Journal of Electronic Imaging 11(3), 1–0 (July 2002).

Discrete wavelet transform implementation in Fourier domain for multidimensional signal

Frederic Nicolier
IUT de Troyes
LAM, URCA–Université de Champagne-Ardenne
9 Rue de Quebec
Boîtel Postale 396
F-10026 Troyes, Cedex, France
E-mail: f.nicolier@iut-troyes.univ-reims.fr

Olivier Laligant
Frederic Truchetet
IUT Le Creusot
LE2I, Université de Bourgogne
12 Rue de la Fonderie
F-71200 Le Creusot, France

Abstract. *Wavelet transforms are often calculated by using the Mallat algorithm. In this algorithm, a signal is decomposed by a cascade of filtering and downsampling operations. Computing time can be important but the filtering operations can be speeded up by using fast Fourier transform (FFT)-based convolutions. Since it is necessary to work in the Fourier domain when large filters are used, we present some results of Fourier-based optimization of the sampling operations. Acceleration can be obtained by expressing the samplings in the Fourier domain. The general equations of the down- and upsampling of digital multidimensional signals are given. It is shown that for special cases such as the separable scheme and Feauveau's quincunx scheme, the samplings can be implemented in the Fourier domain. The performance of the implementations is determined by the number of multiplications involved in both FFT-convolution-based and Fourier-based algorithms. This comparison shows that the computational costs are reduced when the proposed implementation is used. The complexity of the algorithm is $O(N \log N)$. By using this Fourier-based method, the use of large filters or infinite impulse response filters in multiresolution analysis becomes manageable in terms of computation costs. Mesh simplification based on multiresolution "detail relevance" images illustrates an application of the implemenentation.* © 2002 SPIE and IS&T. [DOI: 10.1117/1.1479701]

1 Introduction

Space-scale analysis is a very popular tool for signal processing especially for two-dimensional (2D) signal processing. Many applications deal with image compression, multiresolution edge detection and more generally all algorithms that benefit from the multiresolution approach such as Markovian modelization or texture segmentation. The multiresolution analysis algorithm developed by Mallat[1] has largely contributed to this popularity. In this algorithm, a signal is decomposed into approximation and detail signals by a cascade of filtering and downsampling operations.

In all these applications, wavelets are associated with linear phase filters, this property being largely admitted to be a crucial point for good results in image processing. One-dimensional linear-phase filters associated with orthogonal bases have an infinite impulse response (IIR). In multidimensional processing, finite impulse response (FIR) and linear-phase filters associated with orthogonal analysis are theoretically possible but involve very constraining conditions. The problem is often solved by using a biorthogonal basis that leads to compact linear phase filters (FIR).[2] However, due to the maximal independence of the signal components obtained, the interest in an orthonormal basis seems to be obvious. Kovačević and Vetterli have presented such filters associated with orthogonal analysis but the corresponding wavelets are not accurately localized in the frequency domain.[3]

The difficulty is that implementation of IIR filters with classical methods is hardly ever in accordance with real-time applications. When large filters are used in signal processing, it is necessary to work in the Fourier domain in order to have reduced computational time. A well-known result is that digital filters with more than 32 taps, in one-dimensional signal filtering, the Fourier domain provides reduced computational time.[4] A sensible implementation would use Fourier-based convolution consisting of a 2D fast Fourier transform (FFT), complex multiplications for the filtering and a 2D inverse FFT (IFFT). In this way, the filtering step of the algorithm can be accelerated. We have looked for some efficient method by which to accelerate the sampling step. If the samplings can be expressed in the Fourier domain, then some FFT and IFFT can be avoided. Therefore, we have looked for efficient implementation of

Paper 99066 received Nov. 17, 1999; revised manuscript received Sep. 4, 2001; accepted for publication Nov. 29, 2001.
1017-9909/2002/$15.00 © 2002 SPIE and IS&T.

the whole Mallat algorithm in the Fourier domain. Some results on sampling in the continuous Fourier domain can be found in Refs. 5–9. We have extended these results to digital multidimensional signals. The algorithm obtained is not tailored for a specific wavelet. It allows fast computation even with FIR filters such as Daubechies ones.

Presentation as well as an evaluation of this approach is proposed in this paper. For digital multidimensional signals, the general transcription in the Fourier domain of spatial sampling is given. Furthermore, several special samplings such as the one-dimensional, two-dimensional separable and quincunx cases are studied.[2] For the quincunx scheme, two representations are developed and we show that in the case of Feauveau's representation[10] the Fourier implementation is quite worthwhile in terms of computational cost.

This paper is organized as follows. The transcription of multidimensional sampling in the discrete Fourier domain is described in Sec. 2. Special cases of multidimensional samplings are studied in Sec. 3. The performance of the implementation in the Fourier domain of the whole wavelet transform algorithm is presented in Sec. 4 by comparing computational costs of the FFT-based-convolution method with those of the Fourier-based method presented. An application of the implementation is given in Sec. 5.

2 Multidimensional Sampling

Multidimensional samplings are an extension of one-dimensional ones. The sampling is expressed by multiplying the signal sample coordinates by a matrix $\mathbf{J}$ denoting the scaling (or dilation) operation. For example, in 2D the coordinates of a given sample are transformed by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{J} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{1}$$

For instance, in the separable case, the sampling matrix is given by

$$\mathbf{J} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

and in the quincunx sampling matrix by

$$\mathbf{J} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

2.1 Sampling in the Continuous Fourier Domain

As previously shown by Viscito, Dubois, Vetterli and co-workers,[5,6,9] downsampling in the continuous time domain can be expressed like in Eq. (2); the transcription in the continuous Fourier domain of these operations is shown in Eq. (3) where X_F is the Fourier transform of x:

$$y[\mathbf{k}] = x[\mathbf{J}\mathbf{k}], \tag{2}$$

$$Y_F(\boldsymbol{\omega}) = \frac{1}{D} \sum_{l=0}^{D-1} X_F(\mathbf{J}^{-T}\boldsymbol{\omega} - 2\pi\mathbf{J}^{-T}\mathbf{v}_l), \tag{3}$$

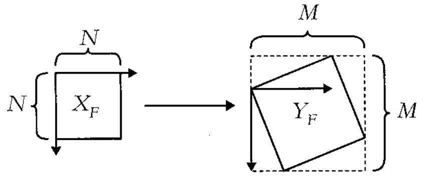


Fig. 1 Two-dimensional upsampling in the Fourier domain: Dilation and rotation are involved in the transformation.

where $D = |\det \mathbf{J}|$, $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_S)^T$ is an S-dimensional real vector and $\mathbf{v}_l$ denotes the vectors associated with the cosets derivated from $\mathbf{J}$. The operations involved in upsampling can also be expressed in the time and Fourier domains as shown in Eqs. (4) and (5).

$$y[\mathbf{k}] = \begin{cases} x[\mathbf{J}^{-1}\mathbf{k}], & \text{if } \mathbf{k} = \mathbf{J}\mathbf{n},\ \mathbf{n} \in \mathbb{Z}^S, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

$$Y_F(\boldsymbol{\omega}) = X_F(\mathbf{J}^T\boldsymbol{\omega}). \tag{5}$$

2.2 Sampling in the Discrete Fourier Domain

One can notice in Eqs. (3) and (5) that a Fourier transform in the continuous domain is used. Since digital signal processing needs discrete signals and operators, here in Sec. 2.2 the sampling operations are transcribed into equations in the discrete Fourier domain. The discrete Fourier transform (DFT) of a multidimensional signal $x[\mathbf{k}]$ is denoted by $\hat{x}[\mathbf{m}]$, where $\mathbf{k}$ and $\mathbf{m}$ are vectors. In the discrete domain, sample indices of the signals have to be integers and not real numbers since they have to be with the use of $\boldsymbol{\omega}$. Thus Eqs. (3) and (5) need to be rewritten according to this. Explicit developments are given only for upsampling but they can be done easily for downsampling. Examples are taken for a 2D signal (image). A dilation and a rotation centered on the origin are involved in the transformation realized by the matrix multiplications. As shown in Fig. 1, an $N \times N$ image X_F is transformed into an $M \times M$ image denoted Y_F and a rectangular lattice is used for storage. Equation (5) is rewritten as

$$Y_F(\boldsymbol{\omega}) = X_F(\boldsymbol{\omega}') \quad \text{with} \begin{cases} \boldsymbol{\omega} = \dfrac{2\pi}{M}\mathbf{m}, \\ \boldsymbol{\omega}' = \dfrac{2\pi}{N}\mathbf{m}', \end{cases} \tag{6}$$

where $\mathbf{m}$ is related to $\mathbf{m}'$ according to $\mathbf{m}' = (N/M)\,\mathbf{J}^T\mathbf{m}$. So Eq. (5) can be rewritten to give upsampling for discrete multidimensional signals:

$$\hat{y}[\mathbf{m}] = \hat{x}\left[\left(\frac{N}{M}\mathbf{J}^T\mathbf{m}\right) \bmod N\right], \tag{7}$$

where mod N is added because the DFT assumes signal periodicity. It can be verified that a similar demonstration can be conducted to express downsampling:

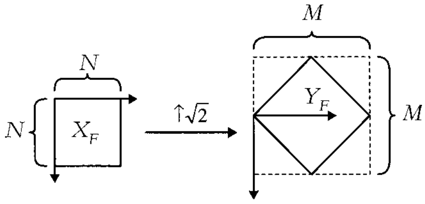


Fig. 2 Quincunx upsampling in the Fourier domain: Dilation and rotation of an $N\times N$ signal X_F in an $N\sqrt{2}\times N\sqrt{2}$ signal Y_F. The rotated Y_F signal is contained in a $M\times M$ image.

$$\hat{y}[\mathbf{m}]=\frac{1}{D}\sum_{l=0}^{D-1}\hat{x}\left[\left(\frac{N}{M}\mathbf{J}^{-T}\mathbf{m}-N\mathbf{J}^{-T}\mathbf{v}_l\right)\mod N\right]. \tag{8}$$

3 Special Two-Dimensional Samplings

It was shown in Sec. 2 that any multidimensional sampling can be transcribed into the discrete Fourier domain. Here in Sec. 3, two examples dealing with 2D sampling are given in detail: the separable scheme and the nonseparable quincunx scheme. Feauveau's quincunx representation is also studied and is shown to lead to very simple operations in the Fourier domain.

3.1 *Separable Case*

From Eq. (5), the upsampling in the discrete Fourier domain in the separable case is expressed by $Y_F(\omega_1,\omega_2)=X_F(\omega_1',\omega_2')$ where $\omega_1'=2\omega_1$ and $\omega_2'=2\omega_2$. In this operation signals are dilated by 2 and there is no rotation involved. An $M\times M$ signal is transformed into a $2M\times 2M$ one. If ω_1 and ω_2 are defined in discrete terms according to $\omega_1=(2\pi/2M)m_1$ and $\omega_2=(2\pi/2M)m_2$, then $\omega_1'=(2\pi/M)m_1'$ and $\omega_2'=(2\pi/M)m_2'$. Since $\omega_1'=2\omega_1$ and $m_1=m_1'$ ($\omega_2'=2\omega_2$ and $m_2=m_2'$) the transformation in the discrete Fourier domain of the separable upsampling is given by

$$\hat{y}[m_1,m_2]=\hat{y}[m_1 \mod M, m_2 \mod M]. \tag{9}$$

3.2 *Quincunx Rotating Scheme*

From Eq. (5), upsampling in the discrete Fourier domain for the quincunx case is expressed as

$$Y_F(\omega_1,\omega_2)=X_F(\omega_1+\omega_2,\omega_1-\omega_2). \tag{10}$$

A $\sqrt{2}$ dilation and a $\pi/4$ rotation are involved in this operation. As shown in Fig. 2, an $N\times N$ signal is transformed into an $M\times M$ one. In continuous terms, because X_F is an $N\times N$ square and Y_F is an $N\sqrt{2}\times N\sqrt{2}$ rotated square, M is related to N by $M=2N$. If equation Eq. (10) is rewritten:

$$Y_F(\omega_1,\omega_2)$$
$$=X_F(\omega_1',\omega_2') \text{ with } \begin{cases}\omega_1=\dfrac{2\pi}{M}m_1,\ \ \omega_2=\dfrac{2\pi}{M}m_2,\\ \omega_1'=\omega_1+\omega_2=\dfrac{2\pi}{N}m_1',\ \ \omega_2'=\omega_1-\omega_2=\dfrac{2\pi}{N}m_2'\end{cases}. \tag{11}$$

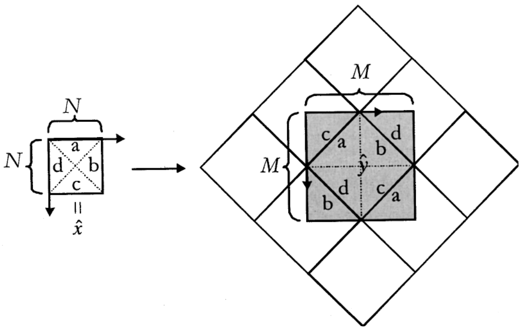


Fig. 3 Complete quincunx upsampling in the Fourier domain of an $N\times N$ signal $\hat{x}$ in an $N\sqrt{2}\times N\sqrt{2}$ signal $\hat{y}$ contained in a $M\times M$ image. Due to Fourier domain periodicity, parts *a*, *b*, *c* and *d* are duplicated. The origin of the signal is translated to produce a one-block signal $\hat{y}$.

So m_1 and m_2 are related to m_1' and m_2' according to

$$\begin{cases} m_1'=\dfrac{N}{M}(m_1+m_2),\\ m_2'=\dfrac{N}{M}(m_1-m_2)\end{cases}. \tag{12}$$

Equation (10) can be rewritten to give quincunx upsampling for digital multidimensional signals as

$$\hat{y}[m_1,m_2]$$
$$=\hat{x}\left[\frac{N}{M}(m_1+m_2)\mod N,\frac{N}{M}(m_1-m_2)\mod N\right], \tag{13}$$

As shown in Fig. 2 the rotation involved in the transformation is centered on the origin of the image. Therefore the transformed image Y_F obtained is distributed in several blocks. In order to produce a one-block signal, an $M/2$ translation into the discrete Fourier domain (i.e., a π translation into the continuous Fourier domain) must be included to change the image's origin. The result of this translation is shown in Fig. 3 where $\hat{y}$ is contained in an $M\times M$ signal and, as the space is periodized, quarters of $\hat{y}$ are duplicated (see a–d in Fig. 3). So the final transformation in the discrete Fourier domain of the quincunx over sampling is given by

$$\hat{y}[m_1,m_2]$$
$$=\hat{x}[n_1,n_2] \text{ with } \begin{cases} n_1=\frac{1}{2}(m_1+m_2)\mod N,\\ n_2=\frac{1}{2}(m_1-m_2)\mod N.\end{cases} \tag{14}$$

Since this transformation is applied on $\hat{x}$ samples, only samples of $\hat{y}$ that correspond to even (m_1+m_2) and (m_1-m_2) terms are calculated (i.e., when n_1 and n_2 are integers).

3.3 *Feauveau's Quincunx Scheme*

One way to represent quincunx sampling is the previous one, where the images of two successive scales are rotated

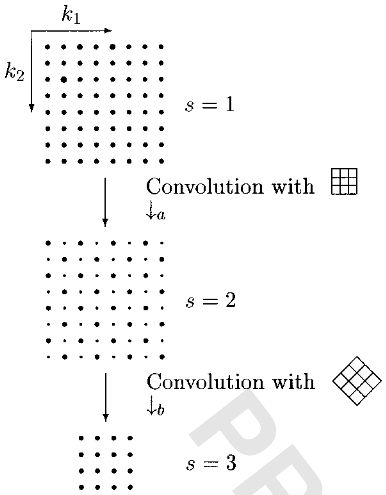


Fig. 4 Feauveau-quincunx scheme: Decomposition of a digital approximation at scale $s=1$ into approximations at coarser scales ($s=2$ and 3). The samples kept are indicated by the big dots. Zeros are put at small dots.

by $\pi/4$. Another representation was proposed by Feauveau,[10] in which the signal axes are kept steady but the filters are rotated and two types of downsampling and upsampling are used. The analysis algorithm is shown in Fig. 4. For even scale computation, convolution with normal filters and keep samples where (k_1+k_2) is even (downsampling a:$\downarrow_a$). For odd scales, convolution with rotated filters and keep one column and one row out of two (downsampling b:$\downarrow_b$). For the synthesis, the upsampling is coded by putting zeros between each column and each row.

These samplings can easily be conducted in the discrete Fourier domain as shown in Eqs. (15)–(17) (see the Appendix).

$$\hat{y}_{\downarrow_a}[k_1,k_2]_{k_1,k_2\in[0,N-1]} = \frac{1}{2}\left(\hat{x}[k_1,k_2]+\hat{x}\left[k_1+\frac{N}{2},k_2+\frac{N}{2}\right]\right), \tag{15}$$

$$\hat{y}_{\downarrow_b}[k_1,k_2]_{k_1,k_2\in[0,N/2-1]} = \frac{1}{4}\left(\hat{x}[k_1,k_2]+\hat{x}\left[k_1+\frac{N}{2},k_2\right]+\hat{x}\left[k_1+\frac{N}{2},k_2\right] +\hat{x}\left[k_1+\frac{N}{2},k_2+\frac{N}{2}\right]\right). \tag{16}$$

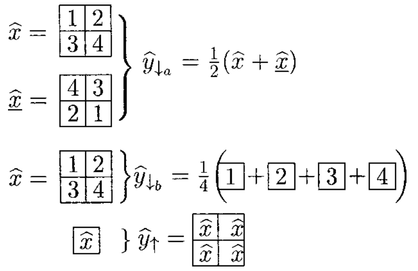

$\widehat{x}=$ 1 2 3 4
$\underline{\widehat{x}}=$ 4 3 2 1
$\widehat{y}_{\downarrow a}=\frac{1}{2}(\widehat{x}+\underline{\widehat{x}})$
$\widehat{x}=$ 1 2 3 4
$\widehat{y}_{\downarrow b}=\frac{1}{4}(1+2+3+4)$
$\widehat{x}$
$\widehat{y}_{\uparrow}=$ $\widehat{x}$ $\widehat{x}$ $\widehat{x}$ $\widehat{x}$

Fig. 5 Feauveau-quincunx scheme: Geometric interpretation of the samplings in the Fourier domain.

$$\hat{y}_{\uparrow}[k_1,k_2]_{k_1,k_2\in[0,N-1]}=\hat{x}[\widetilde{k_1},\widetilde{k_2}] \quad \text{with} \begin{cases} \widetilde{k_1}=k_1 \mod \dfrac{N}{2}, \\ \widetilde{k_2}=k_2 \mod \dfrac{N}{2}. \end{cases} \tag{17}$$

Equation (15) translates downsampling $\downarrow_a$ into the discrete Fourier domain, Eq. (16) is for downsampling $\downarrow_b$ and Eq. (17) deals with upsampling. A geometrical interpretation of these equations is shown in Fig. 5. Downsampling $\downarrow_a$ is interpreted as an addition between $\hat{x}$ and $\underline{\hat{x}}$, where $\underline{\hat{x}}$ is obtained by permutation of $\hat{x}$ quadrants (1–4 and 2–3). $\hat{y}_{\downarrow b}$ is obtained by the addition of all the quadrants of $\hat{x}$. Finally $\hat{y}_{\uparrow}$ is obtained by a duplication of $\hat{x}$.

4 Performance

In the earlier discussion the simplicity of processing samplings in the Fourier domain has been pointed out. The following discussion will characterize the advantages of using the Fourier domain in terms of computational costs (i.e., the number of real additions and multiplications) for all the operations involved in a multiresolution analysis. The results given are general and do not take into account any optimization and therefore do not benefit from any symmetry of the signals or from special samplings.

2D signals of finite size $M\times M$ are considered as are filters associated with a multiresolution analysis. Computational costs are evaluated up to scale L. According to the samplings, at scale s (s varying from 0 to L) the signal size is approximated by $M/D^{s/2}\times M/D^{s/2}$ where $D=|\det \mathbf{J}|$. So the computations at scale s of both detail and approximation are only done for M^2D^{-s} samples.

The comparison is done between the DFT-based convolution algorithm and our Fourier algorithm. These algorithms are synthesized in Figs. 6 and 7. For convenience, only the approximation of each scale is calculated.

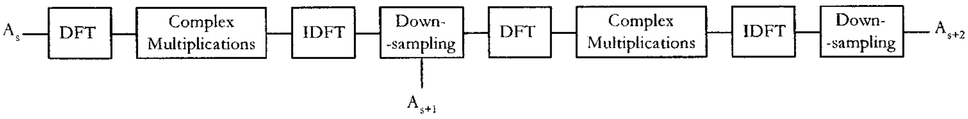


Fig. 6 DFT-based convolution algorithm: Decomposition of a digital approximation (A_s) into approximations at coarser scales (A_{s+1} and A_{s+2}). Filtering is realized in the Fourier domain. Sampling is realized in the time domain.

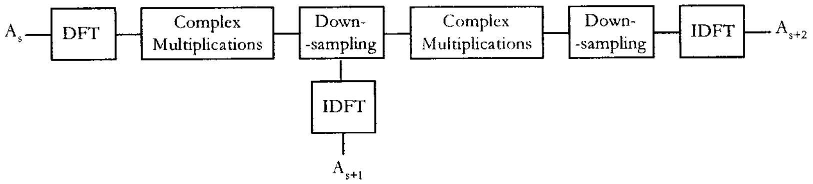


Fig. 7 Fourier domain algorithm: Decomposition of a digital approximation (A_s) into approximations at coarser scales (A_{s+1} and A_{s+2}). Filtering and sampling are realized in the Fourier domain.

4.1 *Computation Cost in the Fourier Domain*

Implementation of the whole wavelet transform in the Fourier domain consists of computing the Fourier transform of the input signal, filtering (complex multiplications), processing the Fourier domain equivalent of the spatial subsamplings and finally computing inverse Fourier transforms for the output signals (approximations and details). The computational cost of the FFT of an $M\times M$ signal is $\hat{\mathcal{N}}^x_{\mathrm{FFT}}=4M^2(\log_2 M-2)$. The output must be given in the time domain, so the cost of the IFFT of the details of scale s is given by

$$\hat{\mathcal{N}}^x_{\mathrm{IFFT}}(s)=4D^{-s}M^2[\log_2(MD^{-(s/2)})-2]. \tag{18}$$

The filtering operations are just complex multiplications (i.e., two real multiplications and four real additions) with the filter in the Fourier domain, so the cost of filtering a signal in V_s is $\hat{\mathcal{N}}^x_f(s)=2D^{-s}M^2$. The downsampling is processed by a matrix multiplication [see Eq. (8)] for each sample of the destination signal. One 2×2 matrix multiplication involves four real multiplications and two real additions. So to obtain a signal at scale s, the multiplication cost for the downsampling is $\hat{\mathcal{N}}^x_{\downarrow}(s)=4D^{-s}M^2$. Finally the computational costs are summed over scales to obtain the final multiplication cost of an L scales two-dimensional wavelet transform in the Fourier domain:

$$\mathcal{N}^x_{\mathcal{F}}(L)=\hat{\mathcal{N}}^x_{\mathrm{FFT}}+\sum_{s=0}^{L-1}\hat{\mathcal{N}}^x_f(s)+\sum_{s=1}^{L}[\hat{\mathcal{N}}^x_{\downarrow}(s)+\hat{\mathcal{N}}^x_{\mathrm{IFFT}}(s)]. \tag{19}$$

4.2 *Computational Cost Using DFT-based Convolutions*

An efficient and classical implementation of the wavelet algorithm consists of processing the convolution in the Fourier domain. The downsampling costs are the same as in the Fourier domain algorithm $[\mathcal{N}^x_{\downarrow}(s)=\hat{\mathcal{N}}^x_{\downarrow}(s)]$. The costs of the convolutions are the costs of a FFT, plus the costs of complex multiplications for filtering, plus the costs of an IFFT.

For one-scale processing, the computational cost of a FFT (or an IFFT) is $\mathcal{N}^x_{\mathrm{FFT}}(s)=4M^2D^{-s}[\log_2(MD^{-(s/2)})-2]$. The filtering operations are just complex multiplications (i.e., two real multiplications and four real additions), thus the cost of filtering a signal in V_s is $\hat{\mathcal{N}}^x_f(s)=2D^{-s}M^2$. These computational costs are summed over scales to obtain the final multiplication cost of an L scale two-dimensional transform using DFT-based convolutions:

$$\mathcal{N}^x_C(L)=\sum_{s=0}^{L-1}(2\mathcal{N}^x_{\mathrm{FFT}}(s)+\hat{\mathcal{N}}^x_f(s))+\sum_{s=1}^{L}\mathcal{N}^x_{\downarrow}(s). \tag{20}$$

4.3 *Comparison Between the Algorithms*

In Secs. 4.1 and 4.2 we have specified how $\mathcal{N}^x_C$ and $\mathcal{N}^x_F$ are evaluated. Their performance is evaluated by analyzing cost ratios. The ratio of the FFT-based convolution algorithm cost to the Fourier algorithm cost is $R(L)=[\mathcal{N}_C(L)/\mathcal{N}_F(L)]$. This ratio depends both on the image size $M\times M$ and on the determinant D of the dilation matrix.

As shown in Figs. 8 and 9, two ratios are calculated: R_2 for the quincunx sampling and R_4 for the separable sampling. Four curves are drawn for each scheme, and they correspond to an analysis made from scale 1 to scale L ($L=1,2,3$ or 4). For the two samplings, the ratio is about 1.8 for the deeper scale analysis ($L=4$).

These comparisons are made with nonoptimized algorithms. In the case of the quincunx algorithm, the samplings can be simplified by using Feauveau's scheme (see Sec. 3.3). Figure 10 shows the ratio of the quincunx algorithm to Feauveau's algorithm cost using the Fourier implementation. The curves show a little decrease in the computational cost (about 6%) but this ratio does not take into account the advantage of implicit periodization of the Feauveau sampling.

The implementation costs are summarized in Fig. 11. The memory requirements given include all the filters

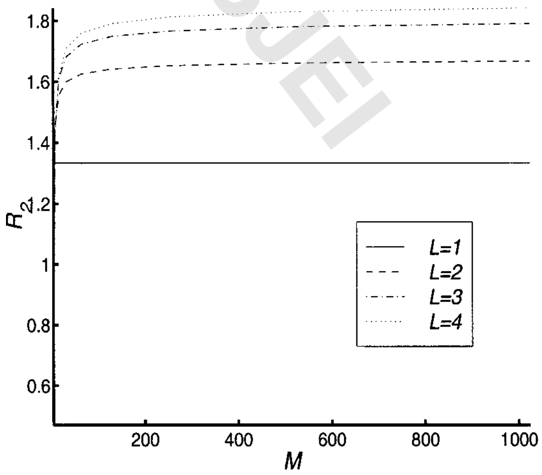


Fig. 8 Comparisons between algorithms: Ratios of FFT-based convolution algorithm costs to Fourier algorithm costs as a function of the image size M and the analysis depth L. Quincunx sampling.

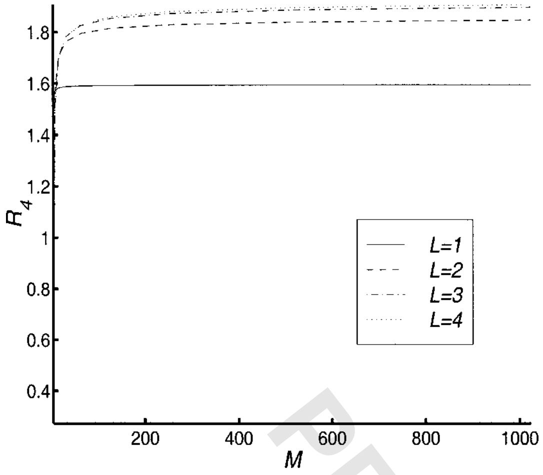


Fig. 9 Comparisons between algorithms: Ratios of FFT-based convolution algorithm costs to Fourier algorithm costs as a function of the image size *M* and the analysis depth *L*. Separable sampling.

needed. These filters can be preloaded into memory in order to avoid hard drive access. To be efficient, the frequency responses of the filters are used in order to avoid pointless FFTs. The number of operations given corresponds to 32 bit real multiplications.

We have made a comparison of our implementation with polyphase factorization. The filter size used in our implementation is equal to the image size. The complexity of the polyphase implementation has thus been evaluated with this size filter. The complexity of standard polyphase factorization is $O(N^2)$.[9] The complexity of our implementation is $O(N \log N)$. A comparison of these two complexities shows that our implementation is more efficient than polyphase factorization.

5 Application

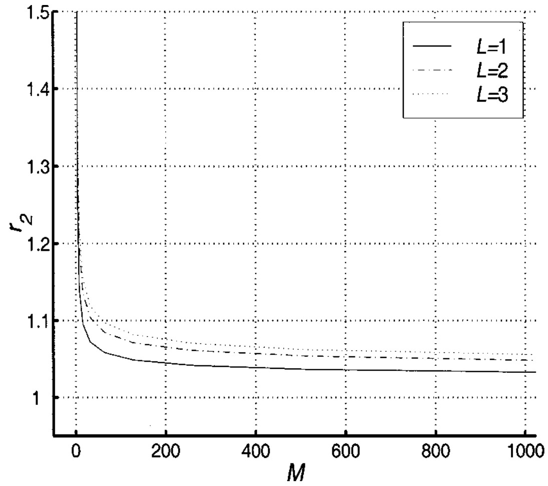


Fig. 10 Comparison between Feauveau schemes: Ratio of the classical-quincunx scheme cost to the Feauveau-quincunx scheme cost as a function of the image size *M* and the analysis depth *L*.

Image Size ($N \times N$)	$N=64$	$N=128$	$N=256$	$N=512$
Mops for $D=2$	0.3	1.7	7.7	34.9
Memory (Mo) ($D=2$)	0.2	0.9	3.6	14.3
Mops for $D=4$	0.5	2.3	10.4	45.9
Memory (Mo) ($D=4$)	0.3	1.3	5.4	21.5

Fig. 11 Number of operation (real multiplications) and memory requirements of our Fourier domain wavelet transform implementation for three level decomposition ($L=3$). The memory requirements are computed using a float 32 number representation.

We have used our Fourier implementation in a mesh simplification algorithm previously developed.[11] This mesh simplification process inherently deals with multitextured data sets using multidimensional feature space. Figure 12 shows a flow chart of the simplification process. Each texture in the data set is a dimension in feature space. A texture of particular interest is what we call the "detail relevance" image. This image is the result of applying the 2D quincunx-wavelet transform to a three-dimensional (3D) mesh. Detail extraction is achieved using multiresolution analysis (MRA) based on the wavelet cascade approach. The MRA process extracts detail information at various resolutions and produces a texture image that shows the relevance of a vertex to the mesh structure. The user has input in this process to select which resolutions are more important. This approach is useful for filtering noise, preserving discontinuities, mining surface details, reducing data, and many other applications. The mesh simplification process is the process of reducing the number of vertices that compose a mesh. Our process is based on an edge collapsing process that collapses two vertices into one. The decision to collapse one edge over others is made in pattern space (or feature space). Each pattern is a dimension in this space. Pattern space is composed of multiple types of information and measurements of the mesh. For this study, they are restricted to geometrical coordinates, curvature, normal, and more importantly the results of multiresolution analysis interpreted as binary information (0: absence of detail, 1: presence of detail).

The binary detail relevance image is obtained from detail images of multiresolution analysis of the range image. The range image is decomposed over scales using quincunx analysis. Each detail image is then reconstructed at the initial scale. This reconstruction is realized via an inverse transform where the detail signals corresponding to other scales are set to zero. Each detail image is reconstruced via this process. Finally, one information image is obtained

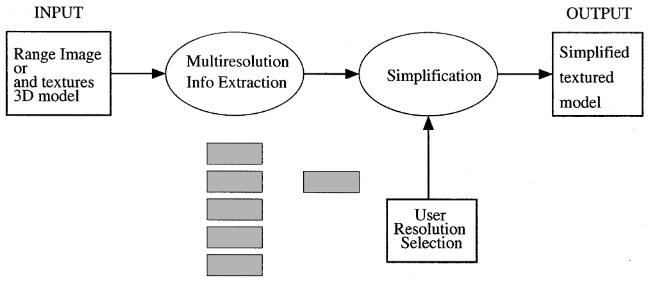

INPUT
OUTPUT
Range Image or and textures 3D model
Multiresolution Info Extraction
Simplification
Simplified textured model
User Resolution Selection

Fig. 12 Flow chart of the mesh simplification algorithm. The user controls the simplification by selecting relevant detail images. The detail images are obtained from a quincunx multiresolution analysis.

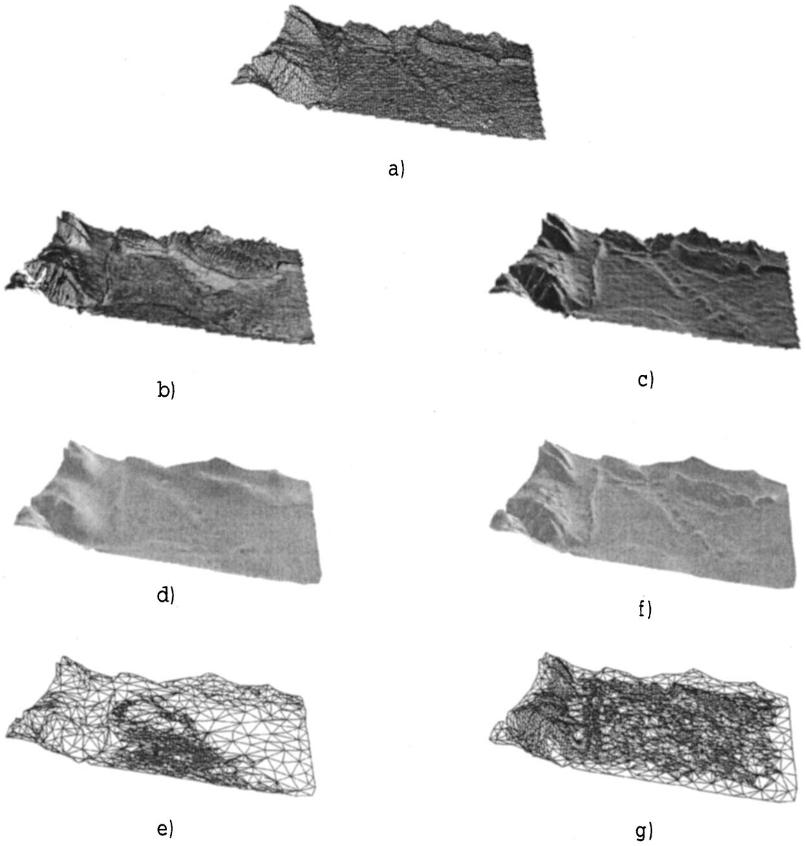


Fig. 13 Simpification of a digital elevation map (DEM) model: (a) initial wire frame (12 000 vertices, texture image 512×256 and 15000 faces), (b) model with thermal texture (texture image 512×256), size: 0.7 Mb, (c) model display with no texture, (d) simplification of the model based on texture information (thermal area in red on the color texture is kept unchanged), (e) its reduced wire frame (reduction factor is 5), size: 124 kb, (f) simplification of the model based on geometric information, (g) its reduced wire frame (reduction factor is 5) constructed with segmented detail images from scale 6, size 124 kb.

from a weighted sum of the reconstructed detail images. The information image is thresholded to obtain the binary detail relevance image. An example of mesh simplification is given in Fig. 13.

The computational time of the information image is very important. This time was reduced using our optimized Fourier domain implementation. The filters were computed off line and are loaded in memory at the beginning of the application. The image size is 256×256, the filter size is 25 ×25, the depth analysis is $L=6$ and since a quincunx method is used, $D=2$. The computational time ratio between the space domain implementation and our implementation is about 41. This means that the computation time for the information image is a few seconds in simple MATLAB implementation.

6 Conclusion

Some new considerations dealing with implementation in the Fourier domain of multiresolution analysis on an orthornormal basis were presented. Since it is more efficient to work in the Fourier domain when large filters are used, in this paper we have emphasized a Fourier equivalent of up- and downsampling operations for multidimensional digital signals. It has been shown that these processings can be easily be performed by simple operations (spectrum repetitions and additions for the Feauveau scheme). The resulting scheme for multiresolution analysis was described and was compared with the classical one. The improvement obtained in terms of computational cost is noticeable. In this way, the use of large or IIR filters in multiresolution analysis is quite manageable in terms of computational costs. The performance of the implementation was given in terms of the number of operations and memory requirement estimations. The complexity of the algorithm is $O(N \log N)$ which is lower than that of polyphase factorization. The proposed implementation was applied to mesh simplification based on quincunx multiresolution analysis. The resulting computational time is a few seconds.

Since the performances given were presented with no simplification we think that in special applications, operations could be directly implemented in the Fourier domain in order to avoid some FFTs. For example, in an image compression–decompression algorithm, quantization (scalar or vectorial) could be achieved with benefit directly in Fourier space, especially when a complete set of coefficients is set to zero (diagonal details of the finest scales, for example), in this case, a Fourier implementation scheme seems to be particularly worthwhile. Other benefits can be taken from the Fourier domain. For instance, as the periodicity of a signal is implicitly included in the definition of the Fourier domain, when signals are processed there are no requirements to make any periodization of the signal, especially when using the separable or Feauveau quincunx schemes.

Appendix

Downsampling a

$$y(k_1,k_2)_{k_1,k_2\in[0,N-1]}=\tfrac{1}{2}x(k_1,k_2)(1+e^{j(k_1+k_2)\pi}),$$

$$\begin{aligned}\hat{y}(n_1,n_2)_{n_1,n_2\in[0,N-1]}&=\frac{1}{2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}x(k_1,k_2)(1+e^{j(k_1+k_2)\pi})e^{-(j2\pi k_1n_1/N)}e^{-(j2\pi k_2n_2/N)},\\ &=\frac{1}{2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}x(k_1,k_2)e^{-[j2\pi(k_1n_1+k_2n_2)/N]}\\ &\quad+\frac{1}{2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}x(k_1,k_2)e^{-(j2\pi k_1/N)[n_1-(N/2)]}e^{-(j2\pi k_2/N)[n_2-(N/2)]},\\ &=\frac{1}{2}\hat{x}(n_1,n_2)+\frac{1}{2}\hat{x}\left(n_1-\frac{N}{2},n_2-\frac{N}{2}\right).\end{aligned}$$

Downsampling b

$$y(k_1,k_2)_{k_1,k_2\in[0,(N/2)-1]}=x(2k_1,2k_2),$$

$$\begin{aligned}\hat{y}(n_1,n_2)_{n_1,n_2\in[0,(N/2)-1]}&=\sum_{k_1=0}^{N/2-1}\sum_{k_2=0}^{N/2-1}x(2k_1,2k_2)e^{-[j2\pi k_1n_1/(N/2)]}e^{-[j2\pi k_2n_2/(N/2)]},\\ &=\frac{1}{2}\sum_{k_1=0}^{N/2-1}\sum_{k_2=0}^{N/2-1}x(2k_1,2k_2)e^{-[j2\pi(k_1n_1+k_2n_2)/(N/2)]}(\underbrace{1+e^{-j2\pi(k_1+k_2)}}_{=2}),\\ &=\frac{1}{2}\sum_{k_1=0}^{N/2-1}\sum_{k_2=0}^{N/2-1}x(2k_1,2k_2)e^{-[j2\pi(2k_1)n_1/N]}e^{-[j2\pi(2k_2)n_2/N]}(\underbrace{1+e^{-j2\pi(k_1+k_2)}}_{=2}),\\ &\quad+\frac{1}{2}\sum_{k_1=0}^{N/2-1}\sum_{k_2=0}^{N/2-1}x(2k_1+1,2k_2+1)e^{-[j2\pi(2k_1+1)n_1/N]}e^{-[j2\pi(2k_1+1)n_2/N]}\\ &\quad\times(\underbrace{1+e^{-j(2k_1+1)}e^{-j(2k_2+1)}}_{=0}),\\ &\quad+x(2k_1,2k_2+1)\ \text{and}\ x(2k_1+1,2k_2)\ \text{null terms},\\ &=\frac{1}{4}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}x(k_1,k_2)e^{-(j2\pi k_1n_1/N)}e^{-(j2\pi k_2n_2/N)}\\ &\quad+\frac{1}{4}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}x(k_1,k_2)e^{-(j2\pi k_1n_1/N)}e^{-(j2\pi k_2n_2/N)}e^{-jk_1\pi}e^{-jk_2\pi}\\ &\quad\times(+e^{-jk_1\pi}\ \text{and}\ e^{-jk_2\pi}\ \text{terms}),\end{aligned}$$

$$\hat{y}(n_1,n_2)_{n_1,n_2\in[0,N/2-1]}=\frac{1}{4}\left[\hat{x}(n_1,n_2)+\hat{x}\left(n_1+\frac{N}{2},n_2+\frac{N}{2}\right)+\hat{x}\left(n_1,n_2+\frac{N}{2}\right)+\hat{x}\left(n_1+\frac{N}{2},n_2\right)\right].$$

Upsampling

$$y(k_1,k_2)_{k_1,k_2\in[0,N-1]}=\frac{1}{2}x\left(\frac{k_1}{2},\frac{k_2}{2}\right)(1+e^{j(k_1+k_2)\pi}),$$

$$\hat{y}(n_1,n_2)_{n_1,n_2\in[0,N-1]}=\frac{1}{2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}x\left(\frac{k_1}{2},\frac{k_2}{2}\right)e^{-j2\pi[(k_1n_1+k_2n_2)/N]}$$

$$+\frac{1}{2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}x\left(\frac{k_1}{2},\frac{k_2}{2}\right)e^{-j2\pi[(k_1n_1+k_2n_2)/N]}e^{jk_1\pi}e^{jk_2\pi}\begin{cases}p_1=\dfrac{k_1}{2}\\[2ex] p_2=\dfrac{k_2}{2}\end{cases}\quad\leftrightarrow\quad\begin{cases}k_1=2p_1,\\ k_2=2p_2,\end{cases}$$

$$\hat{y}(n_1,n_2)_{n_1,n_2\in[0,N-1]}=\frac{1}{2}\sum_{p_1=0}^{N/2-1}\sum_{p_2=0}^{N/2-1}x(p_1,p_2)e^{-j[2\pi p_1n_1/(N/2)]}e^{-j[2\pi p_2n_2/(N/2)]}$$

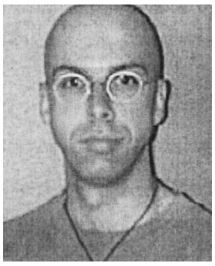
$$+\frac{1}{2}\sum_{p_1=0}^{N/2-1}\sum_{p_2=0}^{N/2-1}e^{-j[2\pi p_1n_1/(N/2)]}e^{j2p_1\pi}e^{-j[2\pi p_2n_2/(N/2)]}e^{j2p_2\pi},$$

$$=\tfrac{1}{2}\hat{x}(n_1,n_2)+\tfrac{1}{2}\hat{x}(n_1,n_2),$$

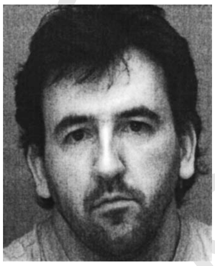
$$\hat{y}(n_1,n_2)_{n_1,n_2\in[0,N-1]}=\hat{x}(\widetilde{n_1},\widetilde{n_2})\quad\text{with}\quad\begin{cases}\widetilde{n_1} & = & n_1\bmod\dfrac{N}{2},\\[2ex] \widetilde{n_2} & = & n_2\bmod\dfrac{N}{2}.\end{cases}$$

References

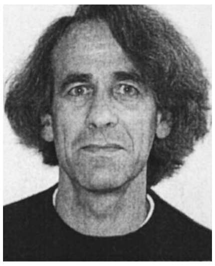
1. S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 674–693 (1989).
2. A. Cohen and I. Daubechier, "Nonseparable bidimensional wavelet bases," *Rev. Matematica Iberoamericana* **9**(1), 51–137 (1993).
3. J. Kovačević and M. Vetterli, "Nonseparable two- and three-dimensional wavelets," *IEEE Trans. Signal Process.* **43**, 1269–1273 (1995).
4. O. Rioul and D. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Trans. Inf. Theory* **38**, 569–586 (1992).
5. E. Viscito and J. P. Allebach, "The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattice," *IEEE Trans. Circuits Syst.* **38**, 29–41 (1991).
6. E. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *Proc. IEEE* **73**, 502–522 (1985).
7. Y. Kok, S. Lee, and H. Tan, "Periodic orthogonal splines and wavelets," *Appl. Comput. Harmon. Anal.* **2**, 201–218 (1995).
8. G. Plonka and M. Tasche, "On the computation of periodic spline wavelet," *Appl. Comput. Harmon. Anal.* **2**, 1–14 (1995).
9. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice–Hall, Englewood Cliffs, NJ (1995).
10. J.-C. Feauveau, "Analyse multirésolution avec un facteur de résolution $\sqrt{(2)}$," *J. Traitement Signal* **7**(2), 117–128 (1990).
11. M. Toubin, C. Dumont, F. Truchetet, and M. Abidi, "Multiresolution simplification of 3D multimodal objects using a 2D quincunx wavelet analysis," *Proc. SPIE* **3837**, 280–288 (1999).



Frédéric Nicolier received a master's degree in applied physics from Université de Bourgogne in 1995 and PhD degree in image processing at the same university in 2000. He is now an assistant professor at LAM, Université de Reims Champagne–Ardenne (URCA). His research interest includes wavelets transform, 3D problems and shape recognition.



Olivier Laligant received his PhD degree in 1995 from the Université de Bourgogne, France. He is an assistant professor in the Computing, Electronic, Imaging Departement (Le2i) at the Université de Bourgogne. His research interests are focused on multiscale edge detection, merging of data, wavelet transform and recently motion estimation.



Fred Truchetet received a master's degree in physics from Dijon University, France, in 1973 and PhD in electronics at the same university in 1977. He was with Thomson-CSF for two years as a research engineer and he is currently full professor and the head of the image processing group at Le2i, Laboratory of Electronic, Computer and Imaging Sciences, Université de Bourgogne and CNRS (FRE 2309), France. His research interests are focused on image processing for artificial vision inspection particularly on wavelet transform, multiresolution edge detection and image compression. He is member of SPIE and IEEE.