HAL
open science

How to Apply a Filter Defined in the Frequency Domain by a Continuous Function

Thibaud Briand, Jonathan Vacher

To cite this version:

Thibaud Briand, Jonathan Vacher. How to Apply a Filter Defined in the Frequency Domain by a Continuous Function. Image Processing On Line, 2016, 6, pp.183-211. 10.5201/ipol.2016.116 . hal-01687457

HAL Id: hal-01687457

https://hal.science/hal-01687457v1

Submitted on 18 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

2015/06/16 v0.5.1 IPOL article class

Published in Image Processing On Line on 2016–11–07.
Submitted on 2014–04–17, first version accepted on 2016–09–07, available at
http://www.ipol.im/pub/art/2016/116/revisions/2016-11-07/article.pdf.
Corrected of errors and republished on 2017-01-13.
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
https://doi.org/10.5201/ipol.2016.116

How to Apply a Filter Defined in the Frequency Domain by a Continuous Function

Thibaud Briand[1], Jonathan Vacher[2]

[1] Université Paris-Est, LIGM (UMR CNRS 8049), ENPC, F-77455 Marne-la-Vallée, France (thibaud.briand@enpc.fr)
[2] CEREMADE, CNRS-Université Paris-Dauphine, France (jonathan.vacher@ceremade.dauphine.fr)

Abstract

We propose algorithms for filtering real-valued images, when the filter is provided as a continuous function defined in the Nyquist frequency domain. This problem is ambiguous because images are discrete entities and there is no unique way to define the filtering. We provide a theoretical framework designed to analyse the classical and computationally efficient filtering implementations based on discrete Fourier transforms (DFT). In this framework, the filtering is interpreted as the convolution of a distribution, standing for the filter, with a trigonometric polynomial interpolator of the image. The various plausible interpolations and choices of the distribution lead to three equally licit algorithms which can be seen as method variants of the same standard filtering algorithm. In general none should be preferred to the others and the choice depends on the application. In practice, the method differences, which come from the boundary DFT coefficients, are not visible to the naked eye. We demonstrate that claim on several experimental configurations by varying the input image and the considered filter. In some cases however, we discuss how the choice of the variant may affect fundamental properties of the filtering.

Source Code

The source code and the online demo are accessible at the IPOL web part of this article[1]. The C99 implementation of the code that we provide is the one which has been peer reviewed and accepted by IPOL. It contains two modules : one performs the standard filtering algorithm (Algorithm 1) presented in Section 3 and the other performs the comparison detailed in Section 4. The comparison module computes for nine filters the filtered images obtained by applying the three variant methods. For each filter a pairwise comparison of the results is done by computing the (absolute) difference image, the maximum difference and the mean difference. It creates fifty-five images: twenty-seven filtered images, twenty-seven difference images and the modulus of the DFT input in logarithmic scale. See Section 4 and the code documentation for additional information.

Keywords: filtering; convolution; Fourier transform; DFT; trigonometric polynomial; interpolation

[1] https://doi.org/10.5201/ipol.2016.116

1 Introduction

The application to images of a filter defined by a continuous function in the frequency domain, more precisely in the Nyquist domain $[-\pi, \pi]^2$, is a well-known problem. Images are discrete entities so there is no unique way to define the filtering whether it be as a continuous or a discrete convolution.

The definition in the Fourier domain by a continuous function may be handy to impose a given property to a filter (e.g. low-pass, high-pass or steered filter). Moreover, it covers all the possible image sizes in a single formula. For instance this is a classical approach used to build multi-scale structures (e.g. the steerable pyramid of E. Simoncelli et al. [13]).

In practice, the filtering is usually applied by performing discrete Fourier transform (DFT) computations [2, 4, 8] but, to the best of our knowledge, no clear theoretical justification can be found in the literature. In particular, the interpretations generally involve a trigonometric polynomial interpolation, which is ambiguous for even-sized images, and rely on a convention whose influence on the boundary DFT coefficients should be taken into account.

The aim of this paper is to give a clear definition, as a continuous convolution, of the filtering of a discrete real-valued image by a filter specified by a continuous function in the Nyquist domain $[-\pi, \pi]^2$. Additionally, for computational purposes it should be compatible with DFT computations.

With this goal in mind, we interpret the filtering as the convolution of a distribution, standing for the filter, with a trigonometric polynomial interpolator of the image. The various plausible interpolations and choices of the distribution lead to three equally licit algorithms which can be seen as variants of the same standard filtering algorithm. In general none should be preferred to the others and the choice depends on the application. The method differences, which come from the boundary DFT coefficients, are not visible to the naked eye in practice. This analysis is illustrated by an application to several fundamental filters. When necessary, we also discuss the impact of the chosen method on some desired properties of the filtering (e.g. semi-group property, exact reconstruction).

The remainder of this paper is organized as follows: We present in Section 2 the theory intended to give a clear interpretation of the filtering of a discrete real-valued image by a filter defined by a continuous function in the Nyquist domain. It underlies the three proposed algorithms which are detailed in Section 3. Finally an experimental study with fundamental filter examples is presented in Section 4.

2 Theoretical Results

In this section, we introduce the theory intended to give a clear interpretation of the filtering of a discrete real-valued image $\underline{u}$ by a filter ϕ defined by a continuous function in the Nyquist domain $[-\pi, \pi]^2$. It is formalized as the convolution between a trigonometric polynomial $P_{\underline{u}}$, associated with the image $\underline{u}$, and a tempered distribution f_ϕ, associated with the filter ϕ, whose Fourier transform is continuous on $[-\pi, \pi]^2$. This theory is aimed at being consistent with the DFT implementations that are typical in the literature. In Section 2.2 we propose for $P_{\underline{u}}$ two canonical choices of trigonometric polynomial interpolators of $\underline{u}$. Section 2.3 justifies that the convolution between the tempered distribution f_ϕ and the trigonometric polynomial $P_{\underline{u}}$ is mathematically well sounded (see Theorem 2). In Section 2.4, according to the choice of $P_{\underline{u}}$ and f_ϕ, we consider three different but equally licit ways of defining the filtering of discrete images (see Definition 15). The resulting filtered images can be efficiently obtained by DFT based computations thanks to Proposition 4.

2.1 Definitions and Notations

First we introduce useful definitions and notations.

Definition 1 (Discrete domains)**.** *Let M and N be two positive integers. We define the discrete spatial domain $\Omega_{M,N} = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$.*
The discrete Fourier domain $\hat{\Omega}_{M,N}$, associated to $\Omega_{M,N}$, is defined by $\hat{\Omega}_{M,N} = \hat{\Omega}_M \times \hat{\Omega}_N$ where, for a positive integer L, we note

$$\hat{\Omega}_L = \begin{cases} \{-\frac{L-1}{2}, \dots, \frac{L-1}{2}\} & \text{if } L \text{ is odd,} \\ \{-\frac{L}{2}, \dots, \frac{L}{2} - 1\} & \text{if } L \text{ is even.} \end{cases}$$

The boundary $\Gamma_{M,N}$ of $\hat{\Omega}_{M,N}$ is defined by $\Gamma_{M,N} = \Big(\Gamma_M \times \hat{\Omega}_N\Big) \cup \Big(\hat{\Omega}_M \times \Gamma_N\Big)$ where, for a positive integer L, we note

$$\Gamma_L = \begin{cases} \emptyset & \text{if } L \text{ is odd,} \\ \{-\frac{L}{2}\} & \text{if } L \text{ is even.} \end{cases}$$

The symmetrized discrete Fourier domain $\hat{\Omega}^s_{M,N}$, associated to $\Omega_{M,N}$, is defined by $\hat{\Omega}^s_{M,N} = \hat{\Omega}^s_M \times \hat{\Omega}^s_N$ where, for a positive integer L, we note

$$\hat{\Omega}^s_L = \begin{cases} \hat{\Omega}_L & \text{if } L \text{ is odd} \\ \hat{\Omega}_L \cup \{\frac{L}{2}\} & \text{if } L \text{ is even.} \end{cases}$$

The boundary $\Gamma^s_{M,N}$ of $\hat{\Omega}^s_{M,N}$ is defined by $\Gamma^s_{M,N} = \hat{\Omega}^s_{M,N} \setminus \Big(\hat{\Omega}_{M,N} \setminus \Gamma_{M,N}\Big)$.

As an example, we display in Figure 1 the discrete domains $\hat{\Omega}_{M,N}$, $\hat{\Omega}^s_{M,N}$, $\Gamma_{M,N}$ and $\Gamma^s_{M,N}$ for $M = N = 4$.

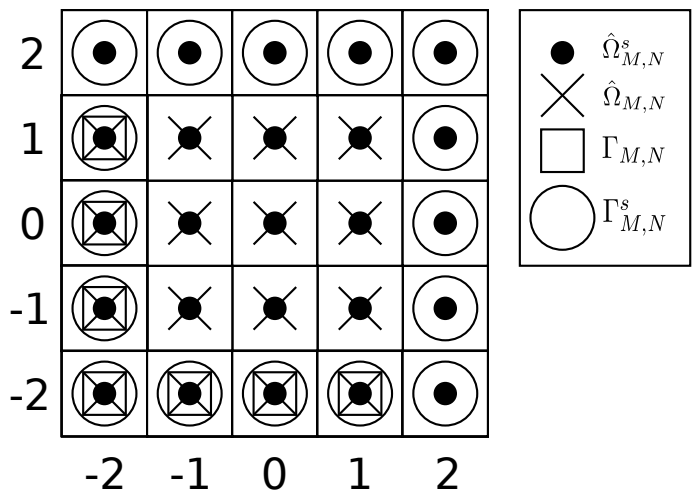


Figure 1: Discrete domains $\hat{\Omega}_{M,N}$, $\hat{\Omega}^s_{M,N}$, $\Gamma_{M,N}$ and $\Gamma^s_{M,N}$ for $M = N = 4$.

Definition 2 (Image)**.** *Let M and N be two positive integers. An image $\underline{u}$ of size $M \times N$ is defined as a two-dimensional finite list of complex numbers $(\underline{u}_{k,l})_{(k,l)\in\Omega_{M,N}}$. The image is said to be real-valued when every number is real.*

Definition 3 (Trigonometric polynomial)**.** *Let M and N be two positive integers. A function $P : \mathbb{R}^2 \to \mathbb{C}$ is said to be a trigonometric polynomial of degree $M \times N$ if there exists $c \in \mathbb{C}^{\hat{\Omega}^s_{M,N}}$, called coefficients of P, s.t.*

$$\forall (x, y) \in \mathbb{R}^2, \quad P(x, y) = \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} c_{m,n} e^{2i\pi(x\frac{m}{M} + y\frac{n}{N})}. \tag{1}$$

The set of trigonometric polynomials of degree $M \times N$, denoted $\mathcal{P}_{M,N}$, is a subset of the set of $(M,N)-$periodic functions. Denote $\mathcal{R}$ the real part operator and let $P \in \mathcal{P}_{M,N}$ then $\mathcal{R}(P) \in \mathcal{P}_{M,N}$.

Definition 4 (Discrete Fourier transform)**.** *Let M and N be two positive integers. The discrete Fourier transform (DFT) of $\underline{u} \in \mathbb{C}^{\Omega_{M,N}}$ is denoted $\mathcal{F}_{M,N}(\underline{u}) \in \mathbb{C}^{\hat{\Omega}_{M,N}}$ and is defined by*

$$\forall (m,n) \in \hat{\Omega}_{M,N}, \quad \mathcal{F}_{M,N}(\underline{u})_{m,n} = \sum_{(k,l)\in\Omega_{M,N}} \underline{u}_{k,l} e^{-2\pi i (k\frac{m}{M} + l\frac{n}{N})}.$$

The inverse discrete Fourier transform (iDFT) of $\underline{v} \in \mathbb{C}^{\hat{\Omega}_{M,N}}$ is denoted $\mathcal{F}^{-1}_{M,N}(\underline{v}) \in \mathbb{C}^{\Omega_{M,N}}$ and is defined by

$$\forall (k,l) \in \Omega_{M,N}, \quad \mathcal{F}^{-1}_{M,N}(\underline{v})_{k,l} = \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}_{M,N}} \underline{v}_{m,n} e^{2\pi i (m\frac{k}{M} + n\frac{l}{N})}.$$

Definition 5 (Fourier transform)**.** *The continuous Fourier transform applied to $L^1(\mathbb{R}^2)$ and its inverse, denoted $\mathcal{F}$ and $\mathcal{F}^{-1}$, are defined by*

$$\forall u \in L^1(\mathbb{R}^2), \quad \forall (\xi,\nu) \in \mathbb{R}^2, \quad \hat{u}(\xi,\nu) = \mathcal{F}(u)(\xi,\nu) = \int_{\mathbb{R}^2} u(x,y) e^{-i(x\xi+y\nu)} dxdy$$

$$\textit{and} \quad \forall (x,y) \in \mathbb{R}^2, \quad \mathcal{F}^{-1}(u)(x,y) = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} u(\xi,\nu) e^{i(x\xi+y\nu)} d\xi d\nu.$$

The Fourier transform extended to tempered distributions [12, 14] is still denoted $\mathcal{F}$.

Definition 6 (Convolution)**.** *Let $u \in L^1(\mathbb{R}^2)$ and $v \in L^\infty(\mathbb{R}^2)$. The convolution of u and v, denoted $u * v$, is defined by*

$$\forall (x,y) \in \mathbb{R}^2, \quad u * v(x,y) = \int_{\mathbb{R}^2} u(x-s, y-t) v(s,t) dsdt.$$

2.2 Trigonometric Polynomial Interpolators

Let us construct interpolators of real-valued images (see Definition 2) which are trigonometric polynomials (see Definition 3).

Definition 7 (Trigonometric polynomial interpolator in complex convention)**.** *Let M and N be two positive integers and let $\underline{u}$ be a real-valued image of size $M \times N$. The trigonometric polynomial interpolator of $\underline{u}$ in complex convention is the trigonometric polynomial $P^{(c)}_{\underline{u}} \in \mathcal{P}_{M,N}$ defined by*

$$\forall (x,y) \in \mathbb{R}^2, \quad P^{(c)}_{\underline{u}}(x,y) = \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}_{M,N}} \mathcal{F}_{M,N}(\underline{u})_{m,n} e^{2i\pi (x\frac{m}{M} + y\frac{n}{N})}. \tag{2}$$

Proposition 1. *Let M and N be two positive integers and let $\underline{u}$ be a real-valued image of size $M \times N$. Then, $P^{(c)}_{\underline{u}}$ interpolates $\underline{u}$ in the sense that*

$$\forall (k,l) \in \Omega_{M,N}, \quad P^{(c)}_{\underline{u}}(k,l) = \underline{u}_{k,l}.$$

Proof. It is a direct consequence of the DFT and iDFT formulas. For $(k,l) \in \Omega_{M,N}$ we have $P^{(c)}_{\underline{u}}(k,l) = \left(\mathcal{F}^{-1}_{M,N}(\mathcal{F}_{M,N}(\underline{u})\right)_{k,l} = \underline{u}_{k,l}$. □

Proposition 1 states that the trigonometric polynomial interpolator in complex convention introduced in Definition 7 is an interpolating function. It is the natural and simplest way to define a trigonometric polynomial interpolator since the coefficients are directly expressed in terms of the DFT of the image. However it may be complex-valued while the image is real-valued. Thus, we build another interpolator that is real-valued.

Definition 8 (Trigonometric polynomial interpolator in real convention)**.** *Let M and N be two positive integers and let $\underline{u}$ be a real-valued image of size $M \times N$. The trigonometric polynomial interpolator of $\underline{u}$ in real convention is defined as the trigonometric polynomial $P_{\underline{u}}^{(r)} \in \mathcal{P}_{M,N}$ whose coefficients c verify*

$$\forall (m,n) \in \hat{\Omega}^s_{M,N}, \quad c_{m,n} = \begin{cases} \mathcal{F}_{M,N}(\underline{u})_{m,n} & \text{if } |m| < \frac{M}{2} \text{ and } |n| < \frac{N}{2} \\ \frac{1}{2}\mathcal{F}_{M,N}(\underline{u})_{-\frac{M}{2},n} & \text{if } |m| = \frac{M}{2} \text{ and } |n| < \frac{N}{2} \\ \frac{1}{2}\mathcal{F}_{M,N}(\underline{u})_{m,-\frac{N}{2}} & \text{if } |m| < \frac{M}{2} \text{ and } |n| = \frac{N}{2} \\ \frac{1}{4}\mathcal{F}_{M,N}(\underline{u})_{-\frac{M}{2},-\frac{N}{2}} & \text{if } |m| = \frac{M}{2} \text{ and } |n| = \frac{N}{2}. \end{cases}$$

Note that this formulation is valid for both even and odd size parameters (M and N). However, some of the cases in the disjunction never occur when one or both size parameters are odd[2]*.*

Proposition 2. *Let M and N be two positive integers and let $\underline{u}$ be a real-valued image of size $M \times N$. Then $P_{\underline{u}}^{(r)}$ is a real-valued interpolating function of $\underline{u}$ since for $(x,y) \in \mathbb{R}^2$,*

$$P_{\underline{u}}^{(r)}(x,y) = \begin{cases} \mathcal{R}(P_{\underline{u}}^{(c)})(x,y) + \mathcal{F}_{M,N}(\underline{u})_{-\frac{M}{2},-\frac{N}{2}} \sin(\pi x)\sin(\pi y) & \text{if } M \text{ and } N \text{ are even} \\ \mathcal{R}(P_{\underline{u}}^{(c)})(x,y) & \text{otherwise.} \end{cases}$$

Proof. This is a consequence of the fact that the DFT of a real sequence has a Hermitian symmetry. Indeed, extending the DFT by $M,N-$periodicity we easily get $\mathcal{F}_{M,N}(\underline{u})_{m,n} = \overline{\mathcal{F}_{M,N}(\underline{u})_{-m,-n}}$ for all $(m,n) \in \mathbb{Z}^2$. Thus by grouping the terms of the sum in Equation (2) we can deduce that the complex contribution comes from the terms with indices $\frac{M}{2}$ or $\frac{N}{2}$. For instance if M is even, the result is obtained by considering relations of the type $\mathcal{F}_{M,N}(\underline{u})_{-\frac{M}{2},n} = \overline{\mathcal{F}_{M,N}(\underline{u})_{-\frac{M}{2},-n}}$ and grouping terms of indices $(-\frac{M}{2},n)$ and $(-\frac{M}{2},-n)$. If M and N are even, we have $\mathcal{F}_{M,N}(\underline{u})_{-\frac{M}{2},-\frac{N}{2}} \in \mathbb{R}$ and we conclude using the relation $\mathcal{R}(e^{-i\pi(x+y)}) + \sin(\pi x)\sin(\pi y) = \frac{1}{4}\left(e^{i\pi(x+y)} + e^{-i\pi(x+y)} + e^{i\pi(x-y)} + e^{i\pi(-x+y)}\right)$. □

Let M and N be two integers and suppose that $\underline{u}$ is of size $M \times N$. The coefficients in the two conventions are potentially different[3] only for indices $(m,n) \in \Gamma^s_{M,N}$ on the boundary of $\hat{\Omega}^s_{M,N}$ i.e. if $|m| = \frac{M}{2}$ or $|n| = \frac{N}{2}$. In particular, if M and N are odd numbers both conventions match and there is a unique trigonometric polynomial of degree $M \times N$ interpolating $\underline{u}$. When M or N is an even number, the uniqueness is lost. When for instance M is even, one can easily check that the interpolation property is kept when adding any multiple of the trigonometric polynomial of degree $M \times N$ defined by $(x,y) \in \mathbb{R}^2 \mapsto \sin(\pi x)$. Taking the real part of $P_{\underline{u}}^{(c)}$ boils down to correcting the eventual asymmetry of the DFT coefficients. When M and N are even $P_{\underline{u}}^{(r)}$ is preferred to $\mathcal{R}(P_{\underline{u}}^{(c)})$ because it has the same coefficients at the four corners of $\hat{\Omega}^s_{M,N}$. Proposition 2 is a reformulation of Equation (23) in [1]. This interpolation method is said to be DFT based since the coefficients in both conventions depend on the DFT of $\underline{u}$.

[2]For instance, if M is odd, $|m| = \frac{M}{2}$ is not possible within $\hat{\Omega}^s_{M,N}$

[3]Actually the polynomial coefficients located at the boundary are equal if and only if they are null.

2.3 Convolution of Trigonometric Polynomials

In this part we define the convolution between a tempered distribution whose Fourier transform is a continuous function on $[-\pi,\pi]^2$ and a trigonometric polynomial. This convolution will be used in Section 2.4 to define the filtering of an image. The distribution will represent the filter and the trigonometric polynomial the image. To do this we extend known results on the set of integrable functions. As a trigonometric polynomial is bounded, its convolution with an integrable function is well-defined (see Definition 6).

Theorem 1 (Convolution theorem 1 [7])**.** *Let M and N be two positive integers. Let $f \in L^1(\mathbb{R}^2)$ and $P \in \mathcal{P}_{M,N}$ with coefficients c. Then, $f * P \in \mathcal{P}_{M,N}$ and verifies*

$$\forall (x,y) \in \mathbb{R}^2, \quad (f*P)(x,y) = \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} \hat{f}\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})}.$$

Theorem 1 states that this convolution is also a trigonometric polynomial whose coefficients depend on the values of the Fourier transform of the integrable function on $[-\pi,\pi]^2$ and the trigonometric polynomial coefficients. We now want to extend this result to the wider set of distributions defined as follows.

Definition 9. *A distribution T of the plane is said to be continuous on $[-\pi,\pi]^2$ when its restriction*[4] *$T_{|]-\pi,\pi[^2}$ is a function of $]-\pi,\pi[^2$ admitting a continuous extension to $[-\pi,\pi]^2$. By construction, this extension is unique.*

Definition 10. *We define CF_π as the subspace of tempered distributions of the plane whose Fourier transforms are continuous on $[-\pi,\pi]^2$ (in the sense of Definition 9).*

Note that we have $L^1(\mathbb{R}^2) \subset CF_\pi$ since the Fourier transform of an integrable function is continuous on $\mathbb{R}^2$. As the extension of the convolution to CF_π will only depend on the Fourier transforms of the distributions on $[-\pi,\pi]^2$ we consider the following equivalence relation and quotient space.

Definition 11. *We define the equivalence relation $\mathcal{R}$ on CF_π by*

$$f\mathcal{R}g \Leftrightarrow \hat{f} = \hat{g} \text{ on } [-\pi,\pi]^2.$$

The set of equivalence classes is denoted $\langle CF_\pi\rangle$ and the equivalence class of $f \in CF_\pi$ is denoted $\langle f\rangle$. We define on the vectorial space $\langle CF_\pi\rangle$ the norm $N : \langle CF_\pi\rangle \to \mathbb{R}^+$ by

$$\forall \langle f\rangle \in \langle CF_\pi\rangle, \quad N(\langle f\rangle) = \sup_{[-\pi,\pi]^2} |\hat{f}|.$$

The equivalence classes which admit an integrable representative play a special role since we have the following density result.

Proposition 3. *Let $\langle L^1(\mathbb{R}^2)\rangle = \{\langle f\rangle \in \langle CF_\pi\rangle \quad | \quad f \in L^1(\mathbb{R}^2)\}$. Then $\langle L^1(\mathbb{R}^2)\rangle$ is dense in $(\langle CF_\pi\rangle, N)$.*

Proof. Let $\langle f\rangle \in \langle \mathrm{CF}_\pi\rangle$. Starting from the definitions of CF_π and $\langle \mathrm{CF}_\pi\rangle$ we can easily build $g \in \mathrm{CF}_\pi$ such that $\langle g\rangle = \langle f\rangle$ and $\hat{g}$ is a continuous function on $\mathbb{R}^2$. Now let $(h_n)_{n\in\mathbb{N}}$ be a sequence of Schwartz functions (on $\mathbb{R}^2$) that converges uniformly to $\hat{g}$ on $[-\pi,\pi]^2$. Then for each $n \in \mathbb{N}$, $g_n = \mathcal{F}^{-1}(h_n)$ is also a Schwartz function and thus is in $L^1(\mathbb{R}^2)$. By construction, $(\langle g_n\rangle)_{n\in\mathbb{N}} \in \langle L^1(\mathbb{R}^2)\rangle^{\mathbb{N}}$ and converges in the norm N to $\langle g\rangle = \langle f\rangle$. □

[4] Note that we can define distribution restrictions only on open sets.

Starting from Theorem 1 we can define the convolution between an element of $\langle L^1(\mathbb{R}^2)\rangle$ and a trigonometric polynomial.

Definition 12. *Let M and N be two positive integers and let $P \in \mathcal{P}_{M,N}$ with coefficients c. Let $\langle f\rangle \in \langle L^1(\mathbb{R}^2)\rangle$. We define the convolution of $\langle f\rangle$ and P, still denoted $\langle f\rangle * P$, as the trigonometric polynomial of degree $M \times N$ verifying*

$$\forall (x,y) \in \mathbb{R}^2, \quad (\langle f\rangle * P)(x,y) = \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} \hat{f}\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})}.$$

This definition is compatible with the classical definition of the convolution given in Definition 6. It is extended to $\langle \mathrm{CF}_\pi\rangle$ in the following theorem.

Theorem 2 (Convolution theorem 2)**.** *Let M and N be two positive integers and $P \in \mathcal{P}_{M,N}$. The bounded linear transformation $T_P : \langle f\rangle \in (\langle L^1(\mathbb{R}^2)\rangle, N) \mapsto \langle f\rangle * P \in (\mathcal{P}_{M,N}, \|.\|_\infty)$ can be uniquely extended to a bounded linear transformation $\tilde{T}_P : (\langle CF_\pi\rangle, N) \to (\mathcal{P}_{M,N}, \|.\|_\infty)$.*

Let $\langle f\rangle \in \langle CF_\pi\rangle$ then $\tilde{T}_P(\langle f\rangle)$ is called the convolution of $\langle f\rangle$ and P. It is denoted $\langle f\rangle * P$ and verifies

$$\forall (x,y) \in \mathbb{R}^2, \quad (\langle f\rangle * P)(x,y) = \tilde{T}_P(\langle f\rangle)(x,y) = \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} \hat{f}\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})},$$

where c denotes the coefficients of P. The convolution of $f \in CF_\pi$ with a trigonometric polynomial P, still denoted $f * P$, is then defined by $f * P = \langle f\rangle * P$.

Proof. The transformation T_P is clearly linear and is bounded since

$$\forall \langle f\rangle \in \langle \mathrm{CF}_\pi\rangle, \quad \|T_P(\langle f\rangle)\|_\infty \le \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} \left|\hat{f}\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n}\right| \le \left(\frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} c_{m,n}\right) N(\langle f\rangle).$$

As $(\mathcal{P}_{M,N}, \|.\|_\infty)$ is a complete normed linear space and $\langle L^1(\mathbb{R}^2)\rangle$ is dense in $(\langle \mathrm{CF}_\pi\rangle, N)$ (see Proposition 3) then the bounded linear transformation (B.L.T) theorem [10, p. 9] states that T_P can be uniquely extended to a bounded linear transformation $\tilde{T}_P : (\langle \mathrm{CF}_\pi\rangle, N) \to (\mathcal{P}_{M,N}, \|.\|_\infty)$.

Let $\langle f\rangle \in \langle \mathrm{CF}_\pi\rangle$ and consider a sequence $(\langle f_k\rangle)_{k\in\mathbb{N}} \in \langle L^1(\mathbb{R}^2)\rangle^N$ that converges to $\langle f\rangle$ in norm N. Define

$$g_{\langle f\rangle,P} : (x,y) \in \mathbb{R}^2 \mapsto \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} \hat{f}\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})}.$$

Then,

$$\begin{aligned} \left\|\langle f_k\rangle * P - g_{\langle f\rangle,P}\right\|_\infty &\le \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} \left|(\hat{f} - \hat{f}_k)\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n}\right| \\ &\le \left(\frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} |c_{m,n}|\right) N(f - f_k) \\ &\xrightarrow[k\to\infty]{} 0. \end{aligned}$$

Thus by continuity we have $\tilde{T}_P = g_{\langle f\rangle,P}$. □

Therefore, Theorem 2 provides a sound definition for the convolution of a tempered distribution whose Fourier transform is continuous on $[-\pi,\pi]^2$ with a trigonometric polynomial. The resulting distribution is itself also a trigonometric polynomial. It is also interesting to point out that this convolution is entirely independent from the distribution of $\hat{f}$ outside $[-\pi,\pi]^2$.

2.4 Discrete Image Filtering

Thanks to Theorem 2 we can define the filtering of discrete real-valued images. A continuous function on $[-\pi,\pi]^2$ is sufficient to define a filter. Let $\phi : [-\pi,\pi]^2 \to \mathbb{C}$, then in the rest of the paper we will indistinctly refer to ϕ as a function that defines a filter and the filter itself.

Definition 13. *Let $\phi : [-\pi,\pi]^2 \to \mathbb{C}$ be a continuous function. We denote by f_ϕ any element of the equivalence class*

$$\left\{ f \in CF_\pi \quad | \quad \hat{f} = \phi \text{ on } [-\pi,\pi]^2 \right\}.$$

In Definition 13 we introduce f_ϕ as any representative of the equivalence class of tempered distributions whose Fourier transforms are equal to the filter ϕ on $[-\pi,\pi]^2$. The filtering of real-valued images by ϕ can be defined as the convolution[5] of f_ϕ with trigonometric polynomial interpolators of the images. The choice of the interpolation convention provides two equally licit definitions of the filtering. A third one is obtained by considering a slight modification of the filter that removes the ambiguity due to the Fourier coefficients located at the boundary of the Nyquist domain.

Definition 14. *Let M and N be two positive integers and $\phi : [-\pi,\pi]^2 \to \mathbb{C}$ be a continuous function. We define the continuous function $\phi_{M,N} : [-\pi,\pi]^2 \to \mathbb{C}$ by*

$$\forall (\mu,\nu) \in \mathbb{R}^2, \quad \phi_{M,N}(\mu,\nu) = \phi(\mu,\nu) l_M(|\mu|) l_N(|\nu|)$$

where, for a positive integer L, we note

$$l_L : x \in [0,\pi] \mapsto \begin{cases} 1 & \text{if } 0 \le x \le \pi - \frac{1}{L} \\ L(\pi - x) & \text{if } \pi - \frac{1}{L} < x \le \pi. \end{cases}$$

As an example, we display in Figure 2 the continuous piece-wise linear function l_L near π for $L = 32$.

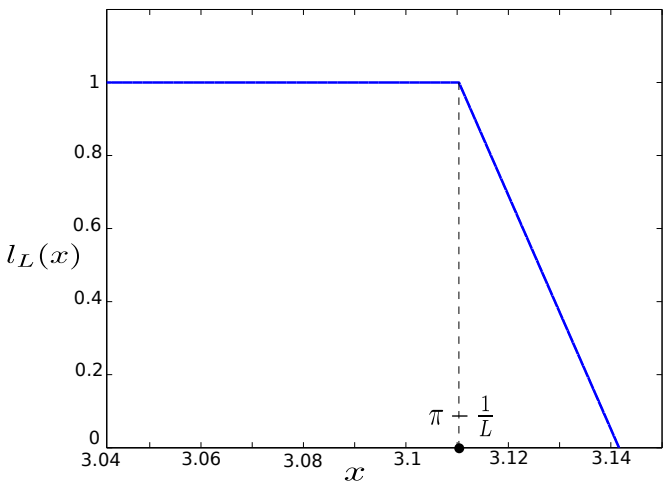


Figure 2: The continuous piece-wise linear function l_L near π for $L = 32$.

Let M and N be two positive integers. The function $\phi_{M,N}$ introduced in Definition 14 is constructed so that for $(m,n) \in \hat{\Omega}^s_{M,N}$,

$$\phi_{M,N}\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) = \begin{cases} \phi\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) & \text{if } |m| < \frac{M}{2} \text{ and } |n| < \frac{N}{2} \\ 0 & \text{if } (m,n) \in \Gamma^s_{M,N}. \end{cases}$$

[5] We recall that this convolution does not depend on the choice of the representative f_ϕ.

This entails that for any image $\underline{u}$ of size $M \times N$ we have $f_{\phi_{M,N}} * P_{\underline{u}}^{(c)} = f_{\phi_{M,N}} * P_{\underline{u}}^{(r)}$. It allows us to define the filtering of real-valued images as below.

Definition 15 (Filtering real-valued images)**.** *Let $\phi : [-\pi, \pi]^2 \to \mathbb{C}$ be a filter. Let M and N be two positive integers and $\underline{u}$ be a real-valued image of size $M \times N$.*

1. *The filtering $\phi_1(\underline{u})$ of $\underline{u}$ by ϕ in complex convention is defined as the convolution*
$$\phi_1(\underline{u}) = f_\phi * P_{\underline{u}}^{(c)}.$$
2. *The filtering $\phi_2(\underline{u})$ of $\underline{u}$ by ϕ in real convention is defined as the convolution*
$$\phi_2(\underline{u}) = f_\phi * P_{\underline{u}}^{(r)}.$$
3. *The windowed filtering $\phi_3(\underline{u})$ of $\underline{u}$ by ϕ is defined as the convolution*
$$\phi_3(\underline{u}) = f_{\phi_{M,N}} * P_{\underline{u}}$$
where $P_{\underline{u}}$ can be taken indifferently as $P_{\underline{u}}^{(c)}$ or $P_{\underline{u}}^{(r)}$.

Let $j \in \{1, 2, 3\}$. The filtered image $\underline{\phi_j(\underline{u})}$ is defined as the image canonically associated to the filtering $\phi_j(\underline{u})$ i.e
$$\underline{\phi_j(\underline{u})} = (\phi_j(\underline{u})(k, l))_{(k,l)\in\Omega_{M,N}}.$$

In Definition 15 the three proposed results for the filtering of an image $\underline{u}$ by the filter ϕ are trigonometric polynomials whose coefficients depend on samples of ϕ and on the DFT of $\underline{u}$. Applying Theorem 2 we see that their coefficients only differ at the boundary $\Gamma_{M,N}^s$ of $\hat{\Omega}_{M,N}^s$. In particular when M and N are odd numbers, the three definitions are equivalent.

Now let us present how the resulting filtered images can be obtained by using DFT computations.

Definition 16 (Spectral samples)**.** *Let $\phi : [-\pi, \pi]^2 \to \mathbb{C}$ be a filter. Let M and N be two positive integers.*

1. *The spectral samples $(S_{m,n}^{1,M,N})_{(m,n)\in\hat{\Omega}_{M,N}}$ of size $M \times N$ in the complex convention are defined for all $(m, n) \in \hat{\Omega}_{M,N}$ by*
$$S_{m,n}^{1,M,N} = \phi\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right). \tag{3}$$
2. *The spectral samples $(S_{m,n}^{2,M,N})_{(m,n)\in\hat{\Omega}_{M,N}}$ of size $M \times N$ in the real convention are defined for all $(m, n) \in \hat{\Omega}_{M,N}$ by*
$$S_{m,n}^{2,M,N} = \begin{cases} \phi(\frac{2\pi m}{M}, \frac{2\pi n}{N}) & \text{if } (m,n) \notin \Gamma_{M,N} \\ \frac{1}{2}\left(\phi(-\pi, \frac{2\pi n}{N}) + \phi(\pi, \frac{2\pi n}{N})\right) & \text{if } m = -\frac{M}{2}, n \neq -\frac{N}{2} \\ \frac{1}{2}\left(\phi(\frac{2\pi m}{M}, -\pi) + \phi(\frac{2\pi m}{M}, \pi)\right) & \text{if } m \neq -\frac{M}{2}, n = -\frac{N}{2} \\ \frac{1}{4}\left(\phi(-\pi,-\pi) + \phi(\pi,-\pi) + \phi(-\pi,\pi) + \phi(\pi,\pi)\right) & \text{if } m = -\frac{M}{2}, n = -\frac{N}{2}. \end{cases} \tag{4}$$
3. *The spectral windowed samples $(S_{m,n}^{3,M,N})_{(m,n)\in\hat{\Omega}_{M,N}}$ of size $M \times N$ are defined for all $(m, n) \in \hat{\Omega}_{M,N}$ by*
$$S_{m,n}^{3,M,N} = \begin{cases} \phi(\frac{2\pi m}{M}, \frac{2\pi n}{N}) & \text{if } (m,n) \notin \Gamma_{M,N} \\ 0 & \text{if } (m,n) \in \Gamma_{M,N}. \end{cases} \tag{5}$$

Proposition 4. *Let $\phi : [-\pi, \pi]^2 \to \mathbb{C}$ be a filter. Let M and N be two positive integers and $\underline{u}$ be a real-valued image of size $M \times N$. Let $j \in \{1, 2, 3\}$. Then, the DFT of the filtered image $\underline{\phi_j(\underline{u})}$ is given for all $(m, n) \in \hat{\Omega}_{M,N}$ by*

$$\mathcal{F}_{M,N}\left(\underline{\phi_j(\underline{u})}\right)_{m,n} = S^{j,M,N}_{m,n} \mathcal{F}_{M,N}(\underline{u})_{m,n}.$$

Proof. We carry the proof for all three cases in turn.

Complex convention : With Theorem 2 and Definition 13 we have for $(x, y) \in \mathbb{R}^2$,

$$\begin{aligned}\phi_1(\underline{u})(x,y) = (f_\phi * P^{(c)}_{\underline{u}})(x,y) &= \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}_{M,N}} \phi\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) \mathcal{F}_{M,N}(\underline{u})_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})} \\ &= \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}_{M,N}} S^{1,M,N}_{m,n} \mathcal{F}_{M,N}(\underline{u})_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})}.\end{aligned}$$

We conclude by evaluating at $(x, y) = (k, l) \in \Omega_{M,N}$.

Real convention : Similarly we have for $(x, y) \in \mathbb{R}^2$,

$$\phi_2(\underline{u})(x,y) = (f_\phi * P^{(r)}_{\underline{u}})(x,y) = \frac{1}{MN} \sum_{(m,n)\in\hat{\Omega}^s_{M,N}} \phi\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})}$$

where the coefficients c are given in Definition 8. Using the partition $\hat{\Omega}^s_{M,N} = \hat{\Omega}_{M,N} \backslash \Gamma_{M,N} \sqcup \Gamma^s_{M,N}$ we get

$$\phi_2(\underline{u})(x,y) = \frac{1}{MN} \left(\sum_{(m,n)\in\hat{\Omega}_{M,N}\backslash\Gamma_{M,N}} + \sum_{(m,n)\in\Gamma^s_{M,N}} \right) \phi\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right) c_{m,n} e^{2i\pi(x\frac{m}{M}+y\frac{n}{N})}.$$

Now let $(x, y) = (k, l) \in \Omega_{M,N}$. By using the simple identity $e^{2i\pi\frac{j}{L}\frac{L}{2}} = (-1)^j = e^{2i\pi\frac{j}{L}\frac{-L}{2}}$, we can perform a straightforward identification of the DFT coefficients of the boundary $\Gamma_{M,N}$.

Windowed filtering : The proof is similar to the complex convention one by replacing $\phi\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right)$ by 0 for $(m, n) \in \Gamma_{M,N}$.

□

As stated in Proposition 4, the filtered image (see Definition 15) can be characterized by its DFT which is the element-wise product of the DFT of the image $\underline{u}$ and the spectral samples (introduced in Definition 16). This observation underlies the efficient FFT based algorithm presented in Section 3.

3 Algorithms

In this section we propose three filtering algorithms taking as inputs a discrete image $\underline{u}$ and a filter specified by a continuous function ϕ defined on the Nyquist domain $[-\pi, \pi]^2$. They are presented as three variants of Algorithm 1, namely the common standard filtering algorithm, as follows:

- Method 1: Filtering in complex convention.
  The spectral samples are the $S^{1,M,N}$ given by Equation (3).

- Method 2: Filtering in real convention.
  The spectral samples are the $S^{2,M,N}$ given by Equation (4).

- Method 3: Windowed filtering.
  The spectral samples are the $S^{3,M,N}$ given by Equation (5).

We summarize in Table 1 how the spectral samples are computed according to the method variant.

Filtering method	1	2	3
Spectral samples	$S^{1,M,N}$ given by Equation (3)	$S^{2,M,N}$ given by Equation (4)	$S^{3,M,N}$ given by Equation (5)

Table 1: How to compute the spectral samples according to the filtering method.

Algorithm 1: Standard filtering algorithm

Input : A real-valued image $\underline{u}$ of size $M \times N$, a filter $\phi : [-\pi, \pi]^2 \to \mathbb{C}$ (continuous function) and the number $j \in \{1, 2, 3\}$ of the method variant.

Output: The filtered image $\underline{v}$ of size $M \times N$ corresponding to the method j.

1 Compute $\tilde{u} = \mathcal{F}_{M,N}(\underline{u})$ the DFT of $\underline{u}$.

2 Compute $S = (S_{m,n})_{(m,n)\in\hat{\Omega}_{M,N}}$ the spectral samples corresponding to method j (see Table 1).

3 Compute $\tilde{v} = (\tilde{u}_{m,n} S_{m,n})_{(m,n)\in\hat{\Omega}_{M,N}}$ the element-wise multiplication of $\tilde{u}$ and S.

4 Compute $\underline{v} = \mathcal{F}^{-1}_{M,N}(\tilde{v})$ the inverse DFT of $\tilde{v}$.

These algorithms rely on DFT computations whose interpretations are given in Section 2. Let M and N be two positive integers. For an image $\underline{u}$ of size $M \times N$ the filtered image is computed in $\mathcal{O}(MN \log(MN))$ operations thanks to Fast Fourier transform (FFT) algorithms [3]. Color images can be filtered by applying the algorithms independently to each color channel. The filtering in complex convention (method 1) corresponds to the classical DFT based computations that can be found in the literature [2]. Note that the three methods can be naturally extended to complex-valued images.

The three methods only differ by the computation of the spectral samples which may vary at the boundary $\Gamma_{M,N}$ of $\hat{\Omega}_{M,N}$. In particular suppose the sizes M and N are odd (i.e. $\Gamma_{M,N} = \emptyset$) then the three methods are equivalent. Similarly, suppose the continuous function ϕ that defines the filter is null at the boundary of the Nyquist domain then the three methods are equivalent for any input image of any size.

These algorithms are applicable for any image size and any filter that can be defined by a continuous function in the Nyquist domain. The current general form can be modified in particular cases. For instance, suppose we want to apply a filter to a set of images of the same sizes. Then the filtering only depends (with respect to the filter) on the spectral samples of ϕ which can be precomputed. This discretization step entails a loss of information that may lead to the following degenerated situation:

- Two different filters may lead to the same filtered images for a fixed size.

- It is easy to construct a filter that behaves in very different ways for different sizes.

4 Experiments

In this experimental part we apply the three methods introduced in the previous section to several fundamental filters. The experiments are reproducible with the source code provided in the IPOL web part of this article[6]. We start by presenting the filters before comparing the results of the different methods applied to an experimental set of four images.

4.1 Filters

We propose to consider the following fundamental filters.

4.1.1 Cardinal Sine (*sinc*) filter

The first filter proposed is the simplest one since it is defined by a constant function on the Nyquist domain.

Definition 17 (Cardinal sine (*sinc*) filter)**.** *The cardinal sine (or sinc) filter*[7] *is defined by the constant function*

$$\phi_{sinc} : (\xi, \nu) \in [-\pi, \pi]^2 \mapsto 1.$$

Since ϕ_{sinc} is symmetric at the boundary of $[-\pi, \pi]^2$ the spectral samples in both conventions coincide i.e. method 1 and method 2 are equivalent. The filtered image obtained by applying method 1 (or 2) is exactly the input image. This property is in general false for method 3.

4.1.2 Shift Filter

Shifting an image is common in image processing. It can be done by using the following filter.

Definition 18 (Shift filter)**.** *Let $a = (a_1, a_2) \in \mathbb{R}^2$. The shift filter of parameter a is defined by the function*

$$\phi_a : (\xi, \nu) \in [-\pi, \pi]^2 \mapsto e^{i(a_1\xi + a_2\nu)} \in \mathbb{C}.$$

4.1.3 Derivative Filter

The derivative filters are used to highlight the variations along an axis.

Definition 19 (Derivative filter)**.** *The partial derivative filter with respect to the variable x is defined by the function*

$$\varphi_x : (\xi, \nu) \in [-\pi, \pi]^2 \mapsto i\xi \in \mathbb{C}.$$

Similarly the partial derivative filter with respect to the variable y is defined by the function

$$\varphi_y : (\xi, \nu) \in [-\pi, \pi]^2 \mapsto i\nu \in \mathbb{C}.$$

In the following we choose to concentrate on the filter φ_x (the results for φ_y are similar). Let M and N be the size of the image to be filtered. We notice that when M is even and N is odd the spectral samples of method 2 are null at the boundary $\Gamma_{M,N}$ i.e. method 2 and method 3 are equivalent. When M is odd and N is even the spectral samples of method 1 and method 2 are the same i.e. method 1 and method 2 are equivalent. Of course, if both M and N are odd, all three methods coincide (since this is true for any filter).

[6] https://doi.org/10.5201/ipol.2016.116

[7] The name of the filter comes from the separable cardinal sine function, which Fourier transform is the indicator function of $[-\pi, \pi]^2$.

4.1.4 Laplacian Filter

The Laplacian filter is used to highlight the high frequencies of images. In particular, it is often used for edge detection.

Definition 20 (Laplacian filter)**.** *The Laplacian filter is defined by the function*

$$L : (\xi, \nu) \in [-\pi, \pi]^2 \mapsto -\xi^2 - \nu^2 \in \mathbb{C}.$$

Since L is a radial function the spectral samples in both conventions coincide i.e. method 1 and method 2 are equivalent.

4.1.5 Gaussian Filter

It is common to use a Gaussian filter (e.g. to blur images).

Definition 21 (Gaussian filter)**.** *Let $\sigma > 0$. The Gaussian filter of standard deviation σ is defined by the function*

$$g_\sigma : (\xi, \nu) \in [-\pi, \pi]^2 \mapsto e^{-\sigma^2 \frac{\xi^2+\nu^2}{2}} \in \mathbb{R}.$$

Since g_σ is a radial function the spectral samples in both conventions coincide i.e. method 1 and method 2 are equivalent.

Let us justify the definition of the filter. Let $\sigma > 0$ and define $f_\sigma : (x, y) \in \mathbb{R}^2 \mapsto \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$. Then we have,

$$\forall (\xi, \nu) \in [-\pi, \pi]^2, \quad \hat{f}_\sigma(\xi, \nu) = g_\sigma(\xi, \nu).$$

The convolution with f_σ, namely Gaussian convolution [4, 11], is a common operation and building block for algorithms in image processing. Note that for all the three methods, the sequential application of g_σ and $g_{\sigma'}$ is equivalent to the direct application of $g_{\sqrt{\sigma^2+\sigma'^2}}$ (semi-group property).

4.1.6 Steerable Pyramid Filters

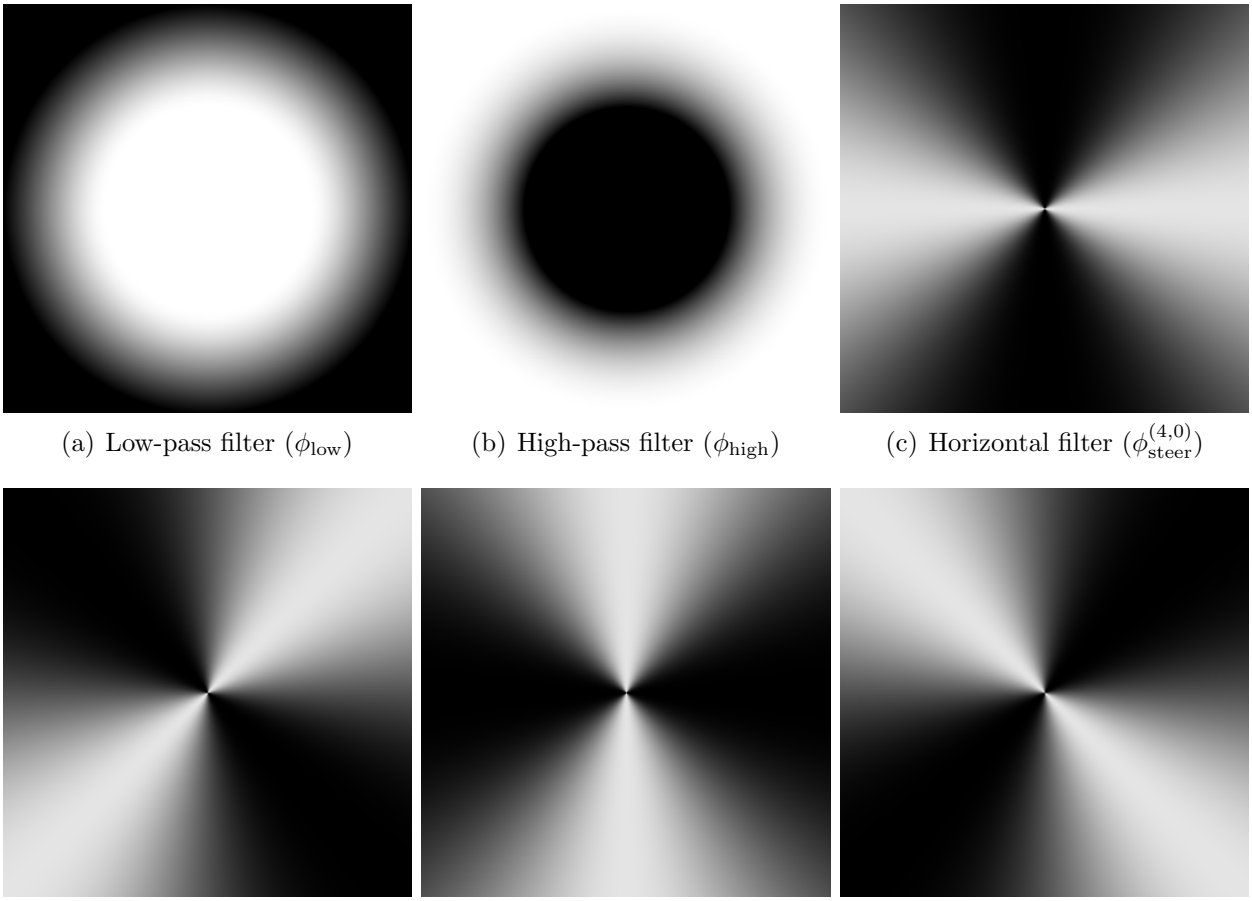
The steerable pyramid is a linear multiscale and multi-orientation image decomposition that has been developed in the 90s by E. Simoncelli and his co-authors [9, 13]. It is designed after the receptive fields found by Hubel and Wiesel [6]. Given an input image, it is obtained by first splitting the image in a high-frequency part and a low-frequency part and then by sequentially applying bandpass oriented filters and downsampling to the low-frequency image. This process results in a sequence of images having different sizes, known as a pyramid. With the exception of two special images in the pyramid, each image corresponds to a certain scale and orientation. The two remaining images are referred to as respectively the high-frequency residual and the low-frequency residual. A fundamental property of the steerable pyramid decomposition is that it can be inverted i.e. the input image can be recovered from its decomposition. For instance, this property is essential in the Heeger & Bergen texture synthesis algorithm [2, 5].

The three building block types (low-pass, high-pass and steered filters) are defined by continuous functions on the Nyquist domain. All of them rely on the transformation between polar and cartesian coordinates given by the following formula:

$$\rho^{-1} : (r, \theta) \in \mathbb{R}^+ \times]-\pi, \pi] \mapsto (r\cos\theta, r\sin\theta) \in \mathbb{R}^2.$$

Definition 22 (Low-pass filter)**.** *The low-pass filter (of the steerable pyramid) is defined by the function $\phi_{low} = l \circ \rho : [-\pi, \pi]^2 \to \mathbb{R}$ where*

$$l(r, \theta) = l(r) = \begin{cases} 1 & \text{if } r \leqslant \frac{\pi}{4}, \\ \cos(\frac{\pi}{2}\log_2(\frac{4r}{\pi})) & \text{if } \frac{\pi}{4} \leqslant r \leqslant \frac{\pi}{2}, \\ 0 & \text{if } r \geqslant \frac{\pi}{2}. \end{cases}$$



(a) Low-pass filter (ϕ_{low}) (b) High-pass filter (ϕ_{high}) (c) Horizontal filter ($\phi_{\text{steer}}^{(4,0)}$)

(d) 1st diagonal filter ($\phi_{\text{steer}}^{(4,1)}$) (e) Vertical filter ($\phi_{\text{steer}}^{(4,2)}$) (f) 2nd diagonal filter ($\phi_{\text{steer}}^{(4,3)}$)

Figure 3: Functions defining the low-pass, high-pass and steered filters of the steerable pyramid in the Nyquist domain (with a number of orientations $Q = 4$). These images are actually the spectral samples corresponding to method 1 for $M = N = 512$ (multiplied by a factor 255 for visualization purposes).

Since ϕ_{low} is null at the boundary of $[-\pi, \pi]^2$ the three methods are equivalent for this filter.

Definition 23 (High-pass filter)**.** *The high-pass filter (of the steerable pyramid) is defined by the function* $\phi_{high} = h \circ \rho : [-\pi, \pi]^2 \to \mathbb{R}$ *where*

$$h(r, \theta) = h(r) = \begin{cases} 0 & \text{if } r \leqslant \frac{\pi}{4}, \\ \cos(\frac{\pi}{2} \log_2(\frac{2r}{\pi})) & \text{if } \frac{\pi}{4} \leqslant r \leqslant \frac{\pi}{2}, \\ 1 & \text{if } r \geqslant \frac{\pi}{2}. \end{cases}$$

Since ϕ_{high} is symmetric at the boundary of $[-\pi, \pi]^2$, method 1 and method 2 are equivalent for this filter.

Definition 24 (Steered filter)**.** *Let Q be the number of orientations of the steerable pyramid and $q \in \{0, \dots, Q-1\}$. The steered filter of orientation q is defined by the function* $\phi_{steer}^{(Q,q)} = s_{Q,q} \circ \rho : [-\pi, \pi]^2 \to \mathbb{R}$ *where*

$$s_{Q,q}(r, \theta) = s_{Q,q}(\theta) = \alpha_Q(\cos(\theta - \frac{\pi q}{Q})^{Q-1} \mathbb{1}_{|\theta - \frac{\pi q}{Q}| \leqslant \frac{\pi}{2}} + \cos(\theta - \frac{\pi(q-Q)}{Q})^{Q-1} \mathbb{1}_{|\theta - \frac{\pi(q-Q)}{Q}| \leqslant \frac{\pi}{2}}),$$

where $\alpha_Q = 2^{Q-1}\frac{(Q-1)!}{\sqrt{Q(2(Q-1))!}}$ *is a normalization constant and* $p!$ *denotes the factorial of the non-negative integer* p*.*

Let Q be the number of orientations of the steerable pyramid. We notice that for the horizontal filter $\phi_{\text{steer}}^{(Q,0)}$ the spectral samples in both conventions are the same i.e. method 1 and method 2 are equivalent.

The reconstruction process (detailed in [2]) is made possible by the relation

$$\phi_{\text{low}}^2 + \phi_{\text{high}}^2 = 1$$

and the normalization constant α_Q which guarantees that $\sum(\phi_{\text{steer}}^{(Q,q)})^2 = 1$. The continuous functions that define the low-pass, high-pass and steered filters in the Nyquist domain for $Q = 4$ are represented in Figure 3.

4.2 Application of the Algorithms

We now apply the three filtering methods to an experimental set of four images and compare the results.

4.2.1 Experimental Set of Images

In order to explain how the set of experimental images is chosen we lead a simple study on the difference between filtered images obtained with two different methods.

Let $\underline{u}$ be a real-valued image and $\phi : [-\pi, \pi]^2 \to \mathbb{C}$ be a filter. Denote by $d(\underline{u}, \phi)$ the maximum absolute difference between filtered images obtained with two different methods and define $\phi_{\max} = \sup_B |\phi|$ where B denotes the boundary of $[-\pi, \pi]^2$. The following definition allows us to obtain a simple upper-bound of $d(\underline{u}, \phi)$.

Definition 25 (Boundary value)**.** *Let* M *and* N *be two positive integers and* $\underline{u}$ *be an image of size* $M \times N$*. The boundary value of* $\underline{u}$ *is denoted* $Bv(\underline{u})$ *and is defined by*

$$Bv(\underline{u}) = \frac{1}{MN} \sum_{(m,n)\in\Gamma_{M,N}} |\mathcal{F}_{M,N}(\underline{u})_{m,n}| \,.$$

The relative boundary value of $\underline{u}$ *is denoted* $Bv_{rel}(\underline{u})$ *and is defined by*

$$Bv_{rel}(\underline{u}) = \frac{MN\,Bv(\underline{u})}{\sum_{(m,n)\in\hat{\Omega}_{M,N}} |\mathcal{F}_{M,N}(\underline{u})_{m,n}|}$$

with the convention "$0/0 = 0$*". It is the ratio between the boundary value and the total value* $\frac{1}{MN}\sum_{(m,n)\in\hat{\Omega}_{M,N}} |\mathcal{F}_{M,N}(\underline{u})_{m,n}|$*.*

A straightforward computation shows that

$$d(\underline{u}, \phi) \leq \phi_{\max} Bv(\underline{u}). \tag{6}$$

The difference between two filtered images obtained by different methods depends on the considered filter and on the frequency content of the input image and more precisely, as stated in Equation (6), on its boundary value. This is why we propose to work with images with diverse frequency contents (from oversampled to textured). The four images used in the experiments are presented in Figure 4.

The modulus of the DFT in logarithmic scale of an image $\underline{u}$ of size $M \times N$ denoted $M(\underline{u})$ is defined by

$$\forall (m,n) \in \hat{\Omega}_{M,N}, \quad M(\underline{u})_{m,n} = \log\left(1 + |\mathcal{F}_{M,N}(\underline{u})_{m,n}|\right).$$

For the visualization (in Figure 4), it is multiplied by the constant factor 15. We notice that the Dice image is almost oversampled (it has a little high-frequency content) whereas the Garden image is textured (it has a large amount of high-frequencies). The Garden and Dice images only have one even size. The Square image is a binary image that has two strong orientations and an important boundary value. The Lena image is an intermediate example of natural image whose relative boundary value is higher than the Garden image one because it is a smaller image with two even sizes.

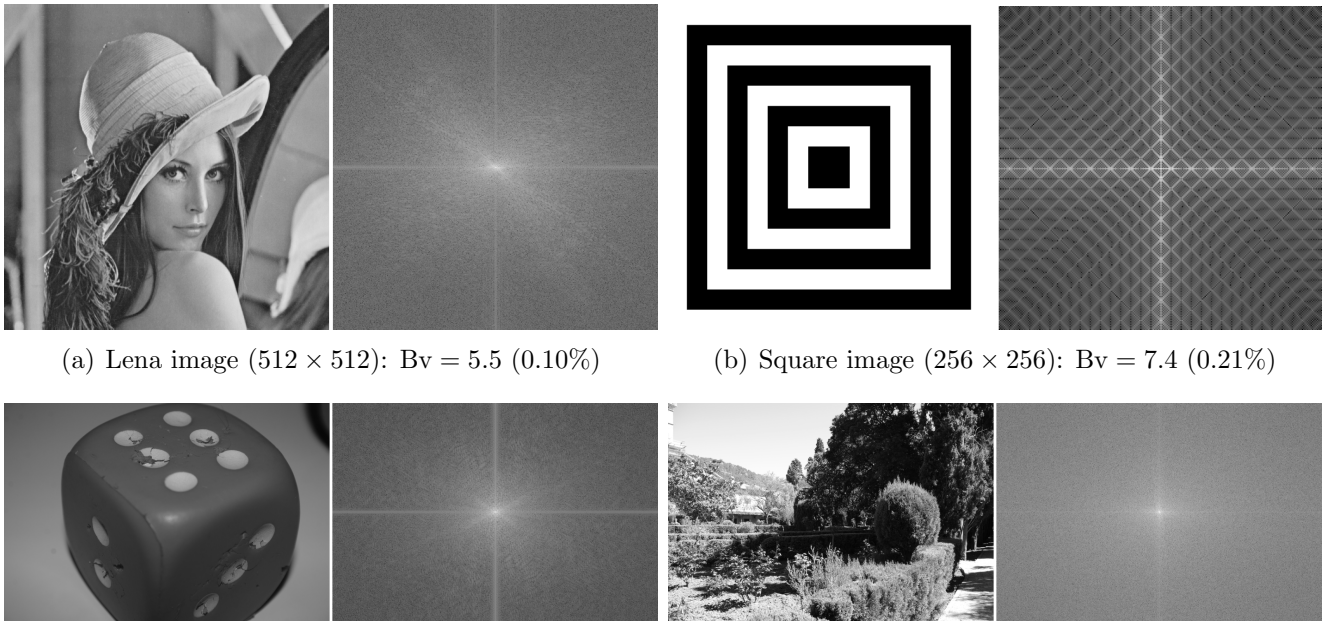
4.2.2 Comparison Procedure of Two Methods

Let us explain in detail how the comparison between the methods is made.

Let ϕ be a filter and $\underline{u}$ be an image of size $M \times N$. Let $(j, j') \in \{1, 2, 3\}^2$ and denote by $\underline{v}_j$ the filtered image obtained by method j. Let R be the range (maximum value minus minimum value) of $\mathcal{R}(\underline{v}_1)$[8]. Then a comparison of $\underline{v}_j$ and $\underline{v}_{j'}$ is done by considering the difference image $\Delta^{j,j'} = |\underline{v}_j - \underline{v}_{j'}|$ and the relative difference image[9] $\Delta^{j,j'}_{\text{rel}} = \Delta^{j,j'}/R$. Before visualization the relative difference image

[8] It's an arbitrary choice.

[9] We use the convention "$0/0 = 0$".



(a) Lena image (512×512): Bv = 5.5 (0.10%)

(b) Square image (256×256): Bv = 7.4 (0.21%)

(c) Dice image (704×469): Bv = 0.68 (0.028%)

(d) Garden image (704×469): Bv = 10 (0.068%)

Figure 4: Set of four images used in the experiments. For each image we plot the image (left) and the modulus of its DFT (right) in logarithmic scale (multiplied by a factor 15). The boundary value is noted Bv and the relative boundary value is written in percentage. We notice that the Dice image is almost oversampled (it has a few high frequencies amount) whereas the Garden image is textured (it has a large amount of high frequencies). Note that the Garden and Dice images only have one even size. The Square image is a binary image that has two strong orientations and an important boundary value. The Lena image is an intermediate example of natural image whose relative boundary value is higher than the Garden image one because it is a smaller image with two even sizes.

is multiplied by the constant factor 10^5 (which is the same whatever are the filter and the input image). We also compute the following values:

1. the maximum difference (maximum value of $\Delta^{j,j'}$ over $\Omega_{M,N}$) noted $d = d^{j,j'}(\underline{u}, \phi)$ and the relative maximum difference (maximum value of $\Delta^{j,j'}_{\text{rel}}$ over $\Omega_{M,N}$) expressed in percentage,

2. the mean difference (mean value of $\Delta^{j,j'}$ over $\Omega_{M,N}$) noted $m_d = m_d^{j,j'}(\underline{u}, \phi)$ and the relative mean difference (mean value of $\Delta^{j,j'}_{\text{rel}}$ over $\Omega_{M,N}$) expressed in percentage.

Note that the differences highly depend on the range of the filtered images, thus the relative values provide more significant information while comparing differences (over the experimental set of images for a given filter or over the filters for a given image).

4.2.3 Results

We apply the comparison procedure to our experimental set of four images and the nine following filters:

- the *sinc* filter (Figure 5),
- the shift filter of parameter $(1/4, 1/4)$ (Figure 6 and Table 2),
- the derivative filter φ_x (Figure 7 and Table 3),
- the Laplacian filter L (Figure 8),
- the Gaussian filter of parameter $\sigma_0 = \frac{\sqrt{\log(2)}}{\pi}$ (Figure 9)[10],
- the low-pass filter (Figure 10),
- the high-pass filter (Figure 11),
- the horizontal filter $\phi^{(4,0)}_{\text{steer}}$ with $Q = 4$ orientations[11] (Figure 12),
- the first diagonal filter $\phi^{(4,1)}_{\text{steer}}$ with $Q = 4$ orientations (Figure 13 and Table 4).

When the three methods provide different filtered images we decompose the presentation of the results into a figure containing filtered images and relative difference images and a table with the maximum difference and mean difference values (and the corresponding relative values).

To display images we use the most common method. The brightness of a pixel is represented by its pixel value which is an integer from 0 to 255 (8-bits images). Let $\underline{u}$ be a real-valued image. The 8-bit image corresponding to $\underline{u}$ is the integral part of $\max(0, \min(255, \underline{u}))$. An affine transformation may have to be applied to filtered images (or their real parts) before obtaining 8-bit images in order to avoid saturation[12] and badly constrasted images. The affine transformation choice (which is arbitrary) depends on the filter and is detailed in the corresponding figure.

The difference is more important when the input image has a strong high-frequency content (Garden and Square images). For 8-bit images the typical range value is $R = 255$, therefore a maximum difference of one grey level corresponds approximately to a relative difference of $1/255 \simeq 0.39\%$. For the proposed set of nine filters and four images, the relative maximum difference values

[10] The value σ_0 is chosen so that $g_{\sigma_0}(\pi, \pi) = 1/2$.
[11] It is the default value used in the Heeger & Bergen texture synthesis algorithm.
[12] For instance saturation occurs for zero-mean images.

are lower than $5/255 \simeq 1.9\%$ and the relative mean difference values are lower than $3/255 \simeq 1.2\%$. It is hardly possible to distinguish with the naked eye a difference between the results of two methods. That is why we only display the real part of the filtered image in the complex convention (method 1). The higher difference values are obtained for the derivative filter and the Laplacian filter. Indeed, it comes from the fact that $|D_x|$ and $|L|$ reach their maxima on the boundary of $[-\pi, \pi]^2$.

4.3 Which Method should I use?

The three methods are equally licit and in general none should be preferred to the others. The results are numerically different but indistinguishable in practice. The maximum difference between the methods of at most 5 grey levels[13] (in a scale from 0 to 255) leads to no visible distinction but determining if it represents a meaningful difference depends on the application. For instance if the goal is to blur an image the user may attach less importance to the differences than when trying to denoise it.

It is left to the user to choose the method. The following remarks may be helpful for making the decision:

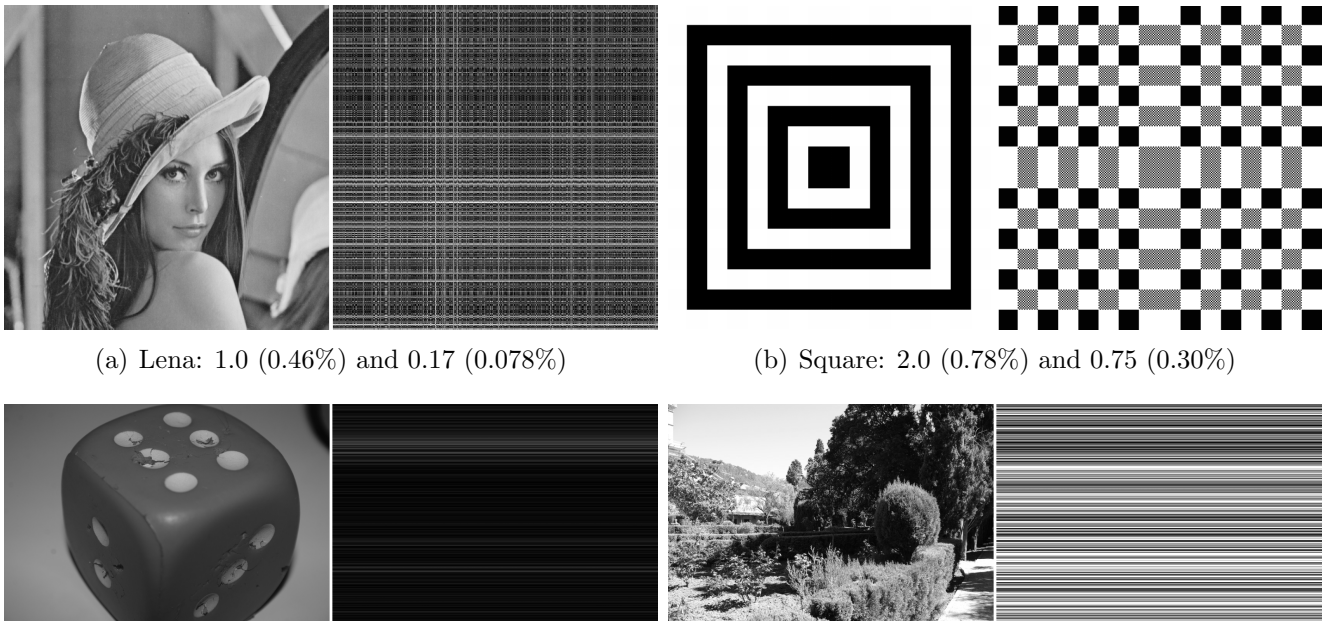
- Method 1 is the easiest one to implement. For simple applications it can be used in first place.
- Suppose the filter ϕ has Hermitian symmetry. Then the filtered images obtained by methods 2 and 3 are real-valued.
- Method 3 destroys high-frequency information but there is no ambiguity in the interpolation. It should not be used in analysis/synthesis scheme based on a reconstruction property.

We recall that when the sizes of the image are odd the three methods are equivalent. If the user has the possibility to modify the size of its input images odd sizes should be considered. However generally the images sizes cannot be changed and sometimes are powers of two (e.g. in a pyramidal decomposition).

		Lena	Square	Dice	Garden
1 vs 2	d	0.77 (0.34%)	1.6 (0.41%)	0.13 (0.067%)	1.4 (0.43%)
	m_d	0.12 (0.052%)	0.55 (0.14%)	0.020 (0.010%)	0.30 (0.094%)
1 vs 3	d	1.1 (0.48%)	2.2 (0.58%)	0.18 (0.093%)	2.0 (0.61%)
	m_d	0.17 (0.075%)	0.77 (0.20%)	0.029 (0.015%)	0.42 (0.13%)
2 vs 3	d	0.77 (0.34%)	1.6 (0.41%)	0.13 (0.067%)	1.4 (0.43%)
	m_d	0.12 (0.053%)	0.55 (0.14%)	0.020 (0.010%)	0.30 (0.094%)

Table 2: Maximum difference d and mean difference m_d for the shift filter of parameter $(1/4, 1/4)$. Relative values are in brackets.

[13] Actually for most of the examples it is less than 1 grey level.



(a) Lena: 1.0 (0.46%) and 0.17 (0.078%)

(b) Square: 2.0 (0.78%) and 0.75 (0.30%)

(c) Dice: 0.17 (0.092%) and 0.029 (0.016%)

(d) Garden: 2.3 (0.92%) and 0.43 (0.17%)

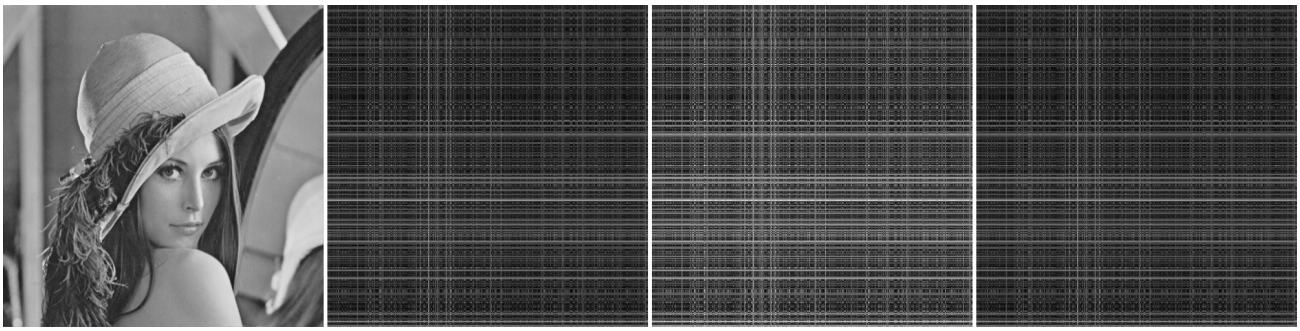
Figure 5: Comparison of the filtering methods for the *sinc* filter. The filtered image $\underline{v}_1$ is the input. For each image we display $\underline{v}_3$ and the relative difference image $\Delta^{1,3}_{\text{rel}}$ (right). The numbers below the images refer respectively to the maximum difference d and the mean difference m_d (with the relative values in brackets).

		Lena	Square	Dice	Garden
1 vs 2	d	1.6 (0.67%)	3.1 (0.84%)	0.53 (0.32%)	7.2 (1.4%)
	m_d	0.44 (0.19%)	1.6 (0.42%)	0.091 (0.055%)	1.3 (0.26%)
1 vs 3	d	1.8 (0.75%)	3.2 (0.86%)	0.53 (0.32%)	7.2 (1.4%)
	m_d	0.50 (0.21%)	1.7 (0.45%)	0.091 (0.055%)	1.3 (0.26%)
2 vs 3	d	0.80 (0.33%)	0.72 (0.19%)	0	0
	m_d	0.16 (0.067%)	0.20 (0.053%)	0	0

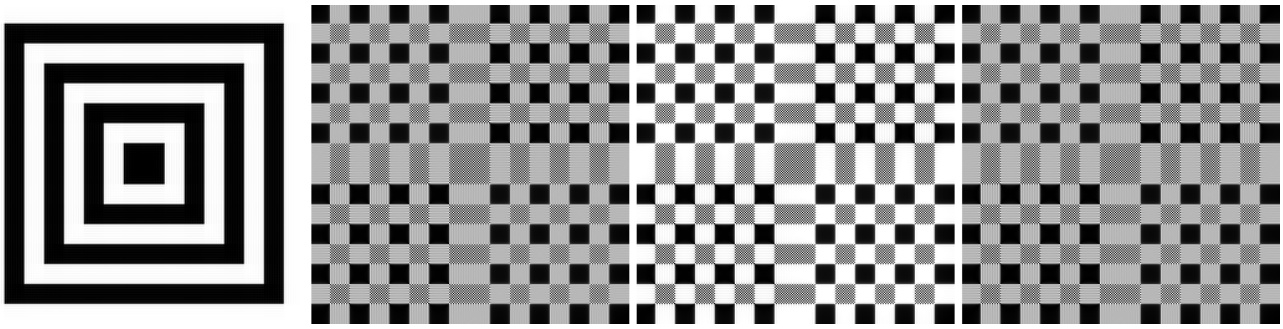
Table 3: Maximum difference d and mean difference m_d for the derivative filter φ_x. Relative values are in brackets.

		Lena	Square	Dice	Garden
1 vs 2	d	0.29 (0.14%)	0.36 (0.10%)	0.058 (0.054%)	0.64 (0.19%)
	m_d	0.046 (0.022%)	0.056 (0.016%)	0.0080 (0.0074%)	0.12 (0.036%)
1 vs 3	d	0.38 (0.18%)	0.79 (0.23%)	0.076 (0.071%)	0.83 (0.25%)
	m_d	0.086 (0.040%)	0.27 (0.077%)	0.015 (0.014%)	0.22 (0.064%)
2 vs 3	d	0.37 (0.17%)	0.70 (0.20%)	0.060 (0.056%)	0.81 (0.24%)
	m_d	0.062 (0.029%)	0.24 (0.070%)	0.011 (0.0097%)	0.16 (0.047%)

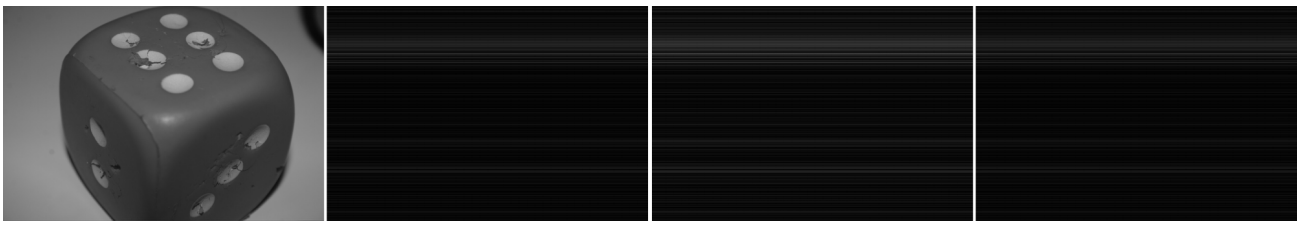
Table 4: Maximum difference d and mean difference m_d for the first diagonal filter $L^{(4,1)}_{\text{steer}}$ with $Q = 4$ orientations. Relative values are in brackets.



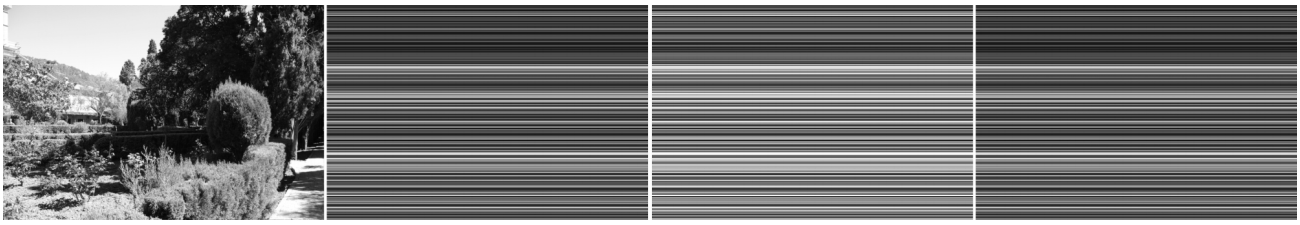
(a) Lena



(b) Square

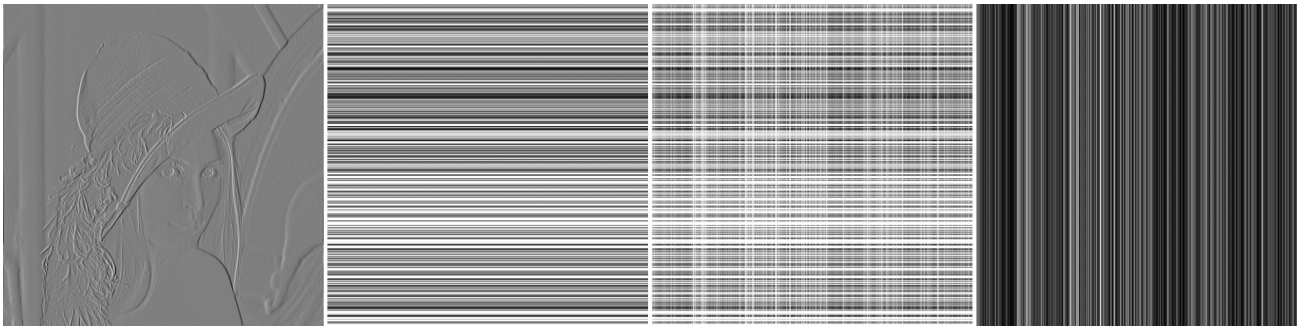


(c) Dice



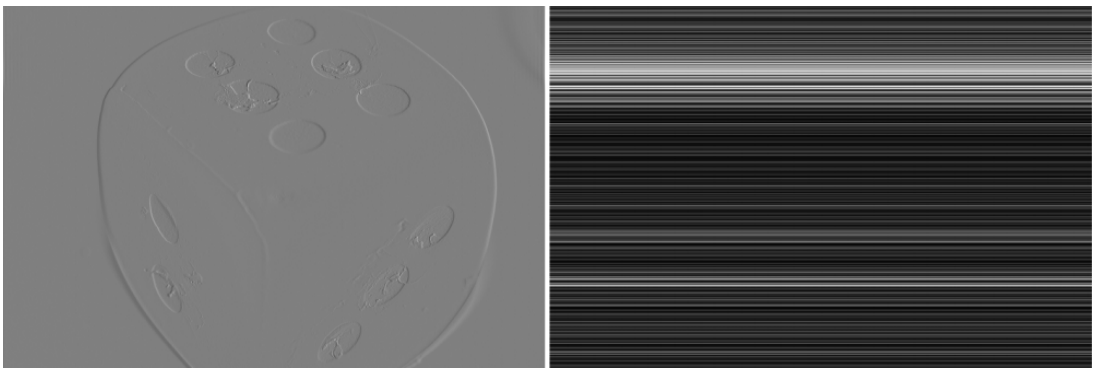
(d) Garden

Figure 6: Comparison of the filtering methods for the shift filter of parameter $(1/4, 1/4)$. From left to right we display for each input image: the real part of $\underline{v}_1$ and the relative difference images $\Delta_{\text{rel}}^{1,2}$, $\Delta_{\text{rel}}^{1,3}$ and $\Delta_{\text{rel}}^{2,3}$.

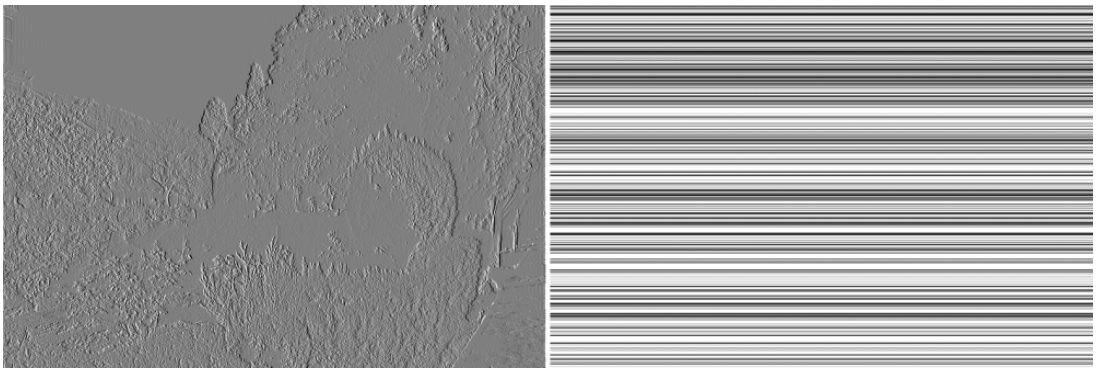


(a) Lena

(b) Square

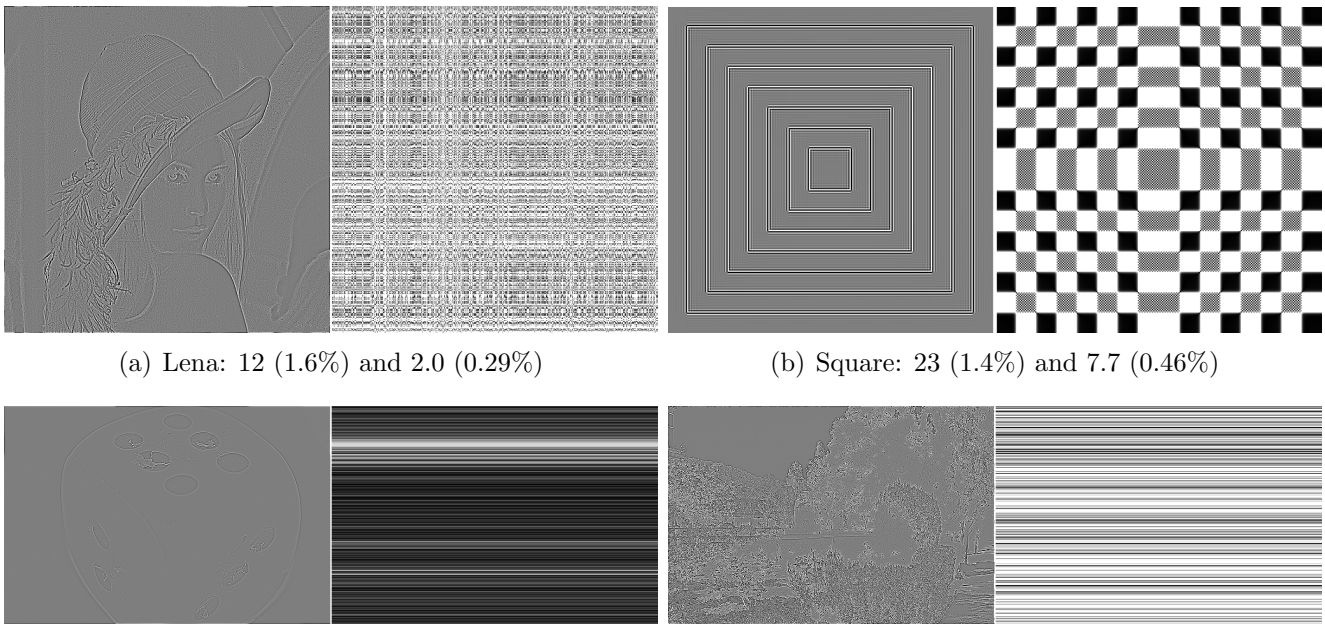


(c) Dice



(d) Garden

Figure 7: Comparison of the filtering methods for the derivative filter φ_x. From left to right we display for the Lena and Square images: the real part of $\underline{v}_1$ and the relative difference images $\Delta_{\text{rel}}^{1,2}$, $\Delta_{\text{rel}}^{1,3}$ and $\Delta_{\text{rel}}^{2,3}$. As method 2 and 3 are equivalent for images of the size of the Dice and Garden image, we only display the real part of $\underline{v}_1$ and $\Delta_{\text{rel}}^{1,2}$ for these images. For visualization purposes we change the mean of $\underline{v}_1$ to 127 (instead of 0). The derivative filter highlights the strong horizontal variations of the input image. We notice a lot of ringing effect (particularly for the Square image).



(a) Lena: 12 (1.6%) and 2.0 (0.29%)

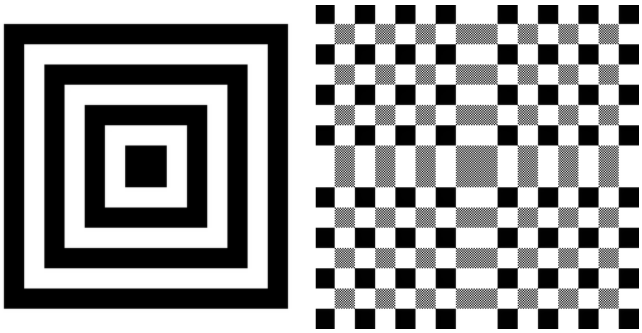
(b) Square: 23 (1.4%) and 7.7 (0.46%)

(c) Dice: 2.1 (0.38%) and 0.35 (0.064%)

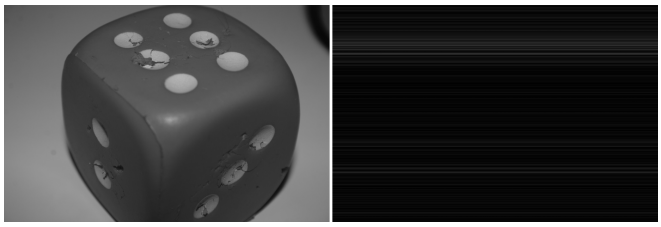
(d) Garden: 26 (1.6%) and 5.2 (0.32%)

Figure 8: Comparison of the filtering methods for the Laplacian filter. For each image we display the real part of $\underline{v}_1$ and the relative difference image $\Delta^{1,3}_{\text{rel}}$ (right). The numbers below the images refer respectively to the maximum difference d and the mean difference m_d (with the relative values in brackets). For visualization purposes we change the mean of $\underline{v}_1$ to 127 (instead of 0). The Laplacian filter highlights the high-frequencies of the input image. We notice a lot of ringing effect (particularly for the Square image).

(a) Lena: 0.68 (0.31%) and 0.11 (0.052%)



(b) Square: 1.4 (0.54%) and 0.53 (0.20%)



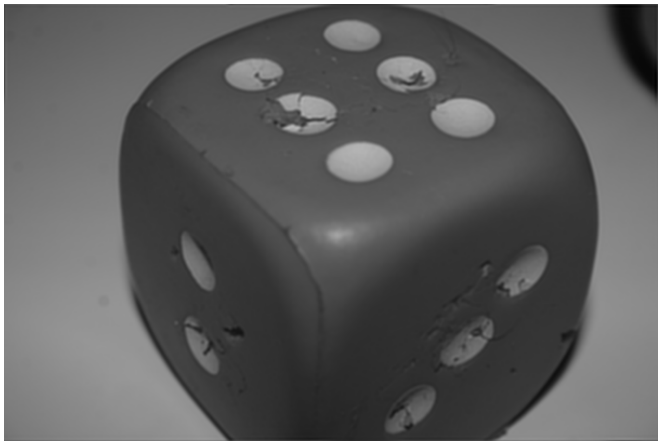
(c) Dice: 0.11 (0.063%) and 0.019 (0.011%)

(d) Garden: 1.6 (0.61%) and 0.28 (0.11%)

Figure 9: Comparison of the filtering methods for the Gaussian filter of standard deviation $\sigma_0 = \frac{\sqrt{\log(2)}}{\pi}$. For each image we display the real part of $\underline{v}_1$ and the relative difference image $\Delta^{1,3}_{\text{rel}}$ (right). The numbers below the images refer respectively to the maximum difference d and the mean difference m_d (with the relative values in brackets). With a higher value of standard deviation the filtered images are more blurry and the differences between the methods decrease.

(a) Lena

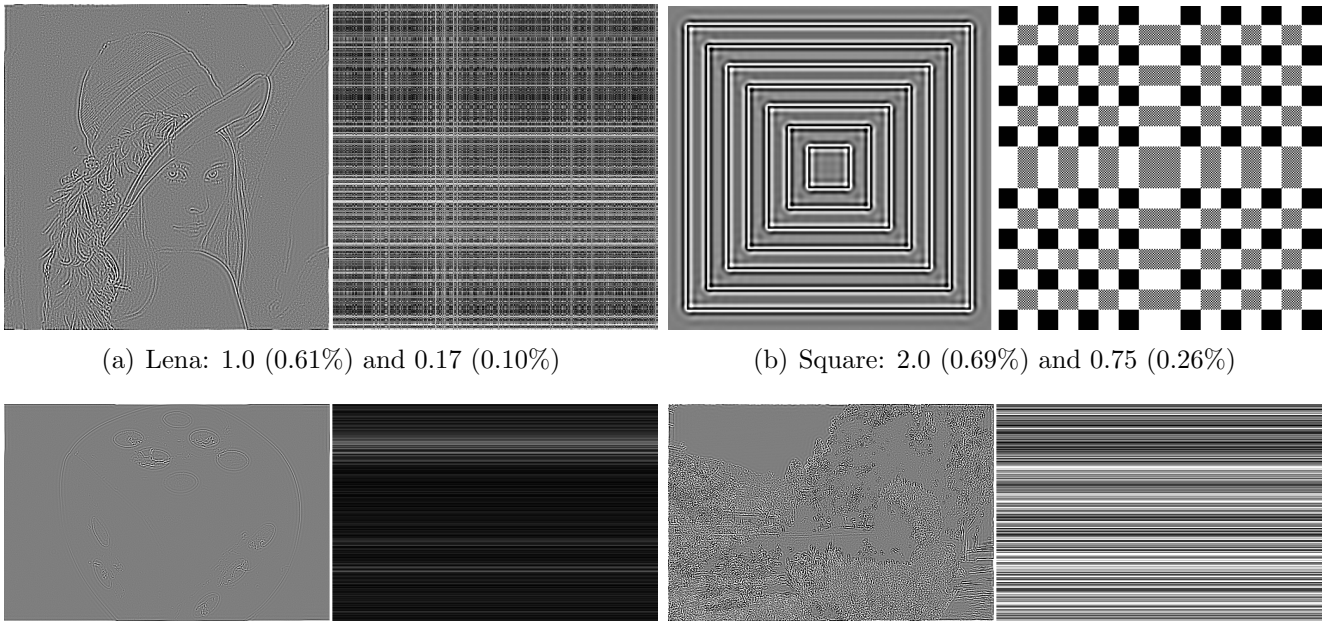
(b) Square



(c) Dice

(d) Garden

Figure 10: Filtering of the images by the low-pass filter. The three methods are equivalent. We notice that as expected the filtered images are blurred versions of the input.



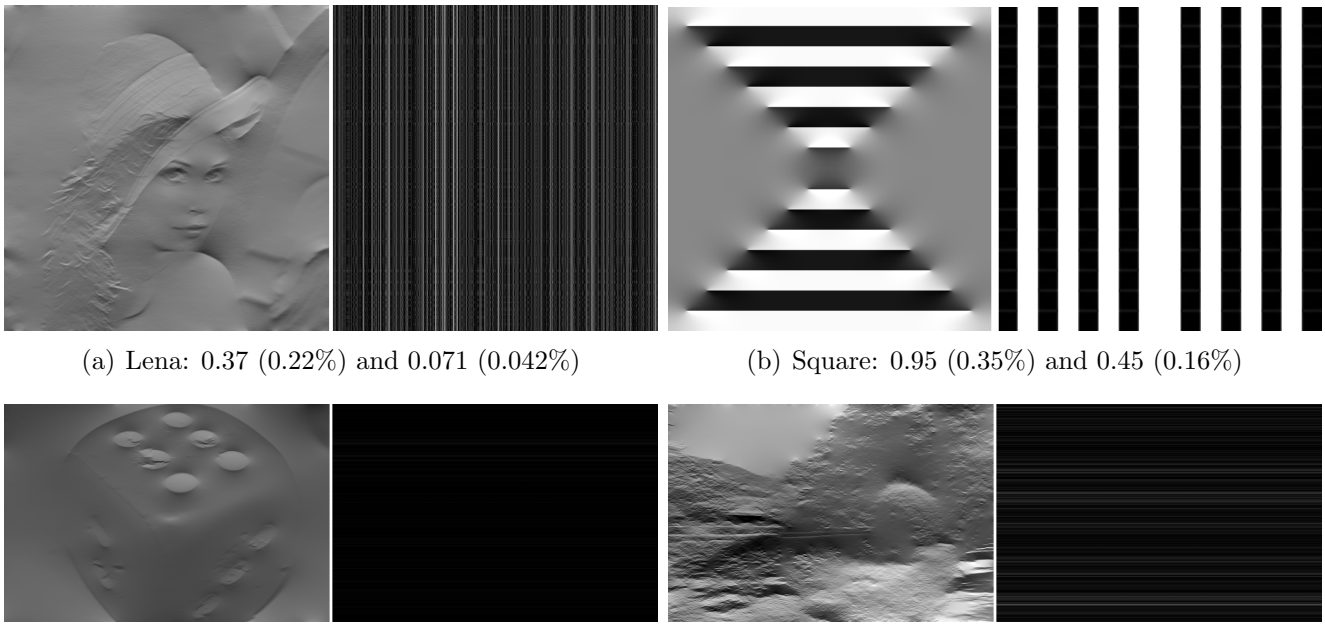
(a) Lena: 1.0 (0.61%) and 0.17 (0.10%)

(b) Square: 2.0 (0.69%) and 0.75 (0.26%)

(c) Dice: 0.17 (0.16%) and 0.030 (0.027%)

(d) Garden: 2.3 (0.84%) and 0.43 (0.16%)

Figure 11: Comparison of the filtering methods for the high-pass filter. For each image we display the real part of $\underline{v}_1$ and the relative difference image $\Delta^{1,3}_{\text{rel}}$ (right). The numbers below the images refer respectively to the maximum difference d and the mean difference m_d (with the relative values in brackets). For visualization purposes we multiply by a factor 5 the image $\underline{v}_1$ and then set the mean to 127 (instead of 0). The high-pass filter highlights the strong variations of the input image.



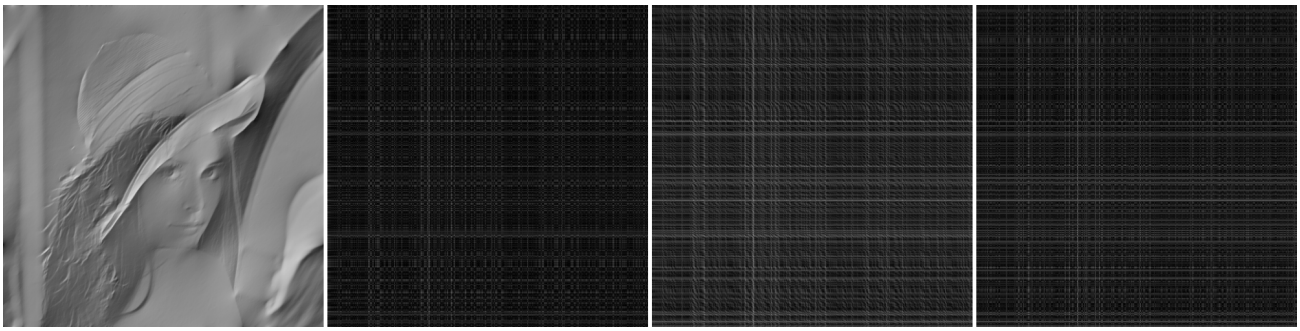
(a) Lena: 0.37 (0.22%) and 0.071 (0.042%)

(b) Square: 0.95 (0.35%) and 0.45 (0.16%)

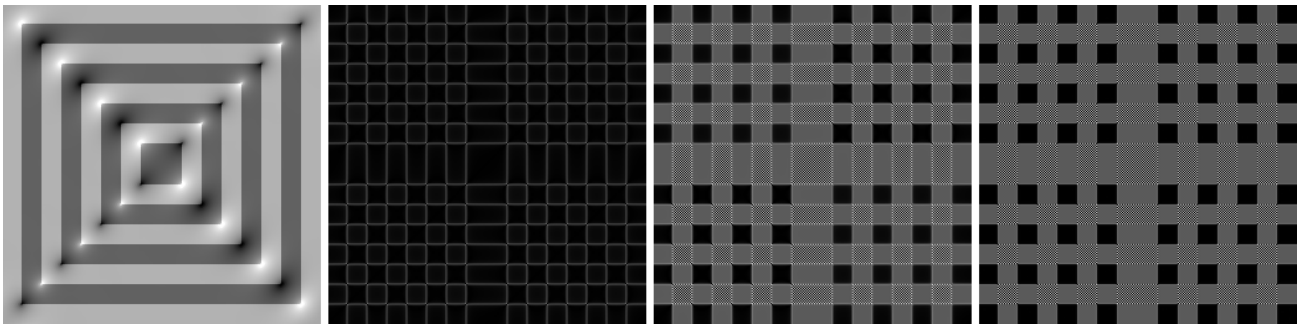
(c) Dice: 0.022 (0.016%) and 0.0031 (0.0023%)

(d) Garden: 0.22 (0.070%) and 0.047 (0.015%)

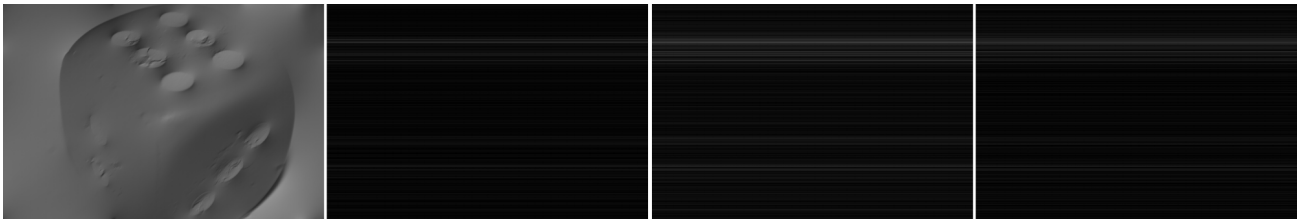
Figure 12: Comparison of the filtering methods for the horizontal filter $\phi_{\text{steer}}^{(4,0)}$ with $Q = 4$ orientations. For each image we display the real part of $\underline{v}_1$ and the relative difference image $\Delta_{\text{rel}}^{1,3}$ (right). The numbers below the images refer respectively to the maximum d and the mean difference m_d (with the relative values in brackets). The horizontal filter highlights the strong horizontal variations of the input image.



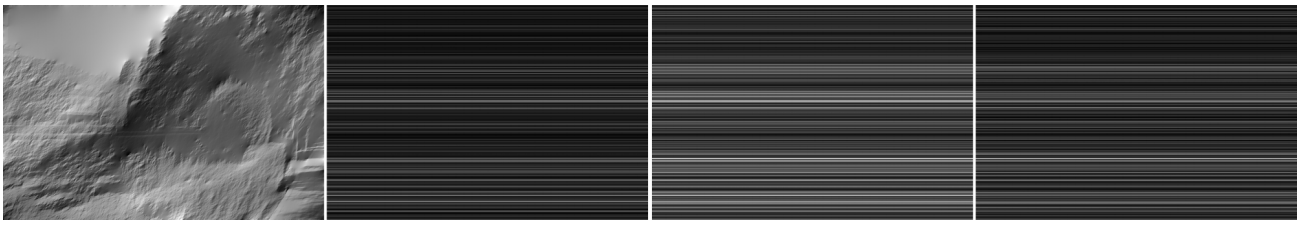
(a) Lena



(b) Square



(c) Dice



(d) Garden

Figure 13: Comparison of the filtering methods for the first diagonal filter $\phi^{(4,1)}_{\text{steer}}$ with $Q = 4$ orientations. From left to right we display for each input image: the real part of $\underline{v}_1$ and the relative difference images $\Delta^{1,2}_{\text{rel}}$, $\Delta^{1,3}_{\text{rel}}$ and $\Delta^{2,3}_{\text{rel}}$. As expected the first diagonal filter highlights the strong variations in the first diagonal orientation.

5 Conclusion

In this paper we give a clear interpretation, as a continuous convolution, of the filtering of a discrete real-valued image by a filter specified by a continuous function in the Nyquist domain $[-\pi, \pi]^2$. The filtering is interpreted as the convolution of a distribution, standing for the filter, with a trigonometric polynomial interpolator of the image. Two plausible interpolations and two choices of distribution are considered. They lead to three equally licit algorithms. All of them can be seen as method variants of the same standard filtering algorithm in the DFT domain. By an application to several fundamental filters we show that the method differences, which come from the boundary DFT coefficients, are not visible to the naked eye in practice. For 8-bit images, the difference order of magnitude is in general of one grey level but may be higher for filters with large values on the boundary of the Nyquist domain. In general none method should be preferred to the others and the choice depends on the desired properties of the filtering. For instance, in our implementation of the Heeger-Bergen pyramid-based texture synthesis algorithm provided in [2], we use method 1 in order to guarantee the exact reconstruction property.

Acknowledgements

This work was partly founded by BPIFrance and Région Ile de France, in the framework of the FUI 18 Plein Phare project, the Office of Naval research (ONR grant N00014-14-1-0023), and ANR-DGA project ANR-12-ASTR-0035.
The authors would like to thank Loïc Simon for his very valuable corrections and comments.

Image Credits

Standard test image

Provided by the authors, CC-BY

Miguel Colom, CC-BY

Antoni Buades, CC-BY

References

[1] R. Abergel and L. Moisan, *The Shannon Total Variation*. Preprint MAP5, https://hal.archives-ouvertes.fr/hal-01349516, 2016.

[2] T. Briand, J. Vacher, B. Galerne, and J. Rabin, *The Heeger & Bergen Pyramid Based Texture Synthesis Algorithm*, Image Processing On Line, 4 (2014), pp. 276–299. http://dx.doi.org/10.5201/ipol.2014.79.

[3] M. Frigo and S.G. Johnson, *The design and implementation of FFTW3*, Proceedings of the IEEE, 93 (2005), pp. 216–231. http://dx.doi.org/10.1109/JPROC.2004.840301.

[4] 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.

[5] D.J. Heeger and J.R. Bergen, *Pyramid-based texture analysis/synthesis*, in Proceedings of the 22nd Annual Conference On Computer Graphics And Interactive Techniques, ACM, 1995, pp. 229–238. http://dx.doi.org/10.1145/218380.218446.

[6] D.H. Hubel and T.N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, The Journal of Physiology, 160 (1962), p. 106. http://dx.doi.org/10.1113/jphysiol.1962.sp006837.

[7] Y. Katznelson, *An introduction to harmonic analysis*, chapter 1. Cambridge University Press, 2004. ISBN 0521543592.

[8] A. Papandreou-Suppappola, *Applications in time-frequency signal processing*, CRC press, 2002. ISBN 1420042467.

[9] J. Portilla and E.P. Simoncelli, *A parametric texture model based on joint statistics of complex wavelet coefficients*, International Journal of Computer Vision, 40 (2000), pp. 49–70. http://dx.doi.org/10.1023/A:1026553619983.

[10] M. Reed and B. Simon, *Methods of Modern Mathematical Physics: Vol.: 1.: Functional Analysis*, Academic press, 1972. ISBN 0125850506.

[11] I. Rey Otero and M. Delbracio, *Computing an Exact Gaussian Scale-Space*, Image Processing On Line, 6 (2016), pp. 8–26. http://dx.doi.org/10.5201/ipol.2016.117.

[12] L. Schwartz, *Théorie des distributions*, Actualités Scientifiques et Industrielles, Institut de Mathématique, Université de Strasbourg, 1 (1966), p. 2. http://dx.doi.org/10.1090/S0002-9904-1952-09555-0.

[13] E.P. Simoncelli and W.T. Freeman, *The steerable pyramid: A flexible architecture for multi-scale derivative computation*, in Proceedings of the International Conference on Image Processing, vol. 3, IEEE, 1995, pp. 444–447. http://dx.doi.org/10.1109/ICIP.1995.537667.

[14] R.S. Strichartz, *A guide to distribution theory and Fourier transforms*, World Scientific, 2003. ISBN 0849382734.

Errata

In the original paper published on 7 November 2016, Proposition 2 was wrong for M and N even. A correct formulation was actually proven in [1] (Equation (23)). The authors would like to thank Rémy Abergel and Lionel Moisan for reporting the error. The trigonometric polynomial corresponding to the coefficients written in the proposition is still a real-valued trigonometric polynomial interpolator of $\underline{u}$ and it should be preferred to $\mathcal{R}(P_{\underline{u}}^{(c)})$. Thus the changes consist essentially in modifying the definition of $P_{\underline{u}}^{(r)}$ in Section 2.2. The three methods can be naturally extended to complex-valued images.

In the new version, compared to the version published on 7 November 2016, the exact changes are the following:

- Page 187 paragraph 1. The sentence "By taking its real part, we build another interpolator." is replaced by "Thus, we build another interpolator that is real-valued."
- Page 187 Definition 8. The definition of $P_{\underline{u}}^{(r)}$ is modified so that its coefficients are the ones of the previous version of Proposition 2.
- Page 187 paragraph 2. Removed.
- Page 187 Proposition 2. Proposition 2 is modified.
- Page 187 Proof of Proposition 2. We add the sentence "If M and N are even, we have $\mathcal{F}_{M,N}(\underline{u})_{-\frac{M}{2},-\frac{N}{2}} \in \mathbb{R}$ and we conclude using the relation $\mathcal{R}(e^{-i\pi(x+y)}) + \sin(\pi x)\sin(\pi y) = \frac{1}{4}\left(e^{i\pi(x+y)} + e^{-i\pi(x+y)} + e^{i\pi(x-y)} + e^{i\pi(-x+y)}\right)$."
- Page 187 paragraph 3. The sentence "Taking the real part of $P_{\underline{u}}^{(c)}$ boils down to correcting the eventual asymmetry of the DFT coefficients." is moved to the end of the paragraph and is now followed by the sentences "When M and N are even $P_{\underline{u}}^{(r)}$ is preferred to $\mathcal{R}(P_{\underline{u}}^{(c)})$ because it has the same coefficients at the four corners of $\hat{\Omega}^s_{M,N}$. Proposition 2 is a reformulation of Equation (23) in [1]."
- Page 192 Proof of Proposition 4. In the first sentence of the real convention case, "Proposition 2" is replaced by "Definition 8".
- Page 193 paragraph 2. The sentence "Note that only methods 1 and 3 can be naturally extended for complex-valued images while on the contrary the trigonometric polynomial used in method 2 is not interpolating when dealing with non-real images." is replaced by "Note that the three methods can be naturally extended to complex-valued images.".
- Page 200 Section 4.3. We removed the third item "Methods 1 and 3 are applicable to complex-valued images while method 2 is not.".
- Page 210 Bibliography. The reference to [1] is added.