A fast wavelet algorithm for image deblurring

D. L. Donoho * M.E Raimondo †

23 August 2004

Abstract

We present a non-linear fully adaptive wavelet algorithm which can recover a blurred image $(n \times n)$ observed in white noise with $O(n^2(\log n)^2)$-steps. Our method exploits both the natural representation of the convolution operator in the Fourier domain and the typical characterisation of Besov classes in the wavelet domain. Particular feature of our method includes "cycle-spinning" band-limited wavelet approximations over all circulant shifts. The speed and the accuracy of the algorithm is illustrated with numerical examples of image deblurring. All figures presented in this paper are reproducible using the `WaveD` software package.

Contents

*Dept. Statistics, Stanford University, Stanford, U.S.A.

†Dept. Maths & Stats, The University of Sydney, Sydney, AUSTRALIA. `mailto:marcr@maths.usyd.edu.au`

1 Introduction

We begin with a function f defined on the unit square $T = [0,1]^2$, this function represents our image. Let g be a blurring kernel also defined on T and let

$$h(x) := f \star g(x) = \int_T f(x_1 - u_1, x_2 - u_2) g(u_1, u_2) du_1 du_2 \,, \tag{1}$$

represent our blurred image. Suppose we observe h with some additive noise

$$Y_n(dx) = h(x)dx + \sigma n^{-1} W(dx), \quad x \in T = [0,1] \times [0,1], \tag{2}$$

where σ is a positive constant and $W(.)$ is a 2D Wiener process. This model is illustrated in Figure 3(b) using the well-known '*Camera*', Figure 3(a). Our goal is to recover the original image f from the noisy blurred observations (2) using a wavelet thresholding estimator. The model (2) is an idealised version of the motion blur model for digital images, see [1] for examples and applications.

In this paper we extend the WaveD paradigm of Johnstone, Kerkyacharian, Picard, and Raimondo [5] to the 2-dimensional setting and propose a non-linear fully adaptive wavelet estimator of the original image f. We show that the Translation-Invariant WaveD algorithm of Donoho and Raimondo [3] extend well to the 2-dimensional setting according to each orientation: vertical, horizontal and diagonal. This allows fast computation to be performed in the Fourier domain where the convolution operator (1) factorises nicely. Our paper is organised as follows: in section 2 we review basic materials on 2-dimensional band-limited wavelet basis and translation invariant wavelet transforms; in section 3, we present the 2-dimensional WaveD estimator; section 4 is concerned with examples and numerical properties as well as a comparison with the recent ForWarD algorithm of [9]. A brief description of the 2-dimensional WaveD algorithm is given in section 5.

2 Preliminaries

2.1 Two-dimensional wavelet basis

Two-dimensional wavelet basis are constructed by tensor products of one-dimensional wavelet orthogonal systems. We recall the following theorem taken from [7].

Theorem 1. *Let ϕ be a scaling function and ψ be the corresponding wavelet generating a wavelet orthonormal basis of $L^2(R)$. We define three wavelets:*

$$\psi^1(x) = \phi(x_1)\psi(x_2) \ , \ \psi^2(x) = \psi(x_1)\phi(x_2) \ , \ \psi^3(x) = \psi(x_1)\psi(x_2), \tag{3}$$

and denote for $1 \leq m \leq 3, k = (k_1, k_2), x = (x_1, x_2)$ and $\kappa = (j, k)$

$$\psi^m_\kappa(x) := \psi^m_{j,k}(x) = 2^j \psi^m \Big(2^j x_1 - k_1, 2^j x_2 - k_2\Big). \tag{4}$$

The wavelet family

$$\Big\{\psi^1_\kappa, \psi^2_\kappa, \psi^3_\kappa\Big\}_{(\kappa \in Z^3)} \tag{5}$$

is an orthonormal basis of $L^2(R^2)$.

We illustrate this theorem using the Meyer wavelet, [8]. A Meyer wavelet ψ is a function whose Fourier transform $F(\psi) := \widehat{\psi}$ is smooth, see formula for the construction of $\hat{\psi}$ p.247 in [7]. In practice, we use a polynomial function to define the so-called Meyer window, see [7] p.248. Note that the Meyer wavelet is band limited and we have $Supp(\widehat{\phi}(w)) = \{w : |w| \in [0, 4\pi/3]\}$ and $Supp(\widehat{\psi}(w)) = \{w : |w| \in [2\pi/3, 8\pi/3]\}$.

Applying Fourier transforms in (3) we construct two-dimensional Meyer wavelets in the Fourier domain,

$$\widehat{\psi}^1(w_1, w_2) = \widehat{\phi}(w_1)\widehat{\psi}(w_2) \ , \ \widehat{\psi}^2(w_1, w_2) = \widehat{\psi}(w_1)\widehat{\phi}(w_2) \ , \ \widehat{\psi}^3(w_1, w_2) = \widehat{\psi}(w_1)\widehat{\psi}(w_2) \tag{6}$$

and $\widehat{\phi}^1(w_1, w_2) = \widehat{\phi}(w_1)\widehat{\phi}(w_2)$ for the two-dimensional scaling function. This process is illustrated in Figure 1.

The three wavelets defined in Theorem 1 will reveal image details at different frequencies and orientations. ψ^1_κ will extract vertical details and ψ^2_κ will extract horizontal details whereas ψ^3_κ will extract diagonal details. This orientation property is visible in the Fourier domain, for example $|\psi^1_\kappa(w_1, w_2)|$ will be large at low horizontal frequencies and high vertical frequencies, see Figure 1(b).

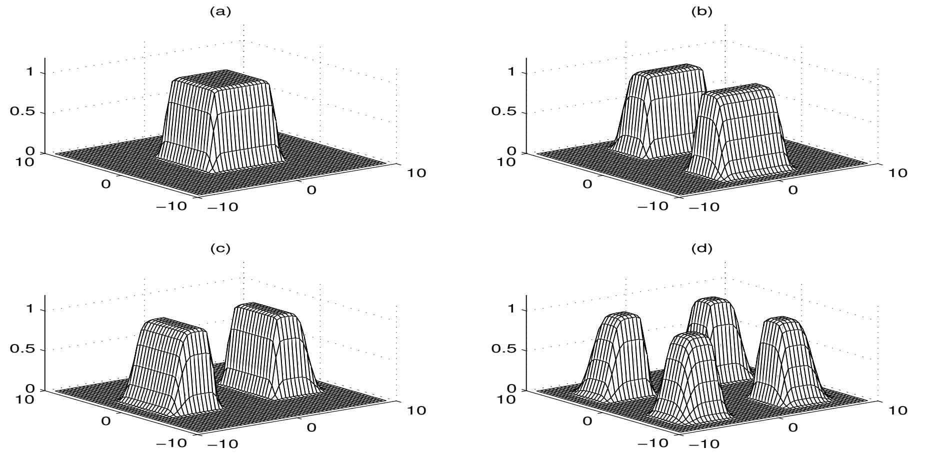

Figure 1: (a) $|\widehat{\phi}(w_1, w_2)|$; (b) $|\widehat{\psi}^1(w_1, w_2)|$; (c) $|\widehat{\psi}^2(w_1, w_2)|$;(d) $|\widehat{\psi}^3(w_1, w_2)|$.

2.2 Translation Invariant wavelet transforms

In this section we recall the formal definition of cycle-spinning a generic (two-dimensional) wavelet thresholding estimator and we define the Translation Invariant (TI) WaveD estimator. We refer to [2] and references therein for a more general discussion. A wavelet thresholding estimator can be described as follows:

$$\begin{array}{cccc} \text{Space domain} & & \text{Wavelet domain} & \\ Y(x) & \longrightarrow^{WT} & (\beta_\kappa) & \\ & & \downarrow_\eta & \textit{thresholding} \\ \tilde{f}(x) & \longleftarrow^{IWT} & \eta(\beta_\kappa) & \end{array}$$

where WT, IWT denotes a generic wavelet transform and its inverse. For any space shift $h = (h_1, h_2)$, we let $T_h f(x) = f(x + h) := f(x_1 + h_1, x_2 + h_2)$ denote the shift operator. For an arbitrary shift h we define one cycle-spin of the estimator $\tilde{f}(t)$ as follows

$$\begin{array}{ccccc} & \text{Space domain} & & \text{Wavelet domain} & \\ & Y(x) & & & \\ \textit{shift} & \downarrow & & & \\ & T_h Y(x) & \longrightarrow^{WT} & (\beta^h_\kappa) & \\ & & & \downarrow_\eta & \textit{thresholding} \\ & \tilde{f}_h(t) & \longleftarrow^{IWT} & \eta(\beta^h_\kappa) & \\ \textit{unshift} & \downarrow & & & \\ & T_{-h}(f_h)(x) & & & \end{array}$$

The idea behind cycle-spinning is to correct possible mis-alignments between features in the image and features in the basis. Of course, if the image is unknown there is little hope to find the best value of h. A well established approach to solve this problem is to apply a range of shifts and average over all several results [2]. Let H be a set of shifts, cycle-spinning the WaveD estimator over H yields a new estimator:

$$\tilde{f}_H = Ave_{h \in H} T_{-h}(\hat{f}_h) = \frac{1}{|H|} \sum_{h \in H} T_{-h}(\hat{f}_h)\,, \tag{7}$$

where $|H|$ denotes the cardinality of H. Yet again the problem of choosing the best H may arise. In fact, it is clear that the bigger is H the better since we take into account more possible mis-alignments. Assuming that the observed data $\mathtt{y} = (\mathtt{y}_{(1,1)}, ...\mathtt{y}_{(i,j)}, ..., \mathtt{y}_{(n,n)})$ are sampled at a discrete set of equally spaced points $y_{(i,j)} = Y(i/n, j/n), i, j = 1, ..., n$; we define $H_1 = \{1/n, 2/n, ..., n-1/n, 1\}$ and $H_2 = H_1 \times H_1$ to be set of all possible circulant shifts. The Translation Invariant WaveD estimator is defined by

$$\tilde{f}_{TI} = Ave_{h \in H_2} T_{-h}(\hat{f}_h). \tag{8}$$

We refer to section 4 of [2] for a discussion of the improvement provided by the cycle spinning process. This includes for example improvement in approximation, suppression of Gibbs phenomenon as well as improvement in de-noising.

2.3 Periodised Meyer wavelets

We start with a 1-dimensional Meyer orthogonal system ϕ, ψ and let

$$\Psi(x) = \sum_{k \in Z} \psi(x+k), \;\; \Psi_{j,k}(x) = 2^{j/2}\Psi(2^j x - k) \tag{9}$$

with a similar formula for the periodised scaling function Φ in terms of the Meyer scaling function ϕ. It is easy to check that (Φ, Ψ) are periodic and yields an orthonormal basis of $\mathrm{L}^2(T)$, see e.g. [4]. The next step is to apply Theorem 1 to the periodised pair (Φ, Ψ) which gives a two-dimensional periodised wavelet basis of $\mathrm{L}^2(T)$:

$$\left\{\Phi_\kappa, \Psi^1_\kappa, \Psi^2_\kappa, \Psi^3_\kappa\right\}_{(\kappa \in Z^3)}. \tag{10}$$

3 The two-dimensional WaveD-method

3.1 Fourier transforms

For $l = (l_1, l_2) \in Z^2$ and $x = (x_1, x_2) \in T$, let $l'x = l_1.x_1 + l_2.x_2$ and $e_l(x) = \exp(2\pi i l'x), l \in Z^2$ denote the Fourier basis of $L^2(T)$. For periodic functions (images) of $L^2(T)$, the Fourier coefficients are given by

$$f_l := \langle f, e_l \rangle = \int_0^1 \int_0^1 f(x_1, x_2)\, e_l(x_1, x_2) dx_1 dx_2. \tag{11}$$

For blurred images $h = f \star g$, we have the nice formula

$$h_l := \langle h, e_l \rangle = \langle f \star g, e_l \rangle = \langle f, e_l \rangle . \langle g, e_l \rangle = f_l . g_l, \tag{12}$$

for noisy images $Y_n(x)$ and Wiener process $W(x)$ the Fourier coefficients are defined by

$$y_l = \langle Y_n, e_l \rangle := \int_0^1 \int_0^1 e_l(x_1, x_2) dY_n(x_1, x_2), \tag{13}$$

$$z_l = \langle W, e_l \rangle := \int_0^1 \int_0^1 e_l(x_1, x_2) dW(x_1, x_2). \tag{14}$$

Applying the two-dimensional Fourier transform to the data (2) yields the sequence model

$$y_l = h_l + \sigma n^{-1} z_l\,, \quad l \in Z^2. \tag{15}$$

The WaveD paradigm of [5] combines Fourier and wavelet transforms. For periodised wavelets (10), the Fourier coefficients are given by

$$\Psi_l^{m,\kappa} := \langle \Psi_\kappa, e_l \rangle = \int_0^1 \int_0^1 \Psi_\kappa^m(x_1, x_2)\, e_l(x_1, x_2) dx_1 dx_2, \tag{16}$$

it is straightforward to check that the latter Fourier coefficients can be computed from the Fourier transform of the original wavelet (evaluated on a dyadic grid). For example,

$$\Psi_l^{m,j,0,0} = \langle \Psi_{j,0,0}^m, e_l \rangle = 2^{-j} \widehat{\psi}(l/(2^j.2\pi)). \tag{17}$$

Hence, using the construction formula for $\widehat{\psi}$, see e.g. [7] p.247, we can compute the Fourier coefficients of $\Psi_{j,0,0}^m$ at any resolution level j. This is illustrated on Figure 2 with $j = 3$ and $m = 2$.

Remark 1. The Fourier coefficients whose modulus are depicted on Figure 2(b) play a key role in the WaveD algorithm. In fact, at resolution level j we only need the Fourier coefficients of $\Psi_{j,0,0}^m$ to compute both the wavelet transform and its inverse, see proposition 1 section 5.

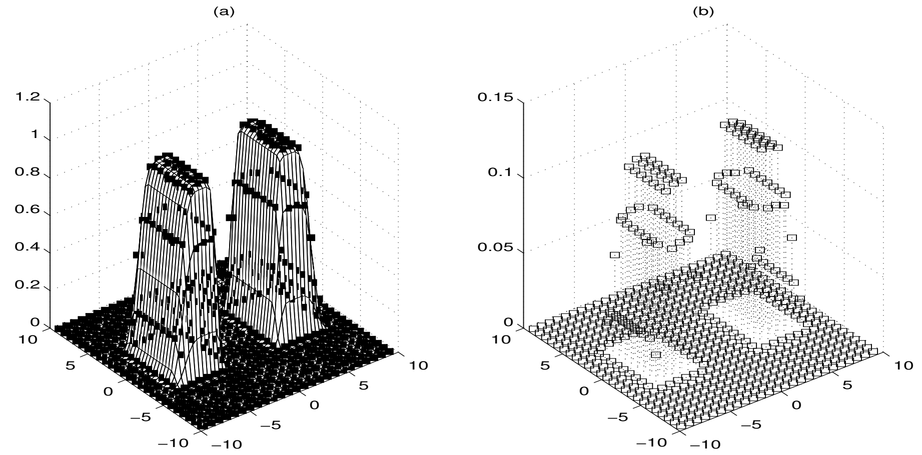

Figure 2: (a) The continuous surface depicts $\omega \to |\hat{\psi}^2(\omega)|$ and points are sampling at $\omega_l = l/(2^j.2\pi), l \in Z$, with resolution level $j = 3$; (b) Modulus of Fourier coefficients $\Psi_l^{m,j,0,0} = \hat{\psi}^2(\omega_l)/8$, with $m = 2$ and $j = 3$, see (17).

3.2 The WaveD paradigm

For periodic images in $L^2(T)$, the wavelet coefficients are given by

$$\beta_\kappa^m := \langle f, \Psi_\kappa^m \rangle = \int_0^1 \int_0^1 f(x_1, x_2)\, \Psi_\kappa^m(x_1, x_2) dx_1 dx_2. \tag{18}$$

For blurred images $h = f \star g$, we can compute the wavelet transform via the Plancherel formula:

$$\langle h, \Psi_\kappa^m \rangle = \sum_l h_l . \Psi_l^{m,\kappa}, \tag{19}$$

the WaveD paradigm stipulates that we can perform wavelet transform and deconvolution simultaneously since by (12) and Plancherel's equality,

$$\mathrm{WaveD}(h, g, \Psi_\kappa^m) := \sum_l \left(\frac{h_l}{g_l}\right) . \Psi_l^{m,\kappa} = \sum_l f_l . \Psi_l^{m,\kappa} = \beta_\kappa^m. \tag{20}$$

In the case of noisy images, we define the (unbiased) Waved estimator by

$$\tilde{\beta}_\kappa^m = \mathrm{WaveD}(Y, g, \Psi_\kappa^m) := \sum_{l \in C_j^m} \left(\frac{y_l}{g_l}\right) \bar{\Psi}_l^{m,\kappa}, \tag{21}$$

where $C_j^m = \{l : \Psi_l^{m,\kappa} \neq 0\}$. In the light of the above WaveD formula, we see that band-limited wavelets (e.g. Meyer) are very attractive since only a finite number of terms is used in the sum (21).

3.3 The WaveD estimator

After performing deconvolution and wavelet transform, simultaneously, according to (21), we define the WaveD estimator by

$$\tilde{f} = \sum_{\kappa \in I_0} \tilde{\alpha}_\kappa I_{\{|\tilde{\alpha}_\kappa| \geq \lambda_{j_0}\}} \Phi_\kappa + \sum_{m=1}^{3} \sum_{\kappa \in I_1} \tilde{\beta}_\kappa^m I_{\{|\tilde{\beta}_\kappa^m| \geq \lambda_j^m\}} \Psi_\kappa^m . \tag{22}$$

where I_0, I_1 are set of indices; $I_0 = \{(j_0, k_1, k_2) : 0 \leq k_i \leq 2^{j_0} - 1, i = 1, 2\}$ corresponds to a coarse resolution level j_0 and $I_1 = \{(j, k_1, k_2) : 0 \leq k_i \leq 2^j - 1,\, i = 1, 2,\, j_0 \leq j \leq j_1\}$ indexes details up to a fine resolution level j_1 . To specify the thresholds λ_j^m, we extend the definition of [5] and set

$$\tau_j^m = \Big(|C_j^m|^{-1} \sum_{l \in C_j^m} |g_l|^{-2} \Big)^{1/2}, \tag{23}$$

where $|C_j^m|$ denotes the cardinality of C_j^m. The WaveD (level-by-level) thresholds are defined according to each orientation ($m = 1, 2, 3$) by

$$\lambda_j^m := \gamma\, \sigma\, \tau_j^m\, (\log n)/n, \tag{24}$$

where γ is a constant and the finest scale j_1 is determined from

$$2^{j_1} = (n/\log n)^{1/(1+2\nu)}. \tag{25}$$

Here ν is a decay parameter which depends on the Degree of Ill-Posedness of the convolution model (2); for example the WaveD asymptotic theory as prescribed in [5] (extended here to the 2D-setting) includes convolution kernel g such that

$$\Big(|C_j^m|^{-1} \sum_{l \in C_j^m} |g_l|^{-2} \Big)^{1/2} \asymp 2^{j\nu}.$$

Condition (C_ν) typically holds for ordinary "smooth" convolution where $|g_l| \sim (|l_1| + |l_2|)^{-\nu}$, and in the 1-dimensional setting it has been shown to hold for certain oscillatory cases, such as arise with irrational boxcar blur [6].

4 Examples

4.1 WaveD Software

The `WaveD2.0` software package used to prepare all figures and table in this paper is available at `http://www.maths.usyd.edu.au:8000/u/marcr/`. The

2D-WaveD algorithm as described in section 5 is implemented in `matlab` and is based on the repeated application of appropriate discrete Fourier transform at each resolution level (and orientation) as illustrated on the diagram p.9. For an $n \times n$ image there are at most $J = \log_2(n) - 1$ resolution levels and each 2D-Fourier transform takes $O(n^2 \log n)$-steps. Hence, in total the 2D-WaveD algorithm requires only $O(n^2(\log n)^2)$-steps for an $n \times n$ image.

4.2 Image deblurring using WaveD

We have tested finite sample properties of the WaveD algorithm using a wide range of images at various Signal-to-Noise-Ratios. We illustrate our findings using the well-known '*Camera*' image, see e.g. [9]. The original image (256×256) is depicted in Figure 3(a). In Figure 3(b) we illustrate the white noise model (2) with medium noise level $\sigma_{med} = 0.002$ and a smooth blurring kernel g with DIP $\nu = 1$. In Figure (b) the Blurred-Signal-to-Noise-Ratio (BSNR) is $40dB$ as in the default setting of [9] . In our study (summarised in Table 1) we also report performances for low noise level $\sigma_{low} = 0.0006$ (BSNR=$50dB$) and high noise level $\sigma_{high} = 0.065$ (BSNR=$10dB$).

Table 1: *Monte-Carlo approximations to RMISE=*$\sqrt{E\|\hat{f} - f\|_2^2}$ *with* $N = 100$ *replications.*

Method	Time (100 repl.)	Low noise	Medium noise	High noise
WaveD	4.5 min	0.1854	0.1886	0.2442
ForWaRD	55.8 min	0.1581	0.1812	0.2953

4.3 Results

For each combination of noise level (low, medium and high) we compute the Monte Carlo approximation to the Root-Mean-Integrated-Square-Error (RMISE) with $N = 100$ replications. For comparison purposes, we included the results of the recently developed ForWard algorithm, see [9].

The general pattern seen in Table 1 suggests that for high noise level WaveD outperforms ForWarD whereas for low noise level ForWarD outperforms WaveD. For medium noise level both WaveD and ForWaRD are very close. A comparison of Figure 3(c) and 3(d) suggests that the image details (e.g. eyes, mouth) are more visible on the WaveD estimation than in the ForWarD image which remains a little blurred due to the regularisation. On

Figure 3: (a) Original image; (b) blurred and noisy image; (c) ForWard;(d) WaveD.

the other hand the general appearance of the ForWarD estimates is slightly smoother than the WaveD image.

4.4 Discussion

These preliminary results are very encouraging given that the ForWarD method is known to have very good numerical properties and outperforms classical deblurring methods based on Wiener filtering. A key feature of the WaveD method is that it is fairly simple and with fast computation (nearly as fast as the fast Fourier Transform). We are confident that the WaveD deblurring algorithm as presented here could further be improved using more elaborate thresholding devices, see e.g. the discussion in [5]. Our numerical study seems to suggest that diagonal coefficients at large resolution level are harder to recover than horizontal or vertical components. This is confirmed by the theory since an inspection of the WaveD paradigm (21) shows that for diagonal coefficients ($m = 3$) the support of $\hat{\Psi}^3$ is wider than that of $\hat{\Psi}^m, m = 1, 2$ (see also Figure 2). This yields larger variances for diagonal coefficients estimates. This results in a lack of diagonal coefficients for the WaveD image (Figure 3 (d)). There is a similar phenomenon in ForWarD images (Figure 3(c)) but the regularisation tends to minimise this effect although this also erases some details in the image. We are confident that the general aspect of the WaveD images could further be improved by using more elaborated thresholding strategires. A promising future research project would be to improve the general (diagonal) appearance of WaveD while keeping its fine details along horizontal and vertical axis.

5 Algorithm

Our algorithm is a based on a mathematical idea introduced in [3] for the 1-dimensional setting. The idea is that, at a given resolution level, both wavelet transform and inverse wavelet transform can be expressed as convolutions with certain wavelet functions. The WaveD algorithm takes full advantage of the Fast Fourier Transform to compute those convolutions. Below we present an extension of the WaveD algorithm to the 2-dimensional setting where we take care of the 3 possible orientations: horizontal, vertical and diagonal.

- Averaging shifts can be done level by level according to given orientation $m = 1, 2$ and 3. To simplify the presentation we omit the scaling term $m = 0$. Recalling the multi-resolution structure of a generic wavelet estimator we have

$$(T_{-h} f_h^m(x)) = \sum_{j=j_0-1}^{j_1} \sum_{k \in I_j^m} \eta(\tilde{\beta}_{j,k}^{m,h}) T_{-h} \Psi_{j,k}^m(x) := \sum_{j=j_0-1}^{j_1} P_j^{m,h}(x).$$

$$f_H^m(x) = Ave_{h \in H}(T_{-h} f_h^m(x)) = \sum_{j=j_0-1}^{j_1} Ave_{h \in H} P_j^{m,h}(x) := \sum_{j=j_0-1}^{j_1} f_j^{m,H}(x), \tag{26}$$

and finally we sum the horizontal, vertical and diagonal contributions:

$$f_H(x) = \sum_{m=1}^{3} f_H^m(x).$$

- For the TI-estimator the approximations $f_j^{m,H}$ $(m = 1, 2, 3)$ can be expressed in terms of convolutions with certain wavelet functions. The following result is a direct extension of proposition 1 of [3].

Proposition 1. *Let $k = (k_1, k_2)$ and $\tilde{\Psi}^m_{j,k}(x) = \Psi^m_{j,k}(-x)$. For any resolution level j and shift $h = (h_1, h_2)$ we have, for $m = 1, 2$ and 3,*

$$\beta^{m,h}_{j,k} = (f \star \tilde{\Psi}^m_{j,0})(\Delta^h_k) := A^m_j(\Delta^h_k), \tag{27}$$

where $\Delta^h_k = k/2^j + h$, and for $H = \{1/n, 2/n, ..., (n-1)/n\} \times \{1/n, 2/n, ..., (n-1)/n\}$ we have

$$f^{m,H}_j(x) = (2^j/n^2) \times (B_j * \Psi^m_{j,0})(x), \tag{28}$$

where

$$B^m_j(\Delta^h_k) := \eta_j(A^m_j(\Delta^h_k)) = A^m_j(\Delta^h_k).I(|A^m_j(\Delta^h_k)| > \lambda^m_j), \tag{29}$$

and $f * g$ denotes the circular convolution product.

For Meyer wavelets which have compact supports in the Fourier domain (see Figure 1), the convolution representations (27) and (28) can be computed in the Fourier domain:

$$\begin{array}{ccccc}
\text{Wavelet domain} & \text{Time domain} & & \text{Fourier domain} & \\
 & f & \longrightarrow^{F} & \hat{f} & \\
 & & & \downarrow & \times \hat{\Psi}^{m,*}_{j,0,0} \\
A^m_j & & \longleftarrow^{F^{-1}} & \hat{A}^m_j & \\
\downarrow_{\eta_j} & & & & \\
B^m_j & & \longrightarrow^{F} & \hat{B}^m_j & \\
 & & & \downarrow & \times \hat{\Psi}^{m}_{j,0,0} \\
 & f^{m,H}_j & \longleftarrow^{F^{-1}} & \hat{f}^{m,H}_j & \times 2^j/n^2
\end{array}$$

In the previous diagram F, F^{-1} denotes the Fourier transform and its inverse.

References

[1] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, Institute of Physics, Bristol and Philadelphia, 1998.

[2] R. Coifman and D. Donoho, *Translation-invariant de-noising*, in *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim, eds., New York: Springer Verlag, vol. 103 of *Lecture Notes in Statistics* (1995) pp. 125–150, pp. 125–150.

[3] D. Donoho and M. Raimondo, *Translation invariant deconvolution in a periodic setting*, The International Journal of Wavelets, Multiresolution and Information Processing **14** (2004). In press.

[4] E. Hernàndez and G. Weiss, *A First Course on Wavelets*, CRC Press, 1996.

[5] I. Johnstone, G. Kerkyacharian, D. Picard, and M. Raimondo, *Wavelet deconvolution in a periodic setting*, Journal of the Royal Statistical Society, Series B **66** (2004), 547–573. With discussion pp.627-652.

[6] I. M. Johnstone and M. Raimondo, *Periodic boxcar deconvolution and diophantine approximation*, Annals of Statistics **32** (2004), 1781–1804.

[7] S. Mallat, *A wavelet tour of signal processing (2nd Edition)*, Academic Press Inc., San Diego, CA, 1998.

[8] Y. Meyer, *Ondelettes et Opérateurs-I*, Hermann, 1990.

[9] R. Neelamani, H. Choi, and R. Baraniuk, *Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems*, IEEE Transactions on signal processing **52** (2004), 418–433.