IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2016–02–02.
Submitted on 2014–04–28, accepted on 2015–12–23.
ISSN 2105–1232 © 2016 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
http://dx.doi.org/10.5201/ipol.2016.117

Computing an Exact Gaussian Scale-Space

Ives Rey-Otero[1], Mauricio Delbracio[2]

[1] CMLA, ENS Cachan, France (ives.rey-otero@cmla.ens-cachan.fr)
[2] ECE, Duke University, USA (mauricio.delbracio@duke.edu)

Abstract

Gaussian convolution is one of the most important algorithms in image processing. The present work focuses on the computation of the Gaussian scale-space, a family of increasingly blurred images, responsible, among other things, for the scale-invariance of SIFT, a popular image matching algorithm. We discuss and numerically analyze the precision of three different alternatives for defining a discrete counterpart to the continuous Gaussian smoothing operator. This study is focused on low blur levels, that are crucial for the scale-space accuracy.

Source Code

An ANSI C source code implementation of the described algorithms is accessible at the IPOL web page of this article[1], together with an on-line demo.

Keywords: blur; scale-space; Gaussian convolution; DFT; Lindeberg

1 Introduction

The Gaussian smoothing operator is one of the most popular tools used in digital image processing. This classic operator has been extensively used, either as a fast pre-process to increase noise robustness before applying another algorithm, or as the fundamental operator in scale-space theory [8, 10, 12, 14].

Let $u(\mathbf{x})$ be a continuous image defined for every $\mathbf{x} = (x, y) \in \mathbb{R}^2$. The continuous Gaussian smoothing operator is defined as the convolution operator on $\mathbb{R}^2$ with the isotropic Gaussian function of integral equal to 1,

$$G_\sigma u(\mathbf{x}) := \int_{\mathbb{R}^2} G_\sigma(\mathbf{x}')u(\mathbf{x}-\mathbf{x}')d\mathbf{x}', \qquad \text{with } G_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2}e^{-\frac{|\mathbf{x}|^2}{2\sigma^2}},$$

where the Gaussian function is parameterized by its standard deviation σ.

This operator is the cornerstone of several image processing algorithms particularly used for building the scale-space, a multi-scale image representation. The *rationale* is that the Gaussian function is the only function that satisfies the following properties [1, 2, 9, 10, 14, 13]:

[1] http://dx.doi.org/10.5201/ipol.2016.117

1. **Linearity.** $G_\sigma(\lambda u(\mathbf{x}) + \mu v(\mathbf{x})) = \lambda G_\sigma u(\mathbf{x}) + \mu G_\sigma v(\mathbf{x})$ for any real λ, μ;

2. **Shift invariance.** If $T_{\boldsymbol{\tau}} u(\mathbf{x}) := u(\mathbf{x} - \boldsymbol{\tau})$ denotes the translation of parameter $\boldsymbol{\tau}$, then $G_\sigma(T_{\boldsymbol{\tau}} u)(\mathbf{x}) = T_{\boldsymbol{\tau}}(G_\sigma u)(\mathbf{x})$;

3. **Scale invariance.** If $H_\lambda u(\mathbf{x}) := u(\lambda \mathbf{x})$ denotes an expansion by a factor λ^{-1}, then $G_\sigma(H_\lambda u)(\mathbf{x}) = H_\lambda(G_{\sigma'} u)(\mathbf{x})$ with $\sigma' = \lambda\sigma$;

4. **Rotation invariance.** If $R_\theta u(\mathbf{x}) := u(R_\theta \mathbf{x})$ denotes the rotation of angle $-\theta$, then $G_\sigma(R_\theta)u(\mathbf{x}) = R_\theta(G_\sigma)u(\mathbf{x})$;

5. **Nonnegativity.** $G_\sigma(\mathbf{x}) \geq 0,\ \forall(\mathbf{x}, \sigma) \in \mathbb{R}^2 \times \mathbb{R}_+$;

6. **Semi-group.** $G_{\sigma_2}(G_{\sigma_1} u)(\mathbf{x}) = G_{\sqrt{\sigma_1^2 + \sigma_2^2}} u(\mathbf{x})$.

Additionally, it can be easily checked that if u is continuous and bounded, then $(t, \mathbf{x}) \mapsto G_{\sqrt{2t}} u(\mathbf{x})$ is the solution of the heat diffusion equation $\partial v / \partial t = \Delta v$ with initial condition $v(0, \mathbf{x}) = u(\mathbf{x})$ (see [6], Chapter 2).

What is the discrete counterpart of this continuous operator? Could it be defined to satisfy the properties of the continuous Gaussian convolution? Despite its central role in image processing, the Gaussian convolution is generally crudely approximated by discrete convolutions or even box filters. Numerous algorithms have been proposed for approximating the Gaussian convolution in digital images. This work concentrates on three of the most relevant ones for the accurate computation of the Gaussian scale-space, namely, the Fourier based convolution, the discrete convolution with a sampled Gaussian function and Lindeberg's discrete scale-space smoothing [10]. These methods can be described either as approximations of the continuous Gaussian convolution or as linear filters designed to satisfy some of the previously introduced properties expressed in the discrete framework. Other methods, not discussed in this work, include the use of recursive filters [15, 3] or the iteration of extended box filters [7]. They provide fast and accurate approximations of the Gaussian convolution for large σ values but they crudely approximate the Gaussian function for low values of σ (typically $\sigma \leq 1$), making them unsuitable for an accurate computation of the Gaussian scale-space. For a complete survey regarding speed and performance of the Gaussian convolution for large values of σ we refer the reader to [5].

In this work, we propose to use the semi-group property for measuring the accuracy of each of the analyzed methods. In particular, we test if multiple iterations of the same Gaussian convolution produce the same result as a single convolution with the blur level foretold by the semi-group property. The conclusions are straightforward. The only method that allows to compute accurately the Gaussian scale-space is the Fourier based convolution. The discrete convolution with samples from a Gaussian function is accurate only if the applied blur level is large enough to avoid aliasing artifacts (i.e. $\sigma > 0.8$). Although Lindeberg's smoothing method satisfies the semi-group property, it introduces a bias in the applied amount of blur, which is significantly lower. Evident though they are, these conclusions may have a strong impact on the conception and performance of algorithms using the Gaussian scale-space.

The rest of the article is organized as follows. Section 2 introduces the mathematical tools used to justify the Fourier based convolution. Each of the three discussed methods is explained in Section 3 where a detailed implementation with a mathematical interpretation is given. The algorithm's pseudocodes can be found in the Appendix. In Section 4 we present some numerical experiments and we finally conclude in Section 5.

2 Mathematical Preliminaries

2.1 Notations

In the sequel, $u_{k,l} \in \mathbb{R}$ for $k = 0, \dots, M-1$ and $l = 0, \dots, N-1$ denote the samples of a digital image of size $M \times N$. By a slight abuse of notation, we will denote this image by $(u_{k,l})$. We denote by $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ the floor and ceiling functions respectively.
The Fourier transform of $f \in L^1(\mathbb{R}^2)$ is the function $\hat{f}$, defined for all $(\xi, \eta) \in \mathbb{R}^2$ by

$$\hat{f}(\xi, \eta) = \int_{\mathbb{R}^2} f(x, y) e^{-i(x\xi + y\eta)} dx dy.$$

The Discrete Fourier Transform (DFT) of $(u_{k,l})$ is defined as the sequence

$$\tilde{u}_{m,n} = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} u_{k,l} e^{-\frac{2i\pi mk}{M}} e^{-\frac{2i\pi nl}{N}},$$

for $m = -\lfloor M/2 \rfloor, \dots, -\lfloor M/2 \rfloor + M - 1$ and $n = -\lfloor N/2 \rfloor, \dots, -\lfloor N/2 \rfloor + N - 1$.

The Inverse Discrete Fourier Transform (IDFT) of $(\tilde{u}_{m,n})$ with $m = -\lfloor M/2 \rfloor, \dots, -\lfloor M/2 \rfloor + M - 1$ and $n = -\lfloor N/2 \rfloor, \dots, -\lfloor N/2 \rfloor + N - 1$ is defined as the sequence

$$u_{k,l} = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\left(-\lfloor \frac{M}{2} \rfloor + M - 1\right)} \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\left(-\lfloor \frac{N}{2} \rfloor + N - 1\right)} \tilde{u}_{m,n} e^{\frac{2i\pi mk}{M}} e^{\frac{2i\pi nl}{N}}$$

for $k = 0, \dots, M-1$ and $l = 0, \dots, N-1$. The DFT and IDFT are inverse transformations: $\text{IDFT} \circ \text{DFT} = \text{Id}$ and $\text{DFT} \circ \text{IDFT} = \text{Id}$.

2.2 DFT and DCT Interpolations

A convenient continuous image model is to represent images as trigonometric polynomials, or equivalently, periodic band-limited functions. The limited bandwidth of camera lenses motivates this approach. The periodic extension of the signal is arbitrary, as is any other signal extension, but it is particularly convenient for Fourier interpolation.

We will say that P is a bi-dimensional trigonometric polynomial of degrees $\lfloor \frac{M}{2} \rfloor$ and $\lfloor \frac{N}{2} \rfloor$, and periodicities a and b, if and only if

$$P(x, y) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\left(-\lfloor \frac{M}{2} \rfloor + M - 1\right)} \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\left(-\lfloor \frac{N}{2} \rfloor + N - 1\right)} a_{m,n} e^{\frac{2i\pi mx}{a}} e^{\frac{2i\pi ny}{b}},$$

where $a_{m,n} \in \mathbb{C}$ for $m = -\lfloor \frac{M}{2} \rfloor, \dots, -\lfloor \frac{M}{2} \rfloor + M - 1$ and $n = -\lfloor \frac{N}{2} \rfloor, \dots, -\lfloor \frac{N}{2} \rfloor + N - 1$.
The following proposition, characterizes the polynomial coefficients that satisfy an interpolation criterion.

Proposition 1. *(The DFT Interpolation) There exists a unique trigonometric polynomial u of degrees $\lfloor \frac{M}{2} \rfloor$ and $\lfloor \frac{N}{2} \rfloor$, and of periodicities a and b, that satisfies the interpolation condition*

$$u\left(k\frac{a}{M}, l\frac{b}{N}\right) = u_{k,l}, \quad \text{for } k = 0, \dots, M-1 \text{ and } l = 0, \dots, N-1,$$

namely

$$u(x,y) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\left(-\lfloor \frac{M}{2} \rfloor + M - 1\right)} \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\left(-\lfloor \frac{N}{2} \rfloor + N - 1\right)} \tilde{u}_{m,n} e^{\frac{2i\pi mx}{a}} e^{\frac{2i\pi ny}{b}},$$

where the polynomial coefficients $\tilde{u}_{m,n}$ *are computed by the DFT of* $(u_{k,l})$.

From now on, we will consider without loss of generality that $a = M$ and $b = N$. This can be fulfilled by an appropriate parameterization of $\mathbb{R}^2$.

DCT interpolation

When manipulating images through the DFT interpolation, the digital image is implicitly extended to $\mathbb{Z}^2$ via periodization. This eventually leads to strong discontinuities at image borders. The discrete cosine transform (DCT) interpolation reduces the discontinuities caused by the brutal periodization by first symmetrizing the image.

The DCT interpolation of the digital image $(u_{k,l})$ of size $M \times N$ is equivalent to the DFT interpolation of the symmetrized signal $(\mathring{u}_{k,l})$ of size $2M \times 2N$ where $\mathring{u}_{k,l} = u_{s_M(k),s_N(l)}$ with $s_M(k) = \min(k, 2M - 1 - k)$ for $0 \leq k \leq 2M - 1$ and $s_N(l)$ is defined similarly. The DFT interpolation (Proposition 1) applied to this particular case defines the DCT interpolation as the only trigonometric polynomial $\mathring{u}$ of degrees M and N and of periodicities $2M$ and $2N$ that interpolates exactly the symmetrized image

$$\mathring{u}\,(k,l) = \mathring{u}_{k,l}, \quad \text{for } k = 0, \ldots, 2M-1 \text{ and } l = 0, \ldots, 2N-1,$$

namely,

$$\mathring{u}(x,y) = \sum_{m=-M}^{M-1} \sum_{n=-N}^{N-1} \widetilde{\mathring{u}}_{m,n} e^{\frac{i\pi mx}{M}} e^{\frac{i\pi ny}{N}}.$$

Thus, the DCT interpolation can be expressed as

$$\mathring{u}(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \alpha_m \alpha_n DCT(u)_{m,n} \cos\left(\frac{\pi(x + 1/2)m}{M}\right) \cos\left(\frac{\pi(y + 1/2)n}{N}\right),$$

with $\alpha_m = 1/2$ when $m = 0$ and $\alpha_m = 1$ elsewhere, and where $(DCT(u)_{m,n})$ denotes the type-II DCT coefficients of image $(u_{k,l})$ defined for $m = 0, \ldots, M-1$ and $n = 0, \ldots, N-1$ by

$$DCT(u)_{m,n} = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} u_{k,l} \cos\left(\frac{\pi(k + 1/2)m}{M}\right) \cos\left(\frac{\pi(l + 1/2)n}{N}\right).$$

Since $\mathring{u}(k,l) = u_{k,l}$ for $k = 0, \ldots, M-1$ and $l = 0, \ldots, N-1$, the inverse DCT transform IDCT, is computed by

$$IDCT(u)_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \alpha_m \alpha_n u_{m,n} \cos\left(\frac{\pi(k + 1/2)m}{M}\right) \cos\left(\frac{\pi(l + 1/2)n}{N}\right),$$

with $\alpha_m = 1/2$ when $m = 0$ and $\alpha_m = 1$ elsewhere.

2.3 The Convolution Theorem

Let $u_{k,l} \in \mathbb{R}$ for $k = 0, \dots, M-1$ and $l = 0, \dots, N-1$ be a real-valued digital image of size $M \times N$, and let $u(x, y)$ be its DFT interpolation,

$$u(x,y) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\left(-\lfloor \frac{M}{2} \rfloor + M - 1\right)} \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\left(-\lfloor \frac{N}{2} \rfloor + N - 1\right)} \tilde{u}_{m,n} e^{\frac{2i\pi mx}{M}} e^{\frac{2i\pi ny}{N}}$$

where $(\tilde{u}_{m,n}) = DFT\left((u_{k,l})\right)$. The following theorem states that the convolution of the DFT interpolation of a digital image with a linear filter can be computed exactly by properly weighting its DFT coefficients. This result plays an important role in the present framework. Indeed, it links the continuous image model to the discrete computations that we are able to compute in practice.

Theorem 1. *The convolution of the trigonometric polynomial*

$$u(x,y) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\left(-\lfloor \frac{M}{2} \rfloor + M - 1\right)} \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\left(-\lfloor \frac{N}{2} \rfloor + N - 1\right)} \tilde{u}_{m,n} e^{\frac{2i\pi mx}{M}} e^{\frac{2i\pi ny}{N}}$$

with a function $f \in L^1(\mathbb{R}^2)$, is the trigonometric polynomial

$$f * u(x,y) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\left(-\lfloor \frac{M}{2} \rfloor + M - 1\right)} \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\left(-\lfloor \frac{N}{2} \rfloor + N - 1\right)} \tilde{u}_{m,n} \widehat{f}\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) e^{\frac{2i\pi mx}{M}} e^{\frac{2i\pi ny}{N}}.$$

Proof. Let us consider the pure wave of frequency $\xi \in \mathbb{R}^2$, $g_\xi(\mathbf{x}) = e^{i\xi \cdot \mathbf{x}}$. Then,

$$f * g_\xi(\mathbf{x}) = \int_{\mathbb{R}^2} f(\mathbf{x}') g_\xi(\mathbf{x} - \mathbf{x}') d\mathbf{x}' = e^{i\xi \cdot \mathbf{x}} \int_{\mathbb{R}^2} f(\mathbf{x}') e^{-i\xi \cdot \mathbf{x}'} d\mathbf{x}' = \widehat{f}(\xi) g_\xi(\mathbf{x}),$$

which is a pure wave of the same frequency. The result follows from the linearity of convolution. □

This theorem can be extended to the DCT interpolation by considering the symmetrized $2M \times 2N$ image.

3 Analysis of Three Digital Gaussian Convolution Algorithms

Several algorithms have been proposed for the Gaussian smoothing of digital images. This work concentrates on three of them, the Fourier based convolution, the discrete convolution with samples from the Gaussian function and Lindeberg's discrete scale-space smoothing. In what follows we describe each of these algorithms.

3.1 DFT Convolution

Since only a finite set of the image values are known, it is not possible to directly compute the continuous Gaussian convolution. However, if for example, we accept that the image is band-limited and periodic, then we can fully recover the image values in the continuous domain. This is done by the DFT interpolation presented in Proposition 1. Moreover, if we accept these convenient hypotheses,

the continuous Gaussian convolution can be computed exactly in the Fourier domain at the cost of two DFTs and one operation per pixel, as indicated by the convolution theorem (Theorem 1).

Remark. The Fourier transform of the isotropic Gaussian function $G_\sigma(x, y)$ with standard deviation σ is $\widehat{G}_\sigma(\xi, \eta) = e^{-\frac{\sigma^2}{2}(\xi^2+\eta^2)}$. Applying Theorem 1 to the Gaussian function, the continuous Gaussian convolution of the DFT interpolation $u(x, y)$ of digital image $(u_{k,l})$ is

$$G_\sigma * u(x, y) = \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\left(-\lfloor \frac{M}{2} \rfloor + M - 1\right)} \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\left(-\lfloor \frac{N}{2} \rfloor + N - 1\right)} \tilde{u}_{m,n} \widehat{G}_\sigma \left(\frac{2\pi m}{M}, \frac{2\pi n}{N} \right) e^{\frac{2i\pi mx}{M}} e^{\frac{2i\pi ny}{N}},$$

where $\widehat{G}_\sigma(\frac{2\pi m}{M}, \frac{2\pi n}{N}) = e^{-\frac{\sigma^2\pi^2}{2}\left(\left(\frac{2m}{M}\right)^2+\left(\frac{2n}{N}\right)^2\right)}$.

A description of the method is presented in Algorithm 1 while Figures 1 and 2 provide illustrations of the algorithm behavior in the image and Fourier domains.

Since this algorithm implements the continuous Gaussian convolution (assumed an underlying continuous image model), all properties of the continuous Gaussian convolution are verified. However, using the DFT interpolation amounts to implicitly assuming that the digital image originates from sampling a band-limited periodic function below the Nyquist rate. The assumption that the image is well sampled is often unrealistic. Although for natural images, the low-pass filter behavior of digital cameras justifies the assumption of an underlying band-limited function, the frequency band will not be necessarily the same as the one covered by the sampling. The periodic assumption is unnatural. The forced periodization can lead to strong discontinuities at image borders (see Figure 1), which contradicts in some extent the band limited-assumption, causing ringing. These artifacts can be reduced by using the DCT variant.

DCT Convolution

Since the DCT interpolation of an $M \times N$ image is equivalent to the DFT interpolation of the $2M \times 2N$ mirror symmetrized image, Theorem 1 can be reformulated in terms of DCT sequences. The underlying continuous image model is then a trigonometric polynomial of degrees M and N and periodicities $2M$ and $2N$. The continuous convolution of the DCT interpolation $u(x, y)$ of the digital image $(u_{k,l})$ with a Gaussian function of standard deviation σ is

$$G_\sigma * u(x, y) = \sum_{m=-M}^{M-1} \sum_{n=-N}^{N-1} DCT(u)_{m,n} \widehat{G}_\sigma \left(\frac{\pi m}{M}, \frac{\pi n}{N} \right) e^{\frac{i\pi mx}{M}} e^{\frac{i\pi ny}{N}},$$

where $\widehat{G}_\sigma(\frac{\pi m}{M}, \frac{\pi n}{N}) = e^{-\frac{\sigma^2\pi^2}{2}\left(\left(\frac{m}{M}\right)^2+\left(\frac{n}{N}\right)^2\right)}$. The complexity of the process is reduced by using the symmetry of the DCT coefficients. The continuous Gaussian convolution involves weighting the $M \times N$ type-II DCT coefficients. The algorithm is detailed in Algorithm 2. Figure 3 shows a Gaussian DCT smoothing of a grayscale image. This figure illustrates how the artifacts induced by the implicit DFT periodization are removed with the DCT variant.

3.2 Convolution with a Sampled Gaussian Function.

The most common discrete approximation of the Gaussian convolution is obtained by sampling the truncated continuous Gaussian function. Although being straightforward to implement, we will show that it does not satisfy the semi-group property for small σ values.

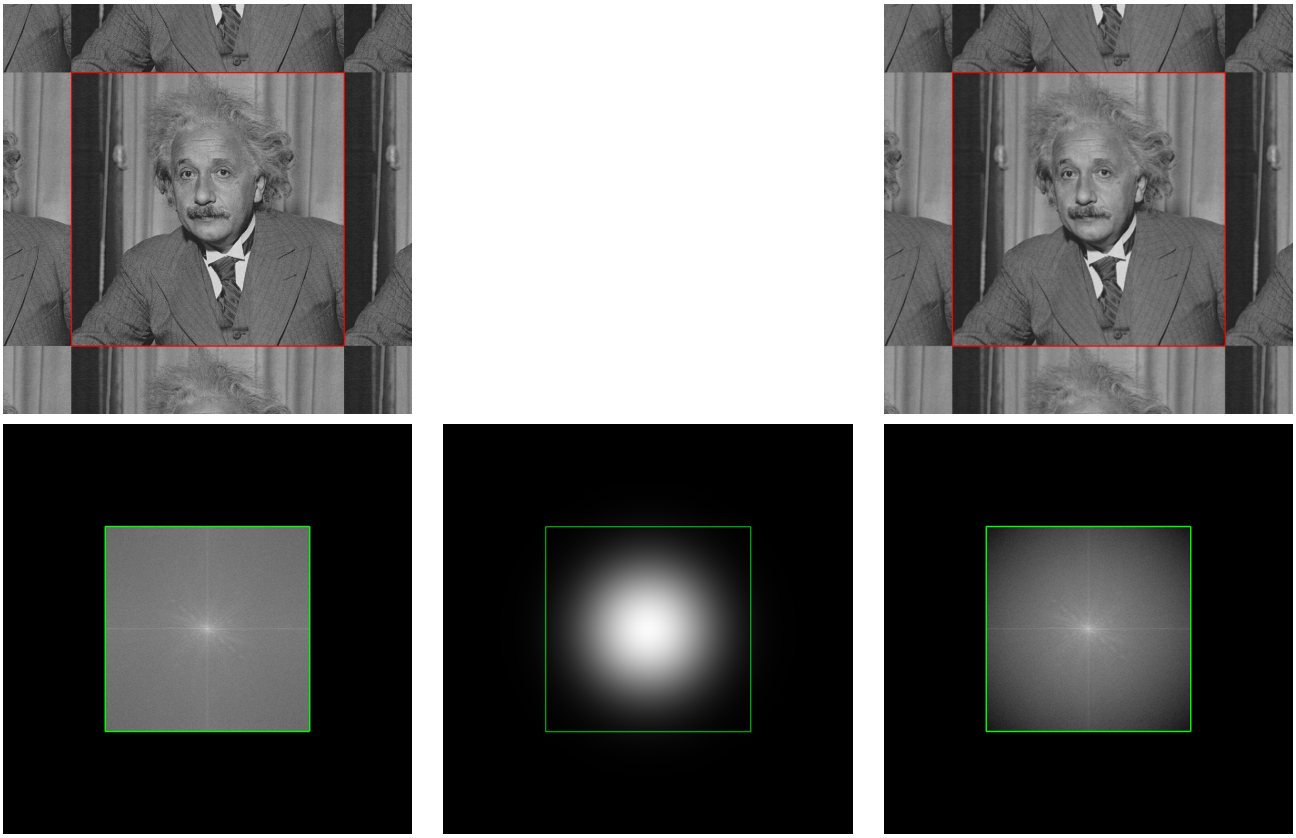

Figure 1: Illustrating the continuous Gaussian convolution through DFT interpolation. The first column illustrates the adopted continuous image model. The image is defined on the $\mathbb{R}^2$ plane. It is periodic $u(x + kM, y + lN) = u(x, y)$ (top) and band-limited with $\mathrm{supp}(\widehat{u}) \in [-\pi, \pi]^2$ (bottom). The borders of the digital image are represented by a red box. The green box indicates $[-\pi, \pi]^2$, the domain relative to the sampling. The domain is extended to illustrate that the continuous image is periodic and band-limited. The second column illustrates the Fourier transform of the Gaussian function defined on the $\mathbb{R}^2$ plane for $\sigma = 0.8$. The convolution in the spatial domain is equivalent to a multiplication in the Fourier domain. The last column illustrates the periodic trigonometric polynomial relative to this Gaussian convolution (top) and a representation of the Dirac amplitudes in its Fourier transform (bottom). It is a classic convention, also adopted here, that a continuous digital image is displayed by showing a constant value on each pixel, equal to its sampled value at the pixel center. Thanks to the optical blur of the screen and of our vision, the result is, visually, a decent representation of the smooth ideal image.

The continuous 1D Gaussian function of standard deviation σ is truncated at width $2K\sigma$; typical values of K are 3 or 4 to gather most of the signal's energy. Then, it is sampled to produce the discrete filter $(g_k)_k$ of width $2\lceil K\sigma \rceil + 1$ and normalized to sum one,

$$g_k = \frac{e^{-\frac{k^2}{2\sigma^2}}}{\sum_{l=-\lceil K\sigma \rceil}^{\lceil K\sigma \rceil} e^{-\frac{l^2}{2\sigma^2}}}, \quad k = -\lceil K\sigma \rceil, \dots, \lceil K\sigma \rceil.$$

The convolution with the sampled Gaussian function algorithm (detailed in Algorithm 3) consists in the computation of the separable 2D discrete convolution

$$v_{k,l} = \sum_{k'=-\lceil K\sigma \rceil}^{\lceil K\sigma \rceil} g_{k'} \sum_{l'=-\lceil K\sigma \rceil}^{\lceil K\sigma \rceil} g_{l'} \; u'_{k-k',l-l'},$$

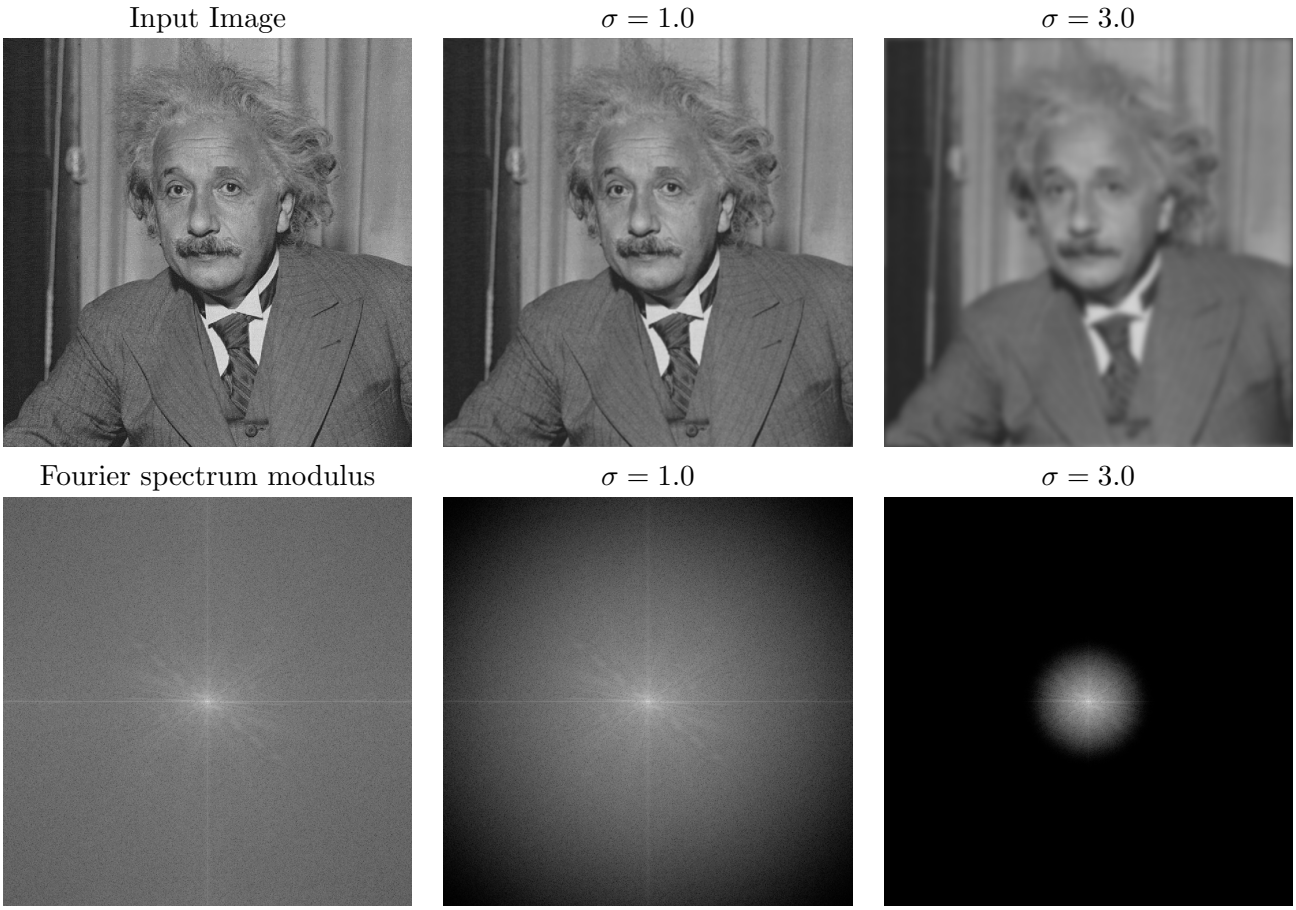

Figure 2: Grayscale input image and the results of applying the DFT Gaussian convolution with parameter $\sigma = 1.0, 3.0$. In the bottom row, the respective moduli of the Fourier spectra show the attenuation of high frequencies. The high values in the Fourier spectra along the vertical and horizontal axes are caused by the strong discontinuities when periodizing the image.

where u' denotes the extension of u to the $\mathbb{Z}^2$ plane either by (M, N)-periodization or symmetrization followed by $(2M, 2N)$-periodization. Formally

$$u'(k, l) = u(s_M(k), s_N(l)) \quad \text{with } s_M(k) = k \bmod M,$$

for periodic extension, or

$$s_M(k) = \min(k \bmod 2M, 2M - 1 - (k \bmod 2M)),$$

in case of the (half-sample) symmetric extension. The pseudo-code in Algorithm 3 incorporates the symmetric extension of the signal, the modification for the periodic extension is straightforward.

Gaussian function aliasing and semi-group property. The Fourier transform of the Gaussian function G_σ, $\widehat{G}_\sigma(\xi, \mu) = e^{-\sigma^2 \frac{\xi^2+\mu^2}{2}}$, has no compact support. Thus, the sampling of the Gaussian function never satisfies the band-limited assumption needed by the Nyquist-Shannon sampling theorem (see e.g. [4]). Since the value of $\widehat{G}_\sigma(\xi, \mu)$ at the Nyquist frequency is $e^{-\pi^2\sigma^2/2}$, the aliasing is not significant for $\sigma > 1$ [11].

As we will show in the numerical experiments in Section 4, the aliasing of the Gaussian function contributes to a significant deviation from the semi-group property.

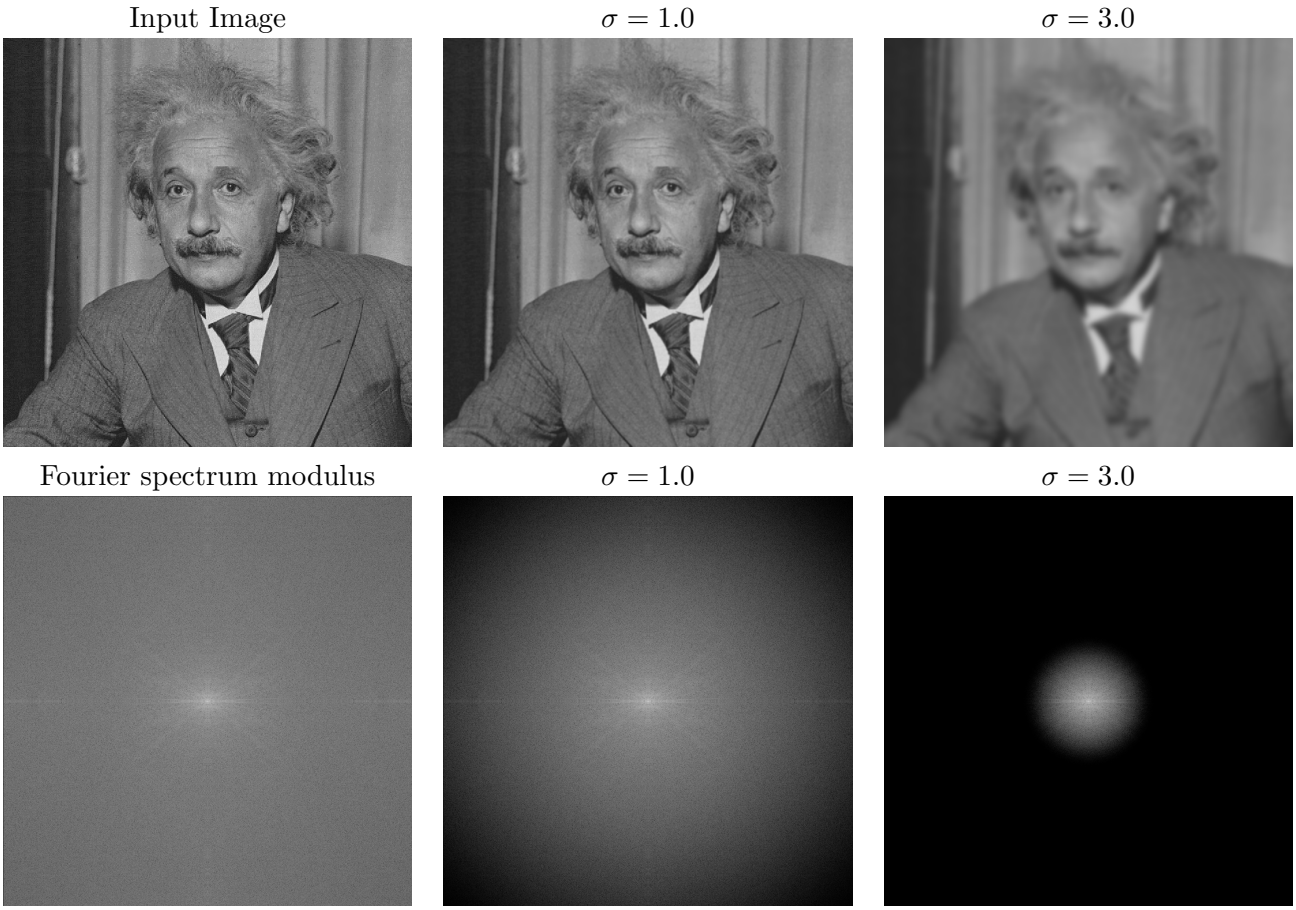

Figure 3: Grayscale input image and the results of applying the DCT Gaussian convolution with parameter $\sigma = 1.0, 3.0$. In the bottom row, the respective moduli of the Fourier spectra show the attenuation of high frequencies similar to the case of DFT convolution. The main difference is that in the DCT Gaussian smoothing, the implicit symmetrization of the image avoids the strong discontinuities when periodizing.

Truncation error. The error due to truncation is shown in Figure 4. The error is very small for large enough values of K (for instance, the error is less than 10^{-4} for $K \geq 4$). The truncation at $\lceil K\sigma \rceil$ also induces oscillations on the spectrum. If $(v_{k,l})$ and $(v^{\text{trunc}}_{k,l})$ denote the respective outputs of the convolutions of u with the infinite and the truncated versions of the sampled Gaussian function, then their DFT coefficients are related by

$$\hat{v}^{\text{trunc}}_{m,n} = \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} \hat{v}_{m',n'} D_{\lceil K\sigma \rceil}\left(2\pi(m-m')/M\right) D_{\lceil K\sigma \rceil}\left(2\pi(n-n')/N\right),$$

for $m = -\lfloor M/2 \rfloor, \ldots, -\lfloor M/2 \rfloor + M - 1$ and $n = -\lfloor N/2 \rfloor, \ldots, -\lfloor N/2 \rfloor + N - 1$ and where D_L denotes the Dirichlet function (also known as the periodic sinc function),

$$D_L(x) = \begin{cases} \frac{\sin((L+1/2)x)}{\sin(x/2)} & \text{if } x \neq 2k\pi, k \in \mathbb{Z} \\ \\ (-1)^{k(L-1)} & \text{if } x = 2k\pi, k \in \mathbb{Z}. \end{cases}$$

The oscillating spectrum due to the convolution with the Dirichlet kernel is noticeable for small values of K, as can be seen in Figure 5.

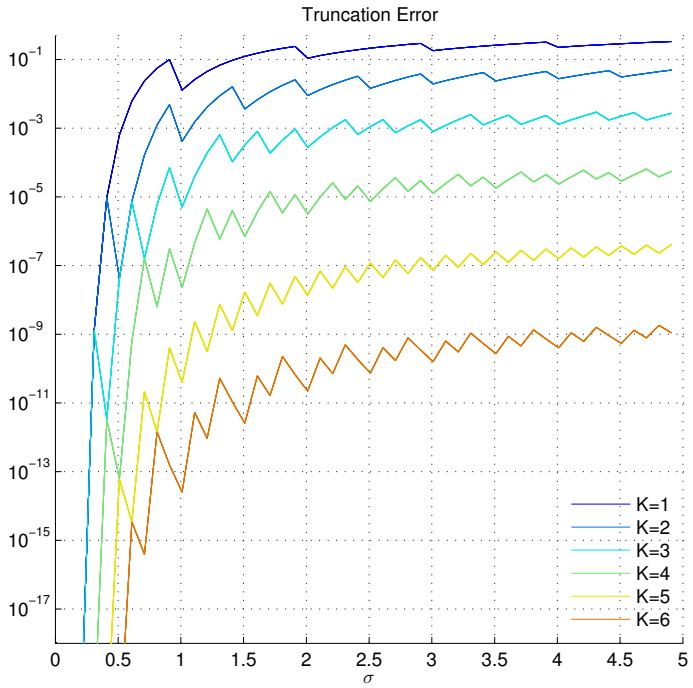

Figure 4: Impact of truncation of the Gaussian function. The sequence of samples of the 1D Gaussian function g_k is truncated at $\lceil K\sigma \rceil$ samples. The truncation error is defined as the square root of the total energy loss, $\sqrt{\sum_{|k|>\lceil K\sigma \rceil} g_k^2}$, divided by the square root of the total energy of the Gaussian function, $\sqrt{\sum_k g_k^2}$. We have approximated the total energy of the Gaussian function by $\sum_{|k|\leq\lceil 50\sigma \rceil} g_k^2$. The truncation error decays rapidly with K and becomes smaller than 10^{-4} for $K \geq 4$.

3.3 Lindeberg's Discrete Scale-Space Smoothing

Let u be a continuous and bounded signal, then the Gaussian convolution $v : t \mapsto G_{\sqrt{2t}}u$ is the solution of the heat equation $\frac{\partial v}{\partial t} = \Delta v$ with initial condition $v(0, \mathbf{x}) = u(\mathbf{x})$ [2] . Lindeberg's smoothing method [10] is based on computing the solution of a spatial discretization of the heat equation $\frac{\partial v}{\partial t} = \Delta v$.

For a one-dimensional sequence $(u_k)_{k\in\mathbb{Z}}$, Lindeberg's smoothing method consists in finding $v_k(t)$, solution of

$$\partial_t v_k(t) = \Delta^{\text{discr}} v_k(t), \quad \text{with} \quad v_k(0) = u_k,$$

where $\Delta^{\text{discr}} v_k(t)$ denotes the 1D Laplacian finite difference scheme $\Delta^{\text{discr}} v_k = v_{k-1} - 2v_k + v_{k+1}$. This solution can be computed via a discrete convolution with the discrete sequence

$$g_n^{\text{Lindeberg}} = e^{-t} I_n(t),$$

where $t = \sigma^2/2$ and I_n denotes the modified Bessel functions.

For a two-dimensional signal $(u_{k,l})_{(k,l)\in\mathbb{Z}^2}$, Lindeberg's smoothing method consists in solving

$$\partial_t v_{k,l}(t) = \Delta_\gamma^{\text{discr}} v_{k,l}(t), \quad \text{with} \quad v_{k,l}(0) = u_{k,l},$$

where $\Delta_\gamma^{\text{discr}}$ denotes the following 2D Laplacian finite difference scheme

$$\Delta_\gamma^{\text{discr}} u = (1-\gamma)\Delta_+ u + \gamma \Delta_\times u,$$

[2] Equivalently, thanks to the re-parameterization $\sigma = \frac{t^2}{2}$, $v : \sigma \mapsto G_\sigma u$ is the solution of the equation $\frac{\partial v}{\partial \sigma} = \sigma \Delta v$.

Sampled Gaussian function truncated at $\lceil 3\sigma \rceil$ Sampled Gaussian function truncated at $\lceil 4\sigma \rceil$

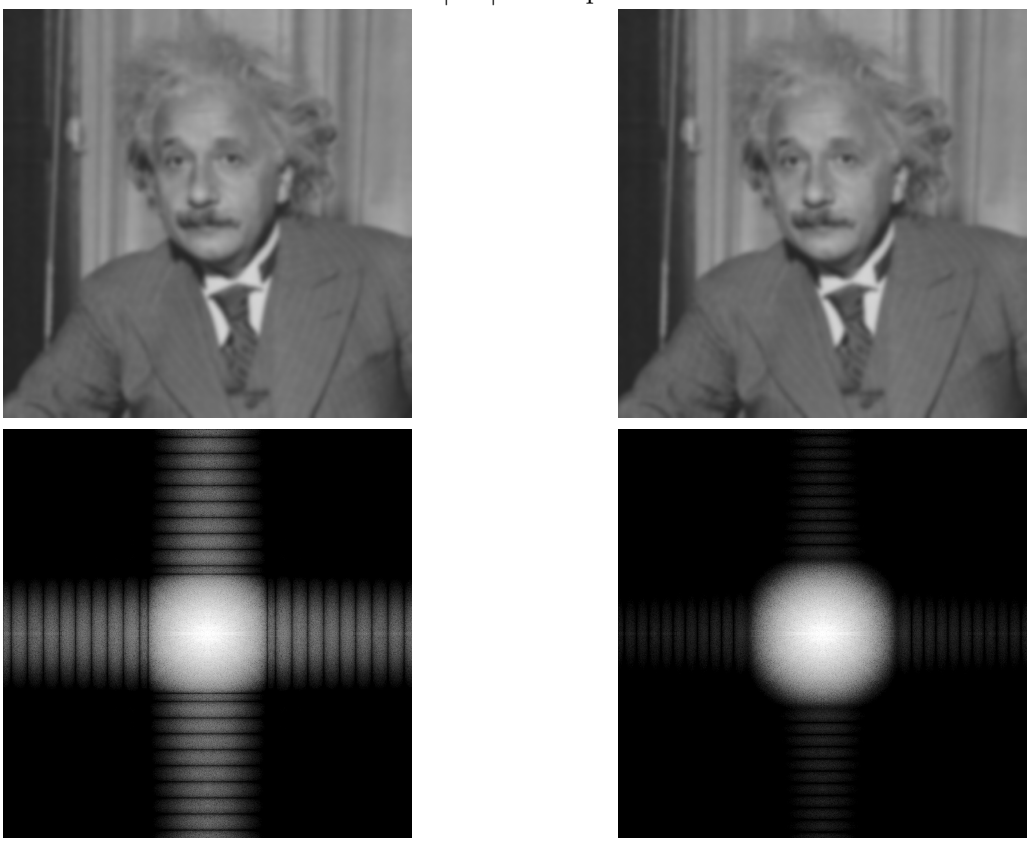

Figure 5: Impact of truncation. In the first row, the convolutions of a test image with sampled Gaussian function of standard deviation $\sigma = 4$ truncated at $\lceil 3\sigma \rceil$ (left) and $\lceil 4\sigma \rceil$ (right). Second row, the respective image spectra (the intensity scale is optimized for better visualization). Notice the oscillations for a truncation at $\lceil 3\sigma \rceil$ of the sampled Gaussian function.

with

$$\Delta_+ u_{k,l} = u_{k+1,l} + u_{k-1,l} + u_{k,l+1} + u_{k,l-1} - 4u_{k,l},$$
$$\Delta_\times u_{k,l} = 1/2(u_{k+1,l+1} + u_{k+1,l-1} + u_{k-1,l+1} + u_{k-1,l-1}) - 2u_{k,l},$$

for $0 \leq \gamma \leq 1/2$. The parameter γ controls the shape of the Laplacian discrete operator[3].

The smoothed image is computed by Euler's method. This explicit time marching scheme consists in applying the following iteration formula

$$\frac{v(p\delta t)_{k,l} - v((p-1)\delta t)_{k,l}}{\delta t} = \Delta_\gamma^{\text{discr}} v((p-1)\delta t)_{k,l}$$

for $1 \leq p \leq P$ with δt the step size and P the total number of iterations (i.e. $P\delta t = \sigma^2/2$). The stability of Euler's method is guaranteed if the step size satisfies $\delta t < 1/8(1 - \gamma/2\sigma)$. The implementation of Lindeberg's smoothing method is detailed in Algorithm 4.

[3]For a thorough analysis of the influence of parameter γ on isotropy, we refer the interested reader to [10] pp. 127-134.

4 Experiments

Let us assume that the Gaussian semi-group property is valid. Then, applying N times a Gaussian filter of parameter σ should produce the same result as filtering only once with a Gaussian function of parameters $\sqrt{N}\sigma$. This allows us to evaluate the validity of the semi-group property for all the described methods.

Indeed, if an image of a Gaussian function is filtered by a Gaussian function of a given standard deviation, the filtered signal should be a Gaussian function of a standard deviation given by the semi-group property. Thus, the following experiment was carried out. A sampled Gaussian function of standard deviation σ_{in} was considered as the input signal. It was filtered N times by each of the different Gaussian filters implementations with parameter σ. A Gaussian function was fitted to the filtered image by least squares. The estimated standard deviation was compared to the theoretical expected value $\sqrt{\sigma_{\text{in}}^2 + N\sigma^2}$. The input Gaussian standard deviation was set to $\sigma_{\text{in}} = 1.0$ to avoid aliasing artifacts, and the number of iterations N was set to 10.

The results are shown in Figures 6 to 9. Each figure shows the estimated blurs, the differences between estimated and theoretical values, and the root-mean-square error between the pixels of the two filtered images.

The experiment demonstrates that, as expected, the DFT convolution (Figure 6) and its DCT variant (Figure 7) fully satisfy the semi-group property with machine precision error.[4] Figure 8 shows the previous experiment for the sampled Gaussian function truncated at $\lceil 5\sigma \rceil$. For low values of σ, the estimated blur level deviates from the theoretical value $\sqrt{N}\sigma$ and the method fails to satisfy the semi-group property. This is due to the aliasing in the sampled Gaussian function. The difference with respect to the theoretical values is less than 10^{-3} for $\sigma \geq 0.8$. Applying Lindeberg's method consists in solving a discretized version of the heat equation. The parameter γ which defines the Laplacian finite difference scheme is set here to $\gamma = 1/2$. Lindeberg's smoothing method satisfies the semi-group property (Figure 9) but the estimated blur is lower than the theoretical value.

Additionally, direct and iterated convolutions were applied on a test image. For all four methods, the RMSE[5] between the direct and the iterated convolutions is displayed in Figure 10, while Figure 11 shows the image difference. The DCT and DFT convolution produce the lowest errors. Nevertheless, the sampled Gaussian function and Lindeberg's method give similar errors for large values of σ (i.e. $\sigma \geq 0.9$).

5 Conclusion

In this work we analyzed three methods for the Gaussian smoothing of digital images. We focused on the most commonly used methods for the computation of the Gaussian scale-space. We have detailed their implementation as well as an analysis of how they differ from the continuous Gaussian convolution.

Computing the Gaussian scale-space with high precision requires an accurate implementation of the Gaussian convolution for low blur levels. With that aim, the present work was particularly focused on the accuracy at low levels of Gaussian blur (i.e. $\sigma \leq 1$).

The DFT and DCT Gaussian convolutions fully satisfy the semi-group property, thus giving an accurate discrete implementation of the continuous Gaussian convolution. The discrete convolution with samples from a Gaussian function also satisfies the semi-group property for large blurs (i.e. $\sigma > 0.8$). However, the aliasing of the sampled function for low blur levels makes it unsuitable for

[4] The algorithms are implemented using single-precision float data type.

[5] Root-mean-square error. For (x_n) and (y_n) with $n = 1 \dots N$, $\texttt{RMSE}((x_n), (y_n)) = \frac{1}{N}\sqrt{\sum_{n=1}^{N}(x_n - y_n)^2}$.

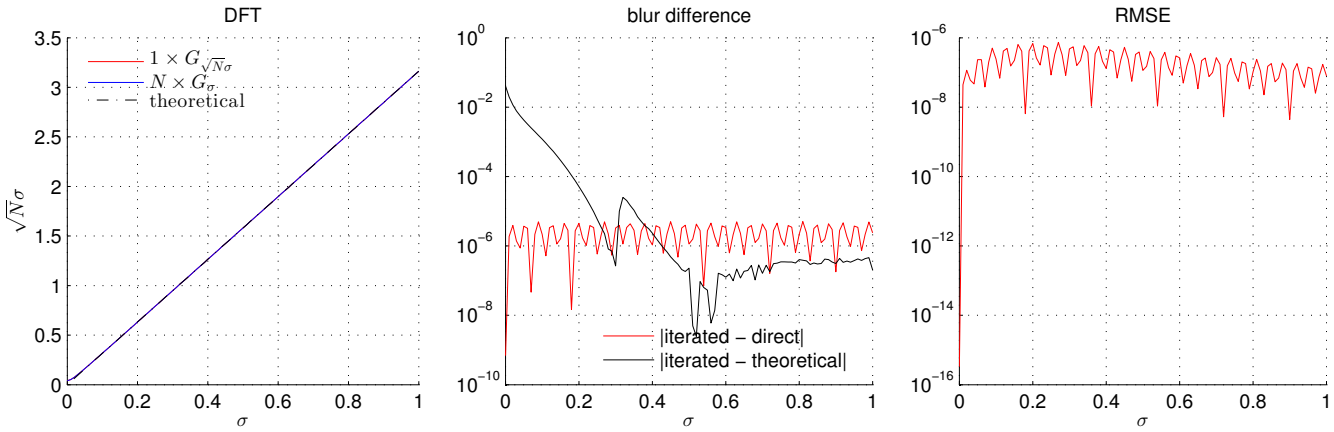

Figure 6: DFT convolution. A Gaussian convolution of parameter $\sqrt{N}\sigma$ is compared to $N = 10$ iterations of a Gaussian convolution of parameter σ (denoted $N \times G_\sigma$) for different values of σ. On the left, the estimated blur levels for the direct and iterated filters are plotted as a function of σ. The theoretical value $\sqrt{N}\sigma$ is plotted in black. In the center, the difference between the estimated blur levels for direct and iterated filters as a function of σ is plotted in red. This difference is below 10^{-5} which indicates that the DFT method satisfies the semi-group property. The difference between the estimated blur level in the iterated filtered image and the theoretical blur level as a function of σ is plotted in black. The DFT convolution is accurate since this difference is below 10^{-3} for $\sigma \geq 0.1$ and is below 10^{-6} for $\sigma \geq 0.4$. On the right, the root-mean-square error (RMSE) between the pixel values of both filtered images confirms the DFT consistency regarding the semi-group property.

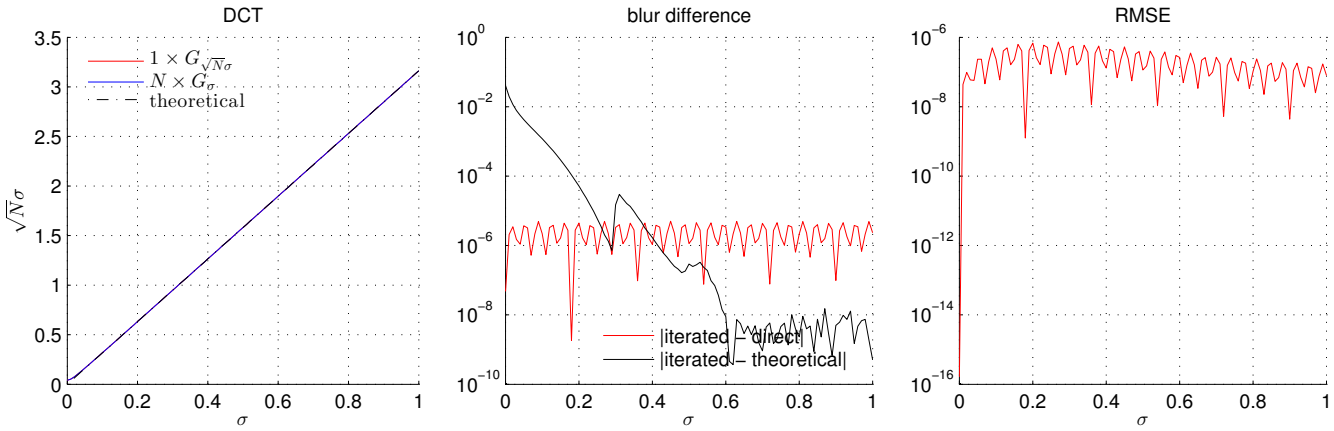

Figure 7: The DCT convolution of an image is the DFT convolution after symmetrization of the image. Unsurprisingly, the semi-group property is satisfied by this variant.

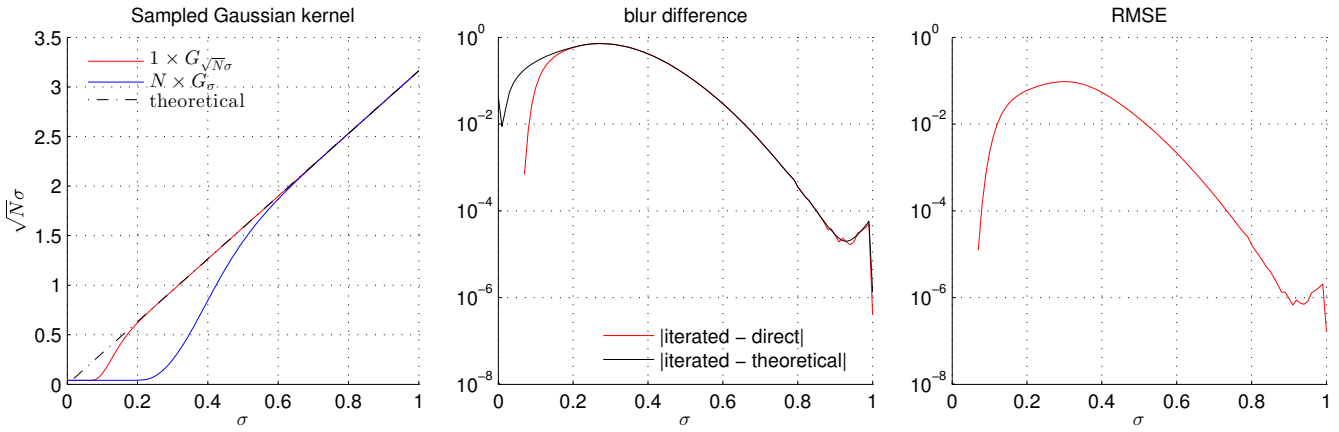

Figure 8: Sampled Gaussian functions truncated at $\lceil 5\sigma \rceil$. A convolution of parameter $\sqrt{N}\sigma$ is compared to $N = 10$ iterations of a filter of parameter σ (denoted $N \times G_\sigma$) for the range $0 \leq \sigma \leq 1$. On the left, the estimated blur levels for the direct and iterated filters are plotted for different values of σ. The theoretical value $\sqrt{N}\sigma$ is plotted in black. The center and right plots show the blur level difference and the RMSE of the filtered images respectively. For low values of σ, the estimated blur after N convolutions is lower that the theoretical value. Indeed, in this case, the sampled functions are aliased and the method does not satisfy the semi-group property. This is confirmed by a blur difference above 10^{-2} for $\sigma \leq 0.6$ (red curve, center plot). The difference with respect to the theoretical values is less than 10^{-3} for $\sigma \geq 0.8$. For very low values of σ (e.g. $\sigma \approx 0.2$), the measured blur is null. This is reasonable since in this case, the sampled function is reduced to a sequence with only one nonzero coefficient.

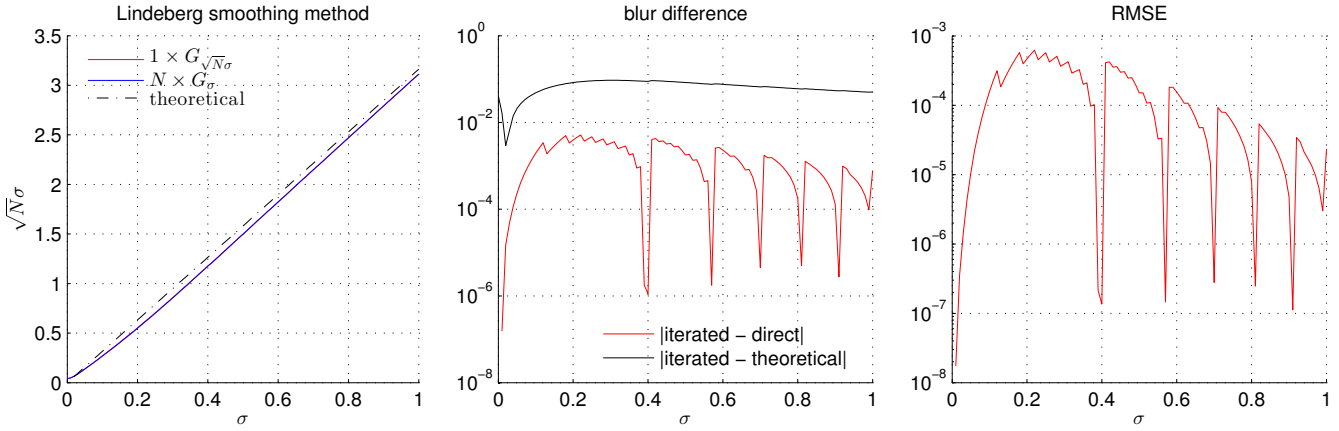

Figure 9: Lindeberg's smoothing method. A smoothing of parameter $\sqrt{N}\sigma$ is compared to $N = 10$ iterations of the method with parameter σ (denoted $N \times G_\sigma$) for the range $0 \leq \sigma \leq 1$. The method consists in the resolution of a discretized version of the heat equation. The experiment demonstrates that Lindeberg's smoothing method satisfies the semi-group property. The two measured blurs are almost identical (difference around 10^{-3}, see center plot). However, the estimated blur is lower than the theoretical value of $\sqrt{N}\sigma$.

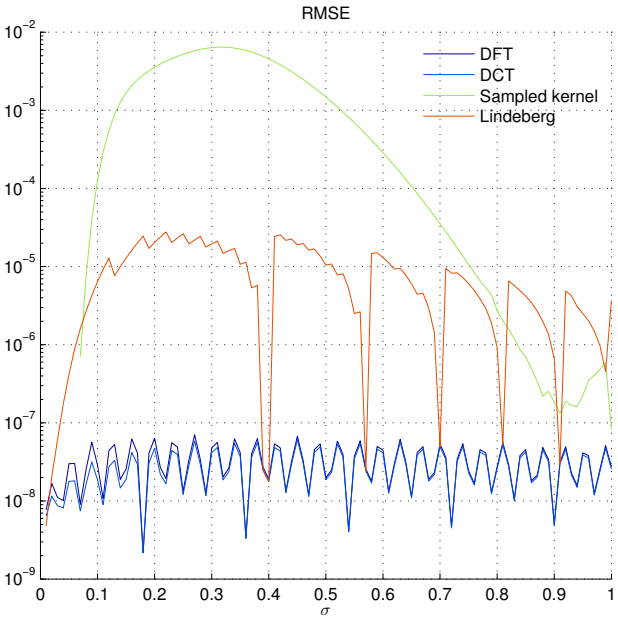

Figure 10: Validity of the semi-group property on a natural image (`portrait`, see Figure 11) for DFT and DCT convolutions, convolution with sampled Gaussian functions and Lindeberg's smoothing method. The RMSE between a convolution of parameter $\sqrt{N}\sigma$ and $N = 10$ iterations of a Gaussian filtering of parameter σ is plotted as a function of $0 \leq \sigma \leq 1$. DCT and DFT produce the lowest errors, followed by Lindeberg's method. For $\sigma \geq 0.9$ the RMSE produced with the sampled Gaussian function is similar to those produced by Lindeberg's method.

accurate computations of the Gaussian scale-pace. Finally, although Lindeberg's smoothing method satisfies the semi-group property, it introduces a bias in the applied amount of blur.

A Pseudocodes

Algorithm 1: DFT convolution.

Inputs: - u, input digital image of $M \times N$ pixels.
- σ, standard deviation of the Gaussian function.

Output: v, output image of $M \times N$ pixels.

`//Compute the DFT coefficients of` u
$(\widetilde{u}_{m,n}) \leftarrow DFT(u_{k,l})$

`//Weight the DFT coefficients`
for $-\lfloor \frac{M}{2} \rfloor \leq m \leq -\lfloor \frac{M}{2} \rfloor + M - 1$ **and** $-\lfloor \frac{N}{2} \rfloor \leq n \leq -\lfloor \frac{N}{2} \rfloor + N - 1$ **do**

$$\widetilde{v}_{m,n} \leftarrow \widetilde{u}_{m,n} \widehat{G}_\sigma \left(\frac{2\pi m}{M}, \frac{2\pi n}{N} \right) = \widetilde{u}_{m,n} e^{-\frac{\sigma^2 \pi^2}{2} \left(\left(\frac{2m}{M}\right)^2 + \left(\frac{2n}{N}\right)^2 \right)}$$

`//Compute Inverse discrete Fourier transform of` $\widetilde{v}$
$(v_{k,l}) \leftarrow IDFT(\widetilde{v}_{m,n})$

return v

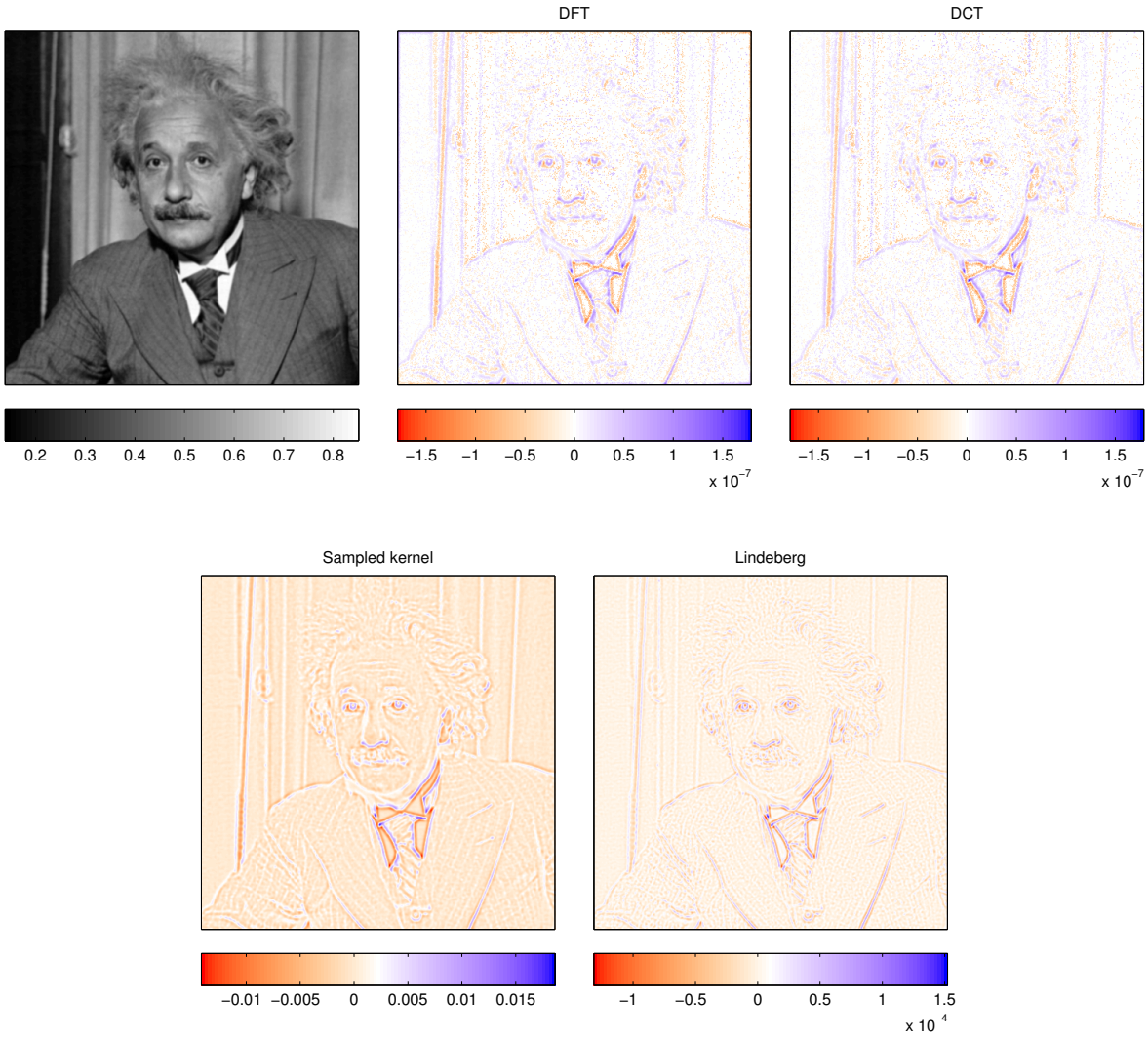

Figure 11: Image difference between direct and iterated convolutions for the four studied algorithms applied on the test image `portrait`. For the DFT and DCT convolutions, for convolution with sampled Gaussian functions and Lindeberg's method, the smoothing parameter σ is set to 0.5 for each iterated filtering and to $0.5\sqrt{10}$ for direct filtering. The methods based on Fourier and Lindeberg's method are consistent with the semi-group property. The measured RMSE between direct and iterated convolution are 7.81×10^{-3} (DFT and DCT), 6.29 (sampled Gaussian) and 5.90×10^{-2} (Lindeberg). The DCT and DFT methods achieve machine precision.

Algorithm 2: DCT convolution.

Inputs: - u, input digital image of $M \times N$ pixels.
- σ, standard deviation of the Gaussian function.

Output: v, output image of $M \times N$ pixels.

Temporary: -$DCT(u)$, type-II DCT coefficients of the input image, $M \times N$ real coefficients.
-$DCT(v)$, type-II DCT coefficients of the output image.

`//Compute the DCT coefficients of` u
$(DCT(u)_{m,n}) \leftarrow DCT(u_{k,l})$

`//Weight the DCT coefficients`
for $0 \leq m \leq M-1$ **and** $0 \leq n \leq N-1$ **do**

$$DCT(v)_{m,n} \leftarrow DCT(u)_{m,n}\widehat{G}_\sigma\left(\frac{\pi m}{M}, \frac{\pi n}{N}\right) = DCT(u)_{m,n}e^{-\frac{\sigma^2\pi^2}{2}\left(\left(\frac{m}{M}\right)^2+\left(\frac{n}{N}\right)^2\right)}$$

`//Compute Inverse discrete cosine transform of` $DCT(v)$
$(v_{k,l}) \leftarrow IDCT(DCT(v)_{m,n})$

return v

Algorithm 3: Convolution with a sampled Gaussian function.

Inputs: u, input digital image of $M \times N$ pixels.
σ, standard deviation of the Gaussian function

Output: v, output digital image of $M \times N$ pixels.

Parameter: K, the Gaussian function is truncated at $-\lceil K\sigma \rceil$ and $\lceil K\sigma \rceil$.

Temporary: w, $M \times N$ image used to store intermediate computations.

`//Sample the truncated Gaussian function.`
for $-\lceil K\sigma \rceil \leq k \leq \lceil K\sigma \rceil$ **do** $g_k = e^{-\frac{k^2}{2\sigma^2}}$

`//Normalize the sequence to sum 1.`
for $-\lceil K\sigma \rceil \leq k \leq \lceil K\sigma \rceil$ **do** $g_k = g_k/(\sum_{k'} g_{k'})$

`//Convolution on columns`
for $0 \leq m \leq M-1$ **and** $0 \leq n \leq N-1$ **do**

$w_{m,n} \leftarrow \sum_{k=-\lceil K\sigma \rceil}^{\lceil K\sigma \rceil} g_k u_{s_M(m-k),n}$
with $s_M(m) = \min(m \bmod 2M, 2M-1-m \bmod 2M))$

`//Convolution on lines`
for $0 \leq m \leq M-1$ **and** $0 \leq n \leq N-1$ **do**

$v_{m,n} \leftarrow \sum_{k=-\lceil K\sigma \rceil}^{\lceil K\sigma \rceil} g_k w_{m,s_N(n-k)}$
with $s_N(n) = \min(n \bmod 2N, 2N-1-n \bmod 2N))$

return v

Algorithm 4: Lindeberg's smoothing method.

Input: u input digital image of $M \times N$ pixels.
Output: v output digital image of $M \times N$ pixels.
Parameters: σ applied blur.
$0 \leq \gamma \leq 1/2$ parameter defining $\Delta_\gamma^{\text{discr}}$ the Laplacian finite difference scheme.
Temporary: P, number of Euler iterations.
δt, Euler step size.
$\Delta_+ v$, $\Delta_\times v$, $\Delta_\gamma^{\text{discr}} v$, auxiliary discrete Laplacians.

```
// Euler method setting
```
$P \leftarrow \lceil 8(1-\gamma/2)\sigma^2 \rceil$
$\delta t \leftarrow \frac{\sigma}{P}$
```
// Initialization
```
$v \leftarrow u$
```
// Euler Method
```
for $p = 1, \ldots, P$ **do**
    // Compute discrete Laplacian
    for $0 \leq k \leq M-1$ **and** $0 \leq l \leq N-1$ **do**
        $\Delta_+ v_{k,l} \leftarrow u_{k+1,l} + u_{k-1,l} + u_{k,l+1} + u_{k,l-1} - 4u_{k,l}$
        $\Delta_\times u_{k,l} \leftarrow \frac{1}{2}(u_{k+1,l+1} + u_{k+1,l-1} + u_{k-1,l+1} + u_{k-1,l-1}) - 2u_{k,l}$
        $\Delta_\gamma^{\text{discr}} v_{k,l} \leftarrow (1-\gamma)\Delta_+ v_{k,l} + \gamma \Delta_\times v_{k,l}$
        note: The half-sample symmetric boundary condition is used.
    // Euler iteration formula
    for $0 \leq k \leq M-1$ **and** $0 \leq l \leq N-1$ **do**
        $v_{k,l} \leftarrow v_{k,l} - \delta t \Delta_\gamma^{\text{discr}} v_{k,l}$
```

**note:** $\lceil \cdot \rceil$ denotes the ceiling function.

# Acknowledgements

This work was partially supported by the Centre National d'Etudes Spatiales (CNES, MISS Project), the European Research Council (Advanced Grant Twelve Labours), the Office of Naval Research (Grant N00014-97-1-0839), Direction Générale de l'Armement (DGA), Fondation Mathématique Jacques Hadamard and Agence Nationale de la Recherche (Stereo project).

# Image Credits

 Standard test image.

# References

[1] L. Alvarez, F. Guichard, P-L. Lions, and J-M. Morel, *Axioms and fundamental equations of image processing*, Archive for Rational Mechanics and Analysis, 123 (1993), pp. 199–257. http://dx.doi.org/10.1007/BF00375127.

[2] J. Babaud, A.P. Witkin, M. Baudin, and R. O. Duda, *Uniqueness of the Gaussian kernel for scale-space filtering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 (1986), pp. 26–33. http://dx.doi.org/10.1109/TPAMI.1986.4767749.

[3] R. Deriche, *Recursively implementating the Gaussian and its derivatives*, research report, INRIA, 1993. http://hal.inria.fr/inria-00074778/en/.
```

[4] C. Gasquet and P. Witomski, *Fourier Analysis and Applications: Filtering, Numerical Computation, Wavelets*, Texts in Applied Mathematics, Springer, 1999. ISBN 9780387984858.

[5] P. Getreuer, *A survey of Gaussian convolution algorithms*, Image Processing On Line, 3 (2013), pp. 286–310. http://dx.doi.org/10.5201/ipol.2013.87.

[6] F. Guichard, J.-M. Morel, and R Ryan, *Contrast invariant image analysis and PDE's*, Springer, 1999.

[7] P. Gwosdek, S. Grewenig, A. Bruhn, and J. Weickert, *Theoretical foundations of gaussian convolution by extended box filtering*, in Scale Space and Variational Methods in Computer Vision, Springer, 2012, pp. 447–458. http://dx.doi.org/10.1007/978-3-642-24785-9˙38.

[8] T. Iijima, H. Genchi, and K. Mori, *A theory of character recognition by pattern matching method*, in Learning systems and intelligent robots, Springer, 1974, pp. 437–450. http://dx.doi.org/10.1007/978-1-4684-2106-4˙22.

[9] J. Koenderink, *The structure of images*, Biological Cybernetics, 50 (1984), pp. 363–370. http://dx.doi.org/10.1007/BF00336961.

[10] T. Lindeberg, *Scale-space theory in computer vision*, Springer, 1993. http://dx.doi.org/10.1007/978-1-4757-6465-9.

[11] J-M. Morel and G. Yu, *Is SIFT scale invariant?*, Inverse Problems and Imaging, 5 (2011), pp. 115–136. http://dx.doi.org/10.3934/ipi.2011.5.115.

[12] J. Sporring, M. Nielsen, L. Florack, and P. Johansen, *Gaussian Scale-Space Theory*, vol. 8, Kluwer Academic Publishers, Dordrecht, 1997.

[13] J. Weickert, S. Ishikawa, and A. Imiya, *Linear scale-space has first been proposed in Japan*, Journal of Mathematical Imaging and Vision, 10 (1999), pp. 237–252. http://dx.doi.org/10.1023/A:1008344623873.

[14] A Witkin, *Scale-space filtering: A new approach to multi-scale description*, in IEEE International Conference on Acoustics Speech and Signal Processing, vol. 9, IEEE, 1984, pp. 150–153. http://dx.doi.org/10.1109/ICASSP.1984.1172729.

[15] I.T. Young and L.J. Van Vliet, *Recursive implementation of the Gaussian filter*, Signal processing, 44 (1995), pp. 139–151. http://dx.doi.org/10.1016/0165-1684(95)00020-E.