lambda-(1+2\alpha)h)^2}{4\Lambda} &= \frac{dh}{4}(1+2\alpha)\Lambda - \frac{dh^2}{2}(1+2\alpha)^2 + \mathcal{O}(h^3),\\
dh^2(1+12\alpha)\frac{\Lambda-(1+2\alpha)h}{6\Lambda} &= \frac{dh^2}{6}(1+12\alpha) + \mathcal{O}(h^3).
\end{aligned}$$

Finally, this yields

$$\sigma^2(E^d_\Lambda) = \frac{d\Lambda^2}{12} - h^2 \cdot \frac{d}{12}\left(12\alpha^2 - 12\alpha + 1\right) + \mathcal{O}(h^3).$$

Thus, the consistency order is $\mathcal{O}(h^2)$ and we can state that

$$\lim_{h\to 0} \sigma^2(E^d_\Lambda) = d\frac{\Lambda^2}{12} = d\int_{-\frac{\Lambda}{2}}^{\frac{\Lambda}{2}} \frac{1}{\Lambda}\cdot x^2 \mathrm{d}x = d\int_{-\infty}^{\infty} B_\Lambda(x) x^2 \mathrm{d}x = \sigma^2(B^d_\Lambda).$$

□

4 Experiments

In order to investigate the properties of the extended box filter on real data, we have implemented the algorithm for application on images. Technically, this means we are dealing with 2-D images $f \in \Omega \subset \mathbb{R}^2$, and assume reflecting

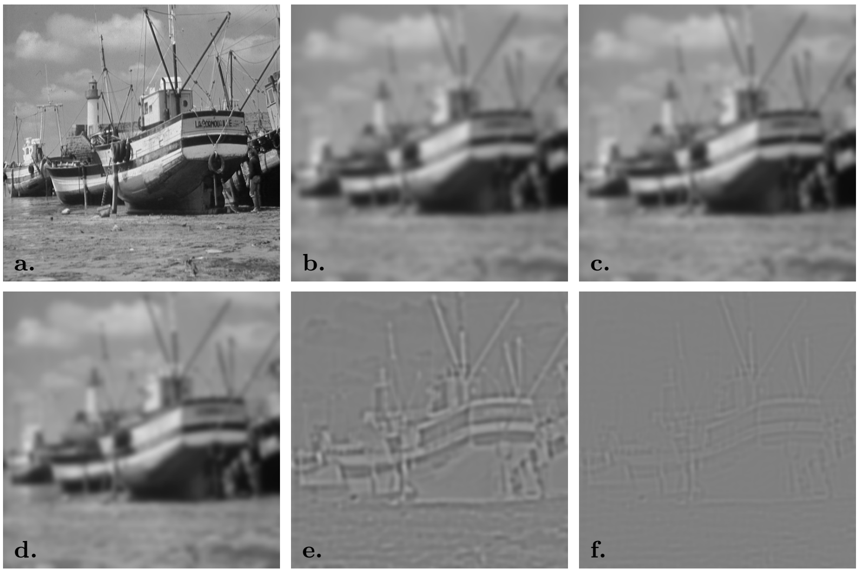

Fig. 3. Visual quality for *Boat*, 512×512 pixels. **a:** Original, **b:** conventional and **c:** extended box filtering with $d = 3$, $\sigma = 5.0$, **d:** Discrete Gaussian filtering with $\sigma = 5.0$ (truncated at 10σ), and **e,f:** differences of b and c to d, respectively, scaled by a factor 10 to increase visibility. 50% grey means the error is zero.

boundary conditions to preserve the average grey value. Using the separability of the kernel, we apply a 'sliding window' technique in both directions (cf. Figure 2). This operation is highly parallel, and can thus be significantly accelerated by the streaming SIMD extension mechanism (SSE) of modern desktop processors, by use of all CPU cores, and by graphics processors.

4.1 Qualitative Gain

Our aim in designing the extended box filter is to propose a fast but accurate way to perform Gaussian convolution for arbitrary standard deviations. Consequently, we are interested in the accuracy of the proposed method.

To evaluate the accuracy, we use the well-known *Boat* test image from the USC SIPI database (cf. Figure 3a), and convolve it with discrete box kernels. These results are then compared to a ground truth obtained by a convolution with a discretised Gaussian kernel that has been truncated at 10σ and renormalised. Please note that this ground truth is also subject to discretisation artefacts and may not exactly reflect the desired solution. A more complicated alternative to this implementation has been proposed in [12], but this variant also suffers from similar truncation problems. Finally, let us note that we chose the *Boat* test image as a good representative for many real-world examples, since it contains many different frequencies and both homogeneous and textured regions.

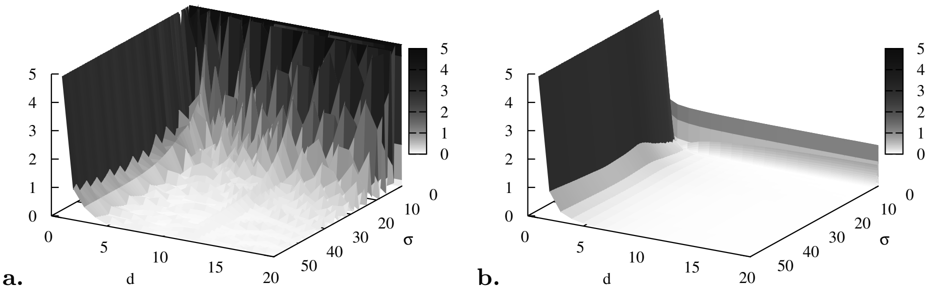

Fig. 4. Plot of the mean square error to discrete Gaussian convolution on *Boat*, 512 × 512 pixels, depending on σ and d. **a:** Conventional box filter. **b:** Extended box filter.

In the first part of our experiment, we use conventional box kernels and a varying number of iterations d, and compare these results to the reference solution. Instead of focussing on one specific standard deviation σ, we evaluate many different values. As an error measure for equivalence, we use the mean square error (MSE) given by

$$MSE(a,b) \; = \; \frac{1}{N}\sum_{i=1}^{N}(a_i - b_i)^2, \tag{17}$$

where N describes the number of pixels. The results for this experiment are given in Figure 4a. For large σ, we see that a box filter of order $d = 5$ is already sufficient to approximate the Gaussian very well. However, one also realises that small standard deviations cannot be represented well at all. This effect is caused by the integer length of the box kernel, and re-occurs for larger d and larger σ for similar reasons.

In the second part, we repeated the same experiment with the proposed kernel. This is shown in Figure 4b. Compared to the conventional box filter, the novel approach attenuates errors much stronger. For any σ, an order of $d = 5$ yields almost identical results to Gaussian filtering. This justifies our model as a qualitatively equal alternative.

To conclude this experiment, we compare the visual quality of both approaches. Figure 3 depicts a sample output of both methods under a standard deviation $\sigma = 5.0$, and further shows the desired result as given by Gaussian convolution. Albeit the visual quality differences are relatively low, the difference images show that our extended box filter performs much better than the conventional box filtering.

4.2 Runtime

In the last experiment, we are interested in the tradeoff between the accuracy and the runtime of extended box filtering compared to other techniques. To this end, we convolve the *Boat* test image using a discrete Gaussian truncated at 3σ,

Table 1. CPU runtime t in milliseconds vs. mean square error (MSE) between the result and the ground truth for different techniques on *Boat* (512×512 pixels).

	$\boldsymbol{\sigma = 0.5}$		$\boldsymbol{\sigma = 5.0}$		$\boldsymbol{\sigma = 25.0}$	
	MSE	$\boldsymbol{t}$	**MSE**	$\boldsymbol{t}$	**MSE**	$\boldsymbol{t}$
Truncated Gaussian	0.000	8	0.001	45	0.007	148
FFT-based	1.032	148	0.000	148	0.000	148
Conventional box	9.580	0	1.400	26	0.154	27
Extended box	0.030	41	0.051	43	0.098	43

an FFT-based approach, a conventional, and an extended box filter (both with $d = 5$). Runtimes were acquired on a single-core 3.2 GHz Pentium 4 with 2MB L2 cache and 2 GB RAM.

Table 1 shows the results of this experiment. The truncated Gaussian convinces for small standard deviations, but scales linearly in σ such that this method becomes infeasible for large σ. Although the runtime of all remaining methods is independent from the standard deviation, box filters have a clear advantage if we consider larger images: While they only scale linearly in the number of pixels n, the FFT-based methods have a complexity of $\mathcal{O}(n \log(n))$. In return, the FFT-based approach offers a much better approximation quality for large σ. In this context, the extended box filter is a good tradeoff between classical box filtering and the FFT-based approaches: It provides a convincing approximation quality for all standard deviations at a slightly higher runtime than a classical box filter.

5 Summary

In view of the omnipresence of Gaussian convolution in scale-space theory and its numerous applications in image processing and computer vision, it is surprising that one can still come up with novel algorithms that are extremely simple and offer a number of advantages. In our paper we have shown that a small modification of classical box filtering leads to an extended box filter which can be iterated in order to approximate Gaussian convolution with high accuracy and high efficiency. In contrast to classical box filtering, it does not suffer from the restriction that only a distinct set of standard deviations of the Gaussian are allowed. Although the main focus of our paper is on establishing the essential mathematical properties of extended box filtering, we have also presented experiments that illustrate the advantages over spatial and Fourier-based implementations of Gaussian convolution. In our ongoing research we will perform a more extensive evaluation with a large number of alternative implementations, taking also into account the potential of modern parallel hardware such as GPUs.

Acknowledgements

Our research was partly funded by the Cluster of Excellence “Multimodal Computing and Interaction”, and by the Deutsche Forschungsgemeinschaft under project We2602/7-1. This is gratefully acknowledged.

References

1. Bracewell, R.N.: The Fourier transform and its applications. McGraw-Hill, third edn. (Jun 1999)
2. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 679–698 (1986)
3. Deriche, R.: Fast algorithms for low-level vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 78–87 (1990)
4. Florack, L.: Image Structure, Computational Imaging and Vision, vol. 10. Kluwer, Dordrecht (1997)
5. Florack, L.: A spatio-frequency trade-off scale for scale-space filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(9), 1050–1055 (Sep 2000)
6. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data. pp. 281–305. Interlaken, Switzerland (Jun 1987)
7. Gourlay, A.R.: Implicit convolution. Image and Vision Computing 3, 15–23 (1985)
8. Iijima, T.: Theory of pattern recognition. Electronics and Communications in Japan pp. 123–134 (Nov 1963), in English
9. Lindeberg, T.: Scale-Space Theory in Computer Vision. Kluwer, Boston (1994)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
11. Marr, D., Hildreth, E.: Theory of edge detection. Proceedings of the Royal Society of London, Series B 207, 187–217 (1980)
12. Norman, E.: A discrete analogue of the Weierstrass transform. Proceedings of the American Mathematical Society 11(596–604) (1960)
13. Sporring, J., Nielsen, M., Florack, L., Johansen, P. (eds.): Gaussian Scale-Space Theory, Computational Imaging and Vision, vol. 8. Kluwer, Dordrecht (1997)
14. Triggs, B., Sdika, M.: Boundary conditions for Young - van Vliet recursive filtering. IEEE Transactions on Signal Processing 54(5), 1–2 (May 2006)
15. Wells, W.M.: Efficient synthesis of Gaussian filters by cascaded uniform filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(2), 234–239 (Mar 1986)
16. Witkin, A.P.: Scale-space filtering. In: Proc. Eighth International Joint Conference on Artificial Intelligence. vol. 2, pp. 945–951. Karlsruhe, West Germany (August 1983)
17. Young, I.T., van Vliet, L.J.: Recursive implementation of the Gaussian filter. Signal Processing 44, 139–151 (Jan 1995)