Signal Compression with Smooth Local Trigonometric Bases

Björn Jawerth* Yi Liu† Wim Sweldens‡

Abstract

We discuss smooth local trigonometric bases and their applications to signal compression. In image compression, these bases can reduce the blocking effect that occurs in JPEG. We present and compare two generalizations of the original construction of Malvar, Coifman and Meyer: biorthogonal and equal parity bases. These have the advantage that constant and linear components, respectively, can be represented efficiently. We show how they reduce blocking effects and improve the signal to noise ratio.

Keywords: compression, trigonometric bases, wavelets

1 Introduction

The general idea behind data compression is to remove the redundancy present in the data to find more compact representations. To achieve higher compression ratios, we give up on trying to represent the original exactly. Instead, we consider lossy compression, i.e. we allow approximate representations of the data which come "close" to the original.

A popular method for lossy compression of signals is so-called *transform coding*, see Figure 1. It involves representing the signal in a different basis, such that its coefficients in this basis are less correlated. The goal is for a small percentage of the coefficients to contain a large percentage of the information content. We obtain compression by only retaining the latter coefficients and setting the others to zero. An approximation of the original signal, which we will refer to as the *compressed signal*, can then be obtained by performing the inverse transform.

The most commonly used transform is the Fourier transform, or some variant of it. Its advantages are frequency localization, orthogonality, and the availability of fast numerical algorithms. Its main disadvantage is that the basis functions are non-local; think of sine and cosine functions, which stretch out over the whole domain. The correlation present in a signal, however, is mostly local. The Fourier basis, on the other hand, looks for correlation over the whole signal, which is usually small. As a consequence, the Fourier coefficients show little decorrelation.

We can get obtain more localization by splitting the signal into pieces, and perform the Fourier transform on each piece separately. We refer to the corresponding basis as the *local trigonometric basis*. This is a basic component of JPEG, the standard image compression algorithm [1]. Here the image is divided into blocks

*Partially supported by DARPA Grant AFOSR F49620-93-1-0083 and ONR Grant N00014-90-J-1343.

†Partially supported by DARPA Grant AFOSR F49620-93-1-0083.

‡Partially supported by NSF EPSCoR Grant EHR 9108772 and ONR Grant N00014-90-J-1343. This author is also Research Assistant of the National Fund of Scientific Research Belgium.

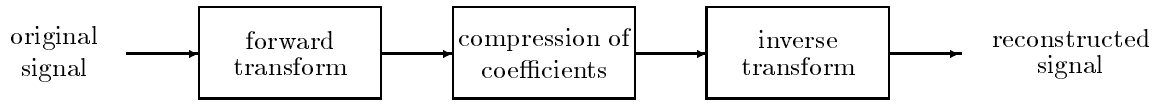

Figure 1: Transform coding.

of 8×8 pixels and a discrete cosine transform is used on each block. The local trigonometric basis, however, has some disadvantages:

1. Fourier like series are best suited for representing periodic signals, or signals with specific parity properties. Clearly, each piece does not necessarily satisfy those properties. This slows down the convergence and hinders high compression ratios.

2. Since each piece is processed individually, the compressed signal can reveal their geometry. In the case of JPEG image compression, this produces the well-known blocking effects.

3. Correlation among the pieces is not exploited.

An improvement was proposed by Coifman and Meyer [2], and by Malvar[3, 4]. Their idea is to use smooth cutoff functions to split the signal and to "fold" overlapping parts back into the pieces. This is done in a clever way such that the orthogonality is preserved and, moreover, that the folded signal is suited for representation by a trigonometric basis. In other words, it satisfies those specific properties. We refer to such a basis as a *smooth local trigonometric basis*. The basis functions are trigonometric functions multiplied with a smooth "bell-shaped" function, and they are closely related to wavelets. An approach based on smooth local trigonometric bases essentially solves the first two disadvantages described above. More specifically, it has been shown that this type of basis can reduce the blocking effect [5, 6]. An expository paper can be found in [7]. Also, a connection between this basis with the Wilson basis of [8] was pointed out in [9] .

The third disadvantage can be resolved by using an adaptive method where the splitting locations can depend on the signal. An algorithm was presented by Coifman and Wickerhauser [10, 11]. An alternative was proposed by Fang and Séré [12].

A disadvantage of smooth local trigonometric bases is that the resolution of the constants is lost, i.e. on each piece the DC component can no longer be encoded with one transform coefficient. In this paper, we present two generalizations of the construction of Coifman and Meyer that do have a resolution of the constants. These are called biorthogonal and equal parity bases.

The paper is organized as follows. In the first section, we discuss trigonometric bases and their properties. Then (Section 3) we consider the basis of Malvar, Coifman, and Meyer. We expand on the connection with wavelets in Section 4. In Section 5, we present the biorthogonal construction, while Section 6 contains a discussion of the equal parity case. Adaptive algorithms are treated in Section 7. Finally, we discuss implementation issues and give some results. Note that this paper does not contain proofs of the mathematical results. For a careful mathematical treatment, we refer to [13].

2 Trigonometric bases

Fourier series are the classic trigonometric basis. Consider the interval $I = [0,1]$, for simplicity. The basis functions are given by $e_k(x) = \exp(i2\pi kx)$, and we know that the set $\{\chi_I\, e_k \mid k \in \mathbf{N}\}$ is an orthogonal basis for $\mathrm{L}^2([0,1])$. Here χ_I is the indicator function on the interval I, i.e. the function that is one on I and zeros elsewhere. The space L^2 is the space of all square integrable functions. The Fourier series of a function is given by

$$f \;=\; \sum_k c_k\, e_k, \quad \text{with} \quad c_k \;=\; \int_0^1 f(x)\, \overline{e_k(x)}\, dx.$$

Its convergence depends on the smoothness of f when I is identified with the torus τ (i.e. the smoothness of the periodic extension of f). More precisely, if $f \in \mathcal{C}^p(\tau)$ (the periodic extension of f is p times continuously differentiable), then $c_k \;=\; \mathcal{O}(|k|^{-p})$. In other words, the coefficients decay rapidly when the function is smooth and periodic. Rapid decay is the key to compression, since it implies that only a few coefficients are needed to represent the signal within a certain accuracy.

However, the restriction of a smooth function to an interval is not necessarily a smooth function when extended periodically. This is due to the fact that the behavior of the functions at the left end of the interval does not necessarily match the behavior at the right end. In case f is smooth, but $f(0) \neq f(1)$, the coefficients decay very slow ($\mathcal{O}(|k|^{-1})$). In this case the Gibbs phenomenon occurs.

From the Fourier series, one can easily derive other orthogonal bases that only have sines or cosines as basis functions. One of them is the sine IV basis where $s_k(x) = \sin((2k+1)\pi x/2)$, and the set $\{\sqrt{2}\,\chi_I\, s_k \mid k \in \mathbf{N}\}$ is an orthogonal basis for $\mathrm{L}^2([0,1])$. It is called the sine IV basis because it uses basis functions that have quarter wavelengths as compared to the Fourier series. The s_k functions all are odd and smooth around the left endpoint and even and smooth around the right endpoint. As a consequence, the convergence of the representation in this basis will be fast whenever the function f, extended as an odd function around the left endpoint and as an even function around the right endpoint, is smooth. Other bases and their parities are: sine II (odd and odd), cosine II (even and even), and cosine IV (even and odd). For each basis a discrete transform and a fast algorithm exist [14, 15]. Whenever the signal has specific parity properties at the endpoints, it is important to choose a basis that reflects this property in order to achieve rapid decay of the coefficients.

3 Smooth local trigonometric bases

We define the *mirror operator* $\mathcal{M}_\alpha$ around a point α by

$$\mathcal{M}_\alpha\, f(x) \;=\; f(2\alpha - x).$$

This operator essentially flips the function around α.

Next, consider an interval of length 2ϵ around α and a continuous cutoff function l_α such that

$$l_\alpha(x) \;=\; \begin{cases} 1 & \text{if} \quad x < \alpha - \epsilon \\ 0 & \text{if} \quad x > \alpha + \epsilon, \end{cases},$$

and let $r_\alpha = \mathcal{M}_\alpha\, l_\alpha$. The *folding operator* around a point α is now defined by

$$\mathcal{F}_\alpha \;=\; \chi_\alpha^l\, (1 + \mathcal{M}_\alpha)\, l_\alpha + \chi_\alpha^r\, (1 - \mathcal{M}_\alpha)\, r_\alpha,$$

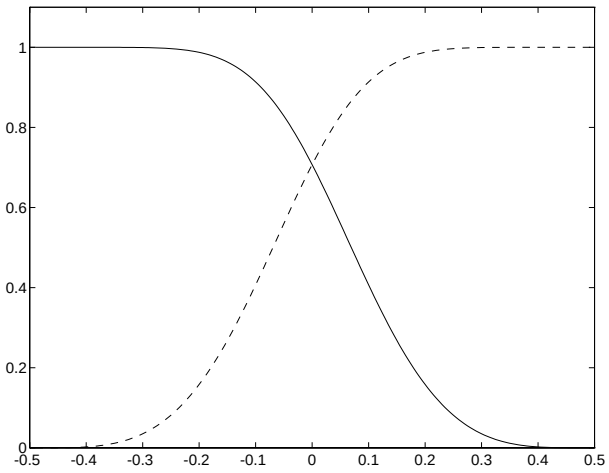

Figure 2: Cutoff functions l_0 and r_0 (dashed).

where $\chi^l_\alpha = \chi_{(-\infty,\alpha]}$ and $\chi^r_\alpha = \mathcal{M}_\alpha \chi^l_\alpha$. The adjoint of the folding operator is given by

$$\mathcal{F}^*_\alpha \;=\; \chi^l_\alpha \,(1 - \mathcal{M}_\alpha)\, l_\alpha + \chi^r_\alpha \,(1 + \mathcal{M}_\alpha)\, r_\alpha.$$

It is not hard to verify that the folding operator is unitary ($\mathcal{F}_\alpha \mathcal{F}^*_\alpha = 1$) if and only if $l^2_\alpha + r^2_\alpha \;=\; 1$. Figure 2 shows cutoff functions that satisfy this condition. For the remainder of this section we shall assume that it is satisfied.

Let us discuss what the action of the folding operator is. Multiplication with l_α lets the function die off smoothly to the left of α. The operator $1 + \mathcal{M}_\alpha$ then adds this function to its mirrored version, which results in a function that is even around α. This function is then cut off by χ^l_α. The right part is similar and creates an odd function. Hence, if f is smooth, then $\chi^l_\alpha \,\mathcal{F}_\alpha \, f$ is a function that is smooth when extended "even to the right" and $\chi^r_\alpha \,\mathcal{F}_\alpha \, f$ is a function that is smooth when extended "odd to the left." (By extending even (respectively odd) we mean applying the operator $1 + \mathcal{M}$ (respectively $1 - \mathcal{M}$).) The adjoint operator, which is also the inverse, does exactly the same except that it switches even and odd. Figure 3 shows the folding of an exponential function round $\alpha = 0$.

Consider now a partition of the real line into intervals $I = (n, n+1]$. The *total folding operator* is defined as

$$\mathcal{T} \;=\; \prod_n \mathcal{F}_n.$$

Being a product of unitary operators, it is also unitary. We can write the total folding operator as a sum of operators, each associated with one interval,

$$\mathcal{T} \;=\; \sum_n \chi_n \,\mathcal{G}_n.$$

Here $\chi_n = \chi_{(n,n+1]}$ and

$$\mathcal{G}_n \;=\; (1 - \mathcal{M}_n + \mathcal{M}_{n+1})\, b_n, \quad \text{with} \quad b_n \;=\; r_n \, l_{n+1}.$$

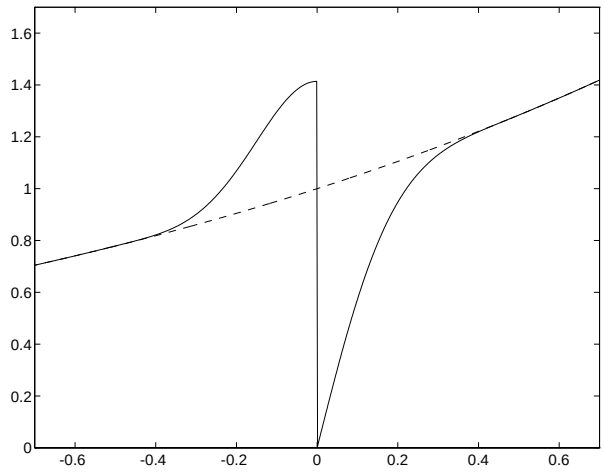

Figure 3: Folding with $\epsilon = 1/2$.

The function b_n is the *bell function* associated with the interval $(n, n+1]$. Note that

$$\mathcal{G}_n^* \;=\; b_n\,(1 - \mathcal{M}_n + \mathcal{M}_{n+1}) \quad \text{and} \quad \sum_l b_l^2 \;=\; 1.$$

Figure 4 shows an example of the bell function for $I = (0, 1]$ and $\epsilon = 1/2$.

We split L^2 into subspaces, such that each subspace contains functions localized around one of the intervals $(n, n+1]$. Moreover, we want each subspace to have a basis that is well-suited for representation of smooth functions. The easiest would be to just let

$$\mathrm{L}^2(\mathbf{R}) \;=\; \bigoplus_n \mathrm{L}^2((n, n+1]).$$

This obviously is an orthogonal decomposition, and χ_n is the orthogonal projection associated with it. As we mentioned in Section 2, the trigonometric bases on each interval are not suited for representations of smooth functions, unless they satisfy specific parity properties. However, if we first apply the total folding operator on a smooth function, we can create a function with these parity properties at the endpoints of each interval. We then use the trigonometric basis with the same parity properties. This results in a rapid decay of the coefficients in case the function is smooth. The orthogonality is preserved because the total folding operator is unitary. The orthogonal projection operator associated with an interval is then given by

$$\mathcal{P}_n \;=\; \mathcal{T}^*\,\chi_n\,\mathcal{T}.$$

We thus decompose $\mathrm{L}^2(\mathbf{R})$ into orthogonal subspaces V_n:

$$\mathrm{L}^2(\mathbf{R}) \;=\; \bigoplus_n V_n \quad \text{with} \quad V_n \;=\; \mathcal{P}_n\,\mathrm{L}^2(\mathbf{R}).$$

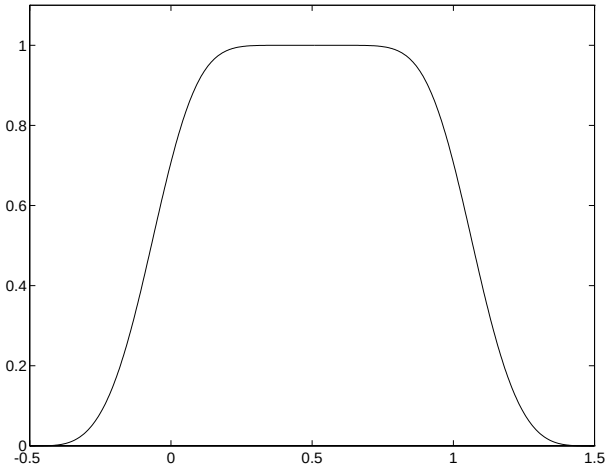

Figure 4: The bell function for $(0, 1]$.

It is possible to show that the projection operator can be written as

$$\mathcal{P}_n \;=\; \mathcal{G}_n^* \, \chi_n \, \mathcal{G}_n \;=\; b_n \, \mathcal{G}_n \;=\; b_n (1 - \mathcal{M}_n + \mathcal{M}_{n+1}) b_n .$$

The fact that the projection operators are orthogonal can also be understood as follows. In case $|n - m| > 1$, the supports of $\mathcal{P}_n f$ and $\mathcal{P}_m g$ do not intersect. In the other case, the function $\mathcal{P}_{n-1} f \; \mathcal{P}_n g$ is only supported on $[n - \epsilon, n + \epsilon]$, where it is equal to $l_n \, r_n \, d \, e$, where d is locally odd and e is locally even around n. Since $l_n \, r_n$ is locally even, the integral vanishes. (A function f is *locally even (respectively odd)* around a point α if $f(x) = \mathcal{M}_\alpha f(x)$ (respectively $-\mathcal{M}_\alpha f(x)$) for $x \in [\alpha - \epsilon, \alpha + \epsilon]$.)

It is possible to show [13] that every element of V_n is of the form $b_n \, s$, where s is locally odd around n and locally even around $n + 1$, and, conversely, every function of this form belongs to V_n. The orthogonal basis for $\mathrm{L}^2((n, n + 1])$ that matches this parity is given by

$$\chi_n \, s_{n,k}, \quad \text{with} \quad s_{n,k} \;=\; \sqrt{2} \sin(2k + 1) \pi (x - n)/2.$$

This immediately corresponds to an orthogonal basis for V_n given by

$$\mathcal{T}^* \, \chi_n \, s_{n,k} \;=\; \mathcal{G}_n^* \, \chi_n \, s_{n,k} \;=\; b_n \, s_{n,k}.$$

A general function f thus has a representation

$$f \;=\; \sum_n \mathcal{P}_n f \;=\; \sum_{n,k} c_{n,k} \, b_n \, s_{n,k},$$

where the coefficients are given by

$$c_{n,k} \;=\; \langle \, f, b_n \, s_{n,k} \, \rangle \;=\; \langle \, \mathcal{T} f, \chi_n \, s_{n,k} \, \rangle \, .$$

If we now look at the operators from an implementation point of view, we see that $\mathcal{F}_\alpha$ (and thus $\mathcal{T}$), χ_n and their adjoints are easy to discretize and implement. Because of this, we will use the second expression for the coefficients in the implementation. The transform algorithm consists of folding the signal and then using a sine transform on each interval.

4 Connection with wavelets

The basic idea behind wavelets is to use the translates and dilates of one function to form a basis. For details on wavelets and more references, we refer to [16, 17] and the other papers in this issue. It turns out that there is a close connection between local trigonometric functions and wavelets. To understand this better, we take a look at the following example. Consider the multiresolution analysis formed by the Shannon wavelet. Let

$$\psi(x) \;=\; \frac{\sin x}{x},$$

and

$$\psi_{j,l}(x) \;=\; \psi(2^j x - l).$$

Define

$$W_j \;=\; \operatorname{clos}\operatorname{span}\{\psi_{j,l} \mid l \in \mathbf{Z}\}.$$

Then

$$\bigoplus_j W_j \;=\; \mathrm{L}^2,$$

and the $\{\psi_{j,l}\}$ form an orthogonal basis for L^2. Since

$$\widehat{\psi}(\omega) \;=\; \chi_I(\omega) \quad \text{with} \quad I = [-2\pi, \pi] \cup [\pi, 2\pi],$$

we see that

$$W_j \;=\; \{f \in \mathrm{L}^2 \mid \operatorname{supp} \widehat{f} \subset 2^j I\}.$$

The splitting of L^2 into the wavelet spaces thus corresponds to splitting the frequency axis into logarithmically spaced intervals, and letting

$$\mathrm{L}^2 \;=\; \bigcup_j \mathrm{L}^2(2^j I).$$

On each interval we use the Fourier series basis since

$$\widehat{\psi}_{j,l} \;=\; \sqrt{2^{-j}} e^{-i\omega l 2^{-j}}\, \chi_{2^j I}.$$

These wavelets have slow decay. This is due to the fact that $\widehat{\psi}_{j,l}$ is not continuous and, as the reader recalls, decay in the spatial domain corresponds to smoothness in the frequency domain.

In any case, the $\hat{\psi}_{j,l}$ form a local trigonometric basis. Hence, it immediately follows that using the *smooth* local trigonometric basis on these intervals yields wavelets with rapid decay. In case the cutoff functions belong to $\mathcal{C}^\infty$, the wavelets have faster than polynomial decay. This is essentially the construction of the Meyer wavelet [18]. These wavelets have an infinite number of vanishing moments, since their Fourier transforms vanish in a neighborhood of the origin.

It was shown in [12] that more general splittings of the frequency axis correspond to wavelet packets in the spatial domain [19, 20].

5 Biorthogonal local trigonometric bases

We say that a basis has a *resolution of the constants* if, on every interval I, one of the basis functions is constant. If a basis has this property, the DC component can be encoded with one coefficient. This is important and leads to more compact representations since a smooth function locally resembles a constant. From the construction in the previous section we see that,

$$\chi_n \mathcal{T} 1 \;=\; \begin{cases} r_n - l_n & \text{on} \quad (n, n+\epsilon] \\ 1 & \text{on} \quad (n+1+\epsilon, n+1-\epsilon) \\ r_{n+1} + l_{n+1} & \text{on} \quad [n+1-\epsilon, n+1]. \end{cases}$$

It would be nice if we were able to find cutoff functions such that this function coincides with the first basis function $s_{n,0}$. This is clearly only possible if $\epsilon = 1/2$. In order to achieve a resolution of the constants and still have an orthogonal basis, we need to chose a left cutoff function of the form $l_n = l(x-n)$, where

$$l(x) \;=\; \frac{\cos(\pi x/2) - \sin(\pi x/2)}{\sqrt{2}} \quad \text{for} \quad x \in [-1/2, 1/2],$$

and $l(x)$ is one (respectively zero) to the left (respectively right) of this interval. This cutoff function is continuous, but it is not differentiable. Hence, the folding operator will introduce discontinuities in derivatives and this will slow down the decay of the coefficients and hinder compression. What we need is a resolution of the constants with smoother cutoff functions.

In order to solve this problem, we add some flexibility to the construction by abandoning the orthogonality requirement. In the remainder we let l_α and $r_\alpha = \mathcal{M}_\alpha l_\alpha$ be continuous cutoff functions that do not necessarily satisfy $l_\alpha^2 + r_\alpha^2 = 1$. We show [13] that in case $l_\alpha^2 + r_\alpha^2$ is bounded from below and above, the folding operator is bounded and invertible. Moreover, the inverse of an invertible folding operator $\mathcal{F}_\alpha$ again is a folding operator. More precisely, $\mathcal{F}_\alpha^{-1} = \widetilde{\mathcal{F}_\alpha}^{\,*}$ where

$$\widetilde{\mathcal{F}_\alpha} \;=\; \chi_\alpha^l \,(1 + \mathcal{M}_\alpha)\, \tilde{l}_\alpha + \chi_\alpha^r \,(1 - \mathcal{M}_\alpha)\, \tilde{r}_\alpha,$$

and

$$\tilde{l}_\alpha \;=\; \frac{l_\alpha}{l_\alpha^2 + r_\alpha^2} \quad \text{and} \quad \tilde{r}_\alpha \;=\; \frac{r_\alpha}{l_\alpha^2 + r_\alpha^2}.$$

Note that now $l_\alpha \tilde{l}_\alpha + r_\alpha \tilde{r}_\alpha \;=\; 1$. We call $\mathcal{F}_\alpha$ and $\widetilde{\mathcal{F}_\alpha}$ *biorthogonal folding operators* and refer to $\widetilde{\mathcal{F}_\alpha}$ as the *dual folding operator*.

This still does not solve the problem completely. The folded constant will have derivative zero at $1/2$ and can never coincide with the first basis function $s_{n,0}$. We need to generalize the construction further and allow different parities. We would like to have a folding operator that takes a smooth function into a function that is either odd at the left *and* right endpoint of an interval or even at both endpoints. One way to accomplish this would be to use folding operators with the same parity left and right of the split point. Unfortunately, such operators are not invertible [13].

Basically, this implies that the only way to get the same parity at both endpoints of an interval, is to alternate the parity of the folding operators. In other words, we need to change the parity from ... (odd – even) (odd – even) (odd – even) ... to ... (even – even) (odd – odd) (even – even) (odd – odd) .... The

corresponding total folding operator is obtained by alternating $\mathcal{F}_\alpha$ and $\mathcal{F}_\alpha^*$. In the intervals with (even even) parity we use the cosine II basis and in the intervals with (odd odd) parity the sine II basis. The total folding operator is then given by

$$\mathcal{T} \;=\; \prod_n \mathcal{F}_{2n}\,\mathcal{F}_{2n+1}^*.$$

This operator is invertible. The dual total folding operator is defined similarly (just add the tildes) and again $\mathcal{T}^{-1} = \widetilde{\mathcal{T}}^*$.

Following a reasoning similar to the one used in the orthogonal case, we see that the total folding operator, too, can be written as sum of folding operators associated with an interval:

$$\mathcal{T} \;=\; \sum_n \chi_n\,\mathcal{G}_n,$$

where

$$\mathcal{G}_{2n} \;=\; (1 - \mathcal{M}_{2n} - \mathcal{M}_{2n+1})\,b_{2n}, \quad \text{and} \quad \mathcal{G}_{2n+1} \;=\; (1 + \mathcal{M}_{2n+1} + \mathcal{M}_{2n+2})\,b_{2n+1}.$$

We may now define the operators $\mathcal{P}_n$ and the subspaces V_n as in the orthogonal case with some minor changes. The projection operator associated with an interval is given by

$$\mathcal{P}_n \;=\; \widetilde{\mathcal{T}}^*\,\chi_n\,\mathcal{T}.$$

The projection operator is given by

$$\mathcal{P}_n \;=\; \widetilde{\mathcal{G}}_n^*\,\chi_n\,\mathcal{G}_n \;=\; \widetilde{b}_n\,\mathcal{G}_n \;=\; \widetilde{b}_n(1 \pm \mathcal{M}_n \pm \mathcal{M}_{n+1})b_n.$$

An element of V_n is equal to $\widetilde{b}_n$ times a function that is locally even (respectively odd) around the endpoints of the interval in case n is odd (respectively even). We now use the basis functions with the right parity on each interval:

$$t_{2n,k} \;=\; \sqrt{2}\sin(k+1)\pi(x-2n) \quad \text{for} \quad k > 0,$$

$$t_{2n+1,k} \;=\; \sqrt{2}\cos k\pi(x-2n-1) \quad \text{for} \quad k > 1, \quad \text{and} \quad t_{2n+1,0} \;=\; 1.$$

Obviously, the $\chi_n\,t_{n,k}$ with $n \in \mathbf{Z}$ and $k \in \mathbf{N}$ form an orthogonal basis for L^2. This implies that the basis formed by the $\widetilde{\mathcal{T}}^*\,\chi_n t_{n,k}$ is a Riesz basis for L^2. These functions are given by

$$\widetilde{\mathcal{T}}^*\,\chi_n\,t_{n,k} \;=\; \widetilde{\mathcal{G}}_n^*\,\chi_n\,t_{n,k} \;=\; \widetilde{b}_n\,t_{n,k}.$$

The representation of a general function f takes the form

$$f \;=\; \sum_n \mathcal{P}_n f \;=\; \sum_{n,k} c_{n,k}\,\widetilde{b}_n\,t_{n,k},$$

where the coefficients are given by

$$c_{n,k} \;=\; \langle\, f, b_n\,t_{n,k}\,\rangle \;=\; \langle\, f, \mathcal{T}^*\,\chi_n\,t_{n,k}\,\rangle \;=\; \langle\, \mathcal{T}f, \chi_n\,t_{n,k}\,\rangle\,.$$

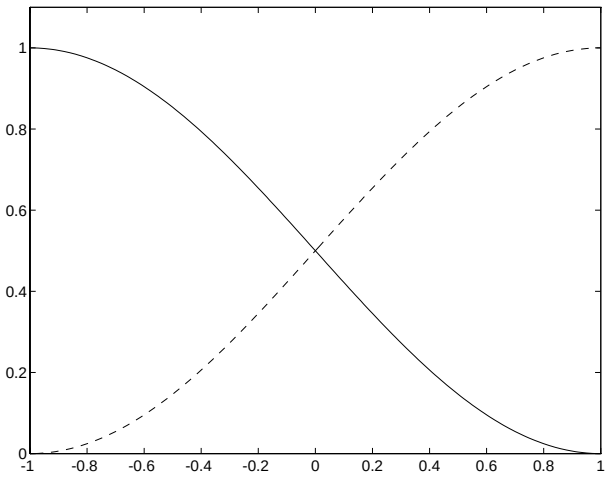

Figure 5: The biorthogonal cutoff functions.

We say that $\{b_n\, t_{n,k}\}$ is the dual basis corresponding to the basis $\{\widetilde{b}_n\, t_{n,k}\}$. Again, the last expression for the coefficients is the easiest to implement. The transform algorithm consists of folding and a sine or cosine transform.

Now, the only thing left is to find cutoff functions such that $\chi_n\, \mathcal{T}\, 1$ coincides (up to a constant factor) with $\chi_n\, t_{n,0}$. This can be done by letting $l_n = l(x-n)$ where

$$l(x) \;=\; \frac{1-\sin(\pi x)}{2} \quad \text{for} \quad x \in [-1/2, 1/2],$$

and $l(x)$ is one (respectively zero) to the left (respectively right) of this interval. This cutoff function is differentiable. It is easy to check that on I, $\mathcal{G}_{2n}\, 1 = \sin(\pi(x-2n))$ and $\mathcal{G}_{2n+1}\, 1 = 1$. The cutoff functions are shown in Figure 5 and the biorthogonal total folding of a function is shown in Figure 6. Note how the folded function on each interval already closely resembles the first basis function. The condition number of the folding operator is $\sqrt{2}$, and this implies that the numerical computation is stable.

6 Equal parity bases

In this section, we take a closer look at the folding operator that takes a smooth function into a function with the same parity left and right of the folding point. We call such a folding operator an *equal parity folding* operator. It is possible to show [13] that these operators are not invertible in L^2. Nevertheless, they have been used successfully for image compression [5]. We study their behavior in more detail in this section.

We shall omit the subscript n for simplicity. The equal parity folding operator with (even – even) parity is given by

$$\mathcal{F} \;=\; \chi^l\,(1+\mathcal{M})\,l + \chi^r\,(1+\mathcal{M})\,r, \tag{1}$$

where l and r are smooth cutoff functions with $r = \mathcal{M}\, l$. We immediately see that the folding operator commutes with $\mathcal{M}$ and that it maps even (respectively odd) functions into even (respectively odd) functions.

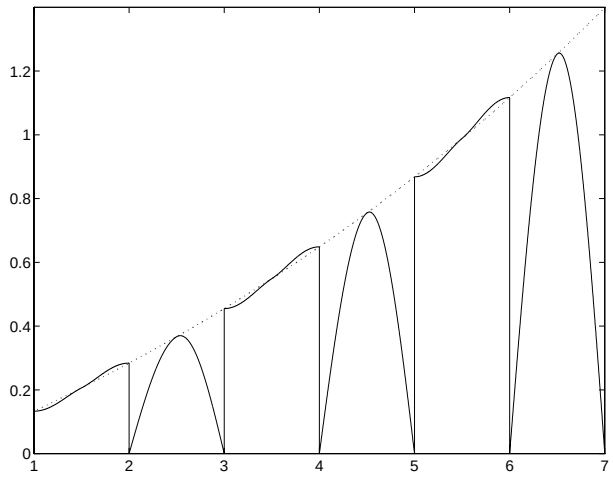

Figure 6: The biorthogonal total folding.

For an even function, $\mathcal{F}$ coincides with multiplication by $l + r$, while for an odd function $\mathcal{F}$ coincides with multiplication by $(\chi^l - \chi^r)\,(l - r)$. This implies that we can write $\mathcal{F}$ as

$$\mathcal{F} \;=\; (l + r)\,(1 + \mathcal{M})/2 + (l - r)\,(\chi^l - \chi^r)\,(1 - \mathcal{M})/2. \tag{2}$$

The folding operator is clearly bounded. However, it is not invertible on its range. The inverse operator, at least formally, is again an equal parity folding operator with the parity (even – even). We denote it by $\widetilde{\mathcal{F}}$, where

$$\widetilde{\mathcal{F}} \;=\; \chi^l\,(1 + \mathcal{M})\,\widetilde{l} + \chi^r\,(1 + \mathcal{M})\,\widetilde{r},$$
$$\widetilde{l} \;=\; \frac{l}{l^2 - r^2}, \quad \text{and} \quad \widetilde{r} \;=\; \frac{r}{r^2 - l^2}.$$

Again formally, we have that $\widetilde{\mathcal{F}}^*\,\mathcal{F} = 1$. The fact that $\mathcal{F}$ is not invertible shows up in the singularities of the dual cutoff functions at the origin. We note that $\mathrm{L}^2 \subset \text{range}\,\widetilde{\mathcal{F}}$. An example of a function that $\widetilde{\mathcal{F}}$ takes out of L^2 is a function with a discontinuity across the folding point. We also note that $\mathcal{F}$ has certain smoothing properties; more precisely, it maps a function with a discontinuity at the origin into a continuous function because $\mathcal{F}\,\chi^l \;=\; l$.

Let us now discuss how these operators can be used in signal compression. The idea would again be to construct a total folding operator and then use a trigonometric basis with the right parity for each interval. In this case, the appropriate basis is the cosine II basis. The compression would be carried out in this basis, after which we use the inverse total folding operator to reconstruct the signal. Now, if we think of $\mathcal{F}$ as some kind of a smoothing operator and of $\widetilde{\mathcal{F}}$ as an operator that can blow up discontinuous functions, it makes sense to use $\widetilde{\mathcal{F}}$ to construct the total folding operator and $\mathcal{F}$ for its inverse. We remark that the idea to switch the two operators around was already suggested in [5]. This has the following advantages:

1. Errors introduced by the compression in the trigonometric basis cannot get blown up since $\mathcal{F}$ is bounded.

2. Discontinuities across the folding points will get smoothed by $\mathcal{F}$. In other words, the blocking effect will be reduced.

The disadvantage is that the $\widetilde{\mathcal{F}}$ operator blows up functions that have discontinuities at the split point. In this case, the folding operator is ill-conditioned and the numerical computations become unstable. However, their are very few images where the edges exactly coincide with the split points; think of a chess board as the worst case.

The construction of the total folding operators now is analogous to the biorthogonal case. On each interval we use the cosine II basis or the functions $\chi_n C_{n,k}$ where $C_{n,k} = \sqrt{2}\cos k\pi(x-n)$. We remark that the $\mathcal{F}\chi_n C_{n,k}$ cannot generate a basis for L^2, but that they merely form a set whose linear span is dense. It is still true that $\mathcal{F}\chi_n C_{n,k} = b_n C_{n,k}$, where b_n is the usual bell function.

It is clear that we get a resolution of the constants in case $l + r = 1$. We now still have one degree of freedom left in the choice of $l - r$. We can use this to also obtain a resolution of the linears, i.e. a representation of each linear function by two basis functions on each interval, if we let

$$l(x) = 1/2 + \frac{x}{1 - 2/\pi\cos(\pi x)} \quad \text{for} \quad x \in [-1/2, 1/2].$$

Figure 7 shows the equal parity folding of the function x with folding points 0 and 1. We see that on the interval [0,1] it coincides with a function of the form $A + B\cos(\pi x)$ (dashed). The constant and linear

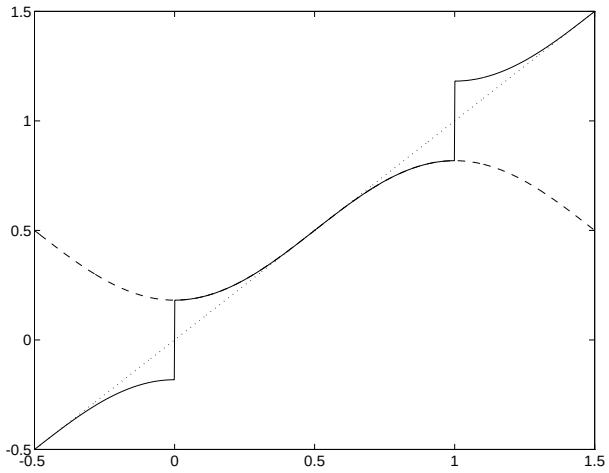

Figure 7: Equal parity folding of a linear function.

component can thus be encoded with two coefficients on each interval. If we eliminate these components from the function, the remainder typically looks like a waveform, which can be represented efficiently by the higher frequency basis functions.

7 Adaptive algorithms

So far we have taken the splitting locations, and thus the pieces in which we split the signal, to be fixed. As we pointed out in the introduction, this has as disadvantage that we cannot exploit correlation among the pieces. In an *adaptive* algorithm, the splitting location, at least to a certain extent, are determined by the signal. In this way we hope to be able to achieve higher compression. In this section we shall restrict ourselves to the orthogonal case, although the ideas can be generalized to the biorthogonal and equal parity case.

The construction of the algorithm relies on the following observation: if $K = I \cup J$, then

$$\mathcal{P}_K \;=\; \mathcal{P}_I + \mathcal{P}_J,$$

where the sum corresponds to an orthogonal decomposition. This follows from the fact that the folding operator is unitary. This implies that we can easily switch from a basis on the interval K to bases on I and J. An adaptive algorithm now uses a certain criterion in deciding whether to use K or I and J in the representation.

Let us be more precise and discuss the *best basis selection algorithm* from [10]. It uses a splitting of the real line into dyadic intervals

$$I_{j,k} \;=\; (2^{-j}k, 2^{-j}(k+1)] \quad \text{with} \quad 0 \leqslant j \leqslant n,\, k \in \mathbf{Z}.$$

We have that

$$I_{j,k} \;=\; I_{j+1,2k} + I_{j+1,2k+1}.$$

Each of these sums represents a branching in a full binary tree. The algorithm starts with the smooth local trigonometric basis on the finest level (largest n) and works towards the coarsest level (smallest n, root of the tree). On each branch it decides whether or not to join the two intervals. This decision is based on which representation shows most decorrelation. One method to measure the decorrelation is entropy [10].

This algorithm has the following drawback. One can show that the entropy of a coefficients depends on the product of the derivative of the bell function and the length of the interval [11]. Thus this product should be constant. In other words, the wider the bell function, the slower it should die off at the edges. Stated more precisely, this means that ϵ_α and ϵ_β, the influence regions of the folding operators around α and β, should be proportional to $|I|$. The algorithm described above, however, does not have this property. The reason is that the influence region ϵ_α of the folding operator $\mathcal{F}_\alpha$ is limited by the requirement that $\beta - \alpha \;\geqslant\; \epsilon_\alpha + \epsilon_\beta$. This implies that if a large interval is surrounded by two smaller intervals, the epsilons of the large interval have to be small.

To overcome this problem we introduce the *multiple folding technique* developed by Fang and Séré [12]. Consider the set $A = \{\alpha\}$ of all folding points and split it into a finite number of disjoint subsets,

$$A \;=\; \bigcup_{j=0}^{n} A_j.$$

The total folding operator can then be written as

$$\mathcal{T} \;=\; \prod_j \mathcal{T}_i \quad \text{with} \quad \mathcal{T}_j \;=\; \prod_{\alpha \in A_j} \mathcal{F}_\alpha.$$

If we give up the commutativity of the $\mathcal{T}_j$ operators, we gain the following flexibility. When applying $\mathcal{T}_0$, the ϵ's can be chosen large since the neighboring folding points are more distant. When applying $\mathcal{T}_1$, the ϵ's are determined by the neighboring split points in $A_0 \cup A_1$, and so on. Finally, for $\mathcal{T}_n$, the epsilons are back to what they used to be. Clearly, the A_i should be chosen in such a way to make maximal use of this flexibility. For example, in the algorithm described above, we have

$$A \;=\; 2^{-n}\,\mathbf{Z}, \quad A_0 = \mathbf{Z},$$

and

$$A_j \;=\; 2^{-j}\,(2\mathbf{Z}+1) \quad (j>0).$$

Consequently, we can take $\epsilon = 2^{-j-1}$ in $\mathcal{T}_j$.

For a thorough treatment of alternative adaptive wavelet techniques, we refer to [21, 22, 23, 24].

8 Implementation

Let us start by sketching an analogue optical implementation. Clearly, the mirror operator can easily be implemented. The same holds for addition and subtraction. Multiplication with the cutoff functions can be done with a lens whose transparency corresponds to the cutoff functions. The sine and cosine transforms can be written in terms of the Fourier transform, which can be implemented with one convex lens, see [25, 26, 27]. For a discussion of the optical implementation of the wavelet transform and many more references, we refer to [28]. However, analogue implementations usually suffer from a lack of accuracy and alignment problems. We therefore focus on discrete implementations.

So far the discussion only involved functions of a continuous variable. In a discrete implementation, a function f is given as a sequence $\{f_n\}$ where the "samples" f_n can be seen as pointwise evaluations of f on a regular grid in case f is continuous, or as average values of f in a neighborhood of the grid point if not. We assume the former. Take equidistant sample points $\{x_n\}$. We then have

$$f_n \;=\; f(x_n), \quad \text{with} \quad x_{n+1} - x_n \;=\; h,$$

where h is the discretization parameter.

We first need to decide whether we want to use a staggered or non-staggered discretization. In a *non-staggered discretization*, the boundaries of the interval coincide with a grid point, $x_n = nh$, while in a *staggered discretization* the boundaries of the interval fall between grid points, $x_n = (n+1/2)h$.

In the orthogonal construction both discretizations are possible. The fact that a folded function is discontinuous at the folding point does not pose a problem in the non-staggered discretization. At the folding point we only need the value of the "even" part since we know that the "odd" part vanishes. In the biorthogonal case both options still can be used. In this case, the non-staggered has the disadvantage that the "even-even" intervals contain two more samples than the "odd-odd" intervals. This makes implementation harder. In the equal parity case one has to use the staggered implementation since some of the cutoff functions have a singularity at the folding point. This makes it possible to implement operators that, in the L^2 sense, are unbounded or not invertible. The fact that the range of $\mathcal{F}$ is dense in L^2 ensures that these discrete operators are invertible. However, this unavoidably results in ill-conditioned discrete operators.

Fast algorithms are available for discrete versions of the sine and cosine transforms [14, 15]. In the biorthogonal case, one needs the alternate the discrete cosine transform (DCT) and discrete sine transform

over the pieces. In the equal parity case, we only need the DCT, which makes the implementation easier. In image compression, the folding can then be seen as a preprocessing step for JPEG. The advantage of a bases that has a resolution of the constants (linears), is that one can still use difference coding for the DC (linear) component.

A software implementation is now straightforward. But also hardware implementations, both optical and electronic, are possible. The folding operator is easy to implement. The discrete cosine and sine transforms can be rewritten in terms of the discrete Fourier transform. Implementations of the FFT are readily available.

9 Results

A transform coding scheme does not only involve a transform, but also a quantization and encoding step. However, we will not enter into these issues here. Instead, we include some results obtained by simply retaining the coefficients above a certain threshold and setting the others to zero ("clipping"). The compression ratio is then defined as the number of data samples divided by the number of transform coefficients that were retained. We are currently in the process of designing quantization and encoding schemes specific to the transform used.

We include an example concerning image compression, where we compare the use of smooth local trigonometric bases with JPEG. Figures 8 and 9 give the peak signal to noise ratio (PSNR) in function of the compression, for two 512×512 grey value images (Lena and Peppers). Figures 10 and 11 give the mean square error (MSE) in function of the compression, for the same images. We see that the smooth local trigonometric bases are better than JPEG. For small compression ratios, the biorthogonal basis outperforms the equal parity one, while for higher compression the situation is reversed. In Figures 9 and 13 (detail) we show the original and compressed (18:1) images for Lena. We note that the biorthogonal basis is the best at reducing the blocking effect.

References

[1] G. K. Wallace. The JPEG still picture compression standard. *Comm. ACM*, 34(4):30–44, 1991.

[2] R. Coifman and Y. Meyer. Remarques sur l'analyse de Fourier à fenêtre. *C. R. Acad. Sci. Paris Sér. I Math.*, I(312):259–261, 1991.

[3] H. S. Malvar. Lapped transforms for efficient transform/subband coding. *IEEE Trans. Acoust. Speech Signal Process.*, 38:969–978, 1990.

[4] H. S. Malvar and D. H. Staelin. The LOT: Transform coding without blocking effects. *IEEE Trans. Acoust. Speech Signal Process.*, 37:553–559, 1989.

[5] G. Aharoni, A. Averbuch, R. Coifman, and M. Israeli. Local cosine transform — A method for the reduction of the blocking effect in JPEG. *J. Math. Imag. Vision*, 3:7–38, 1993.

[6] G. Aharoni, A. Averbuch, R. Coifman, and M. Israeli. Local cosine transform — A method for the reduction of the blocking effect in JPEG. In A. F. Laine, editor, *Mathematical Imaging: Wavelet Applications in Signal and Image Processing, San Diego, CA*, pages 205–217. SPIE Proceedings Series volume 2034, 1993.

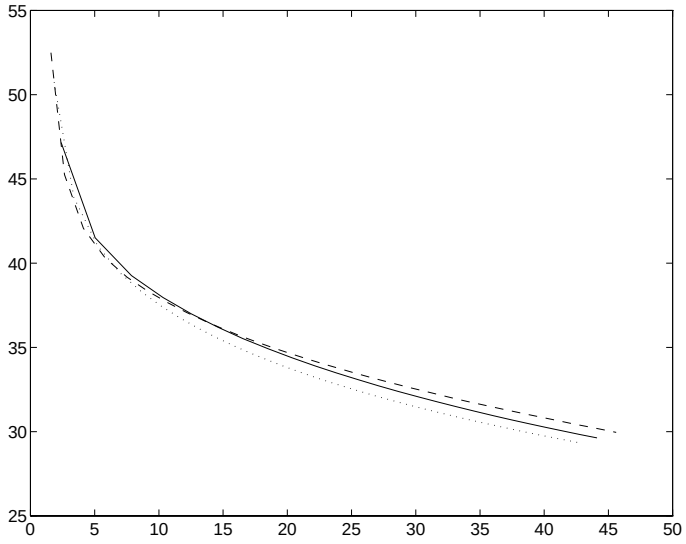

Figure 8: PSNR for Lena.
(JPEG = dotted, biorthogonal = full, equal parity = dashed)

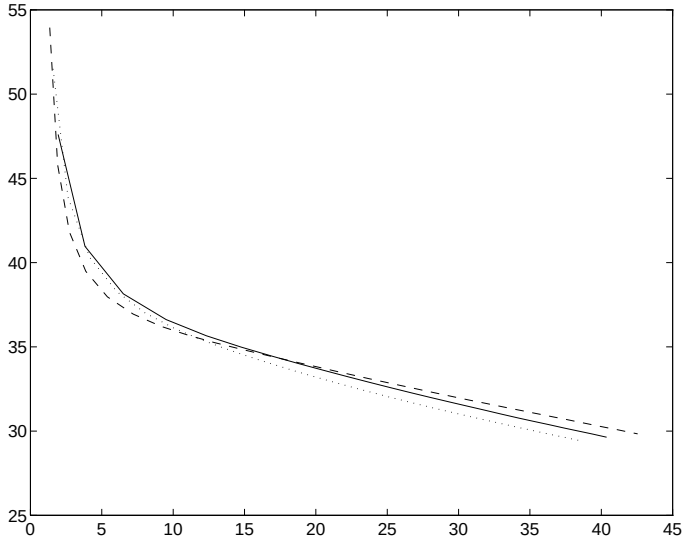

Figure 9: PSNR for Peppers.
(JPEG = dotted, biorthogonal = full, equal parity = dashed)

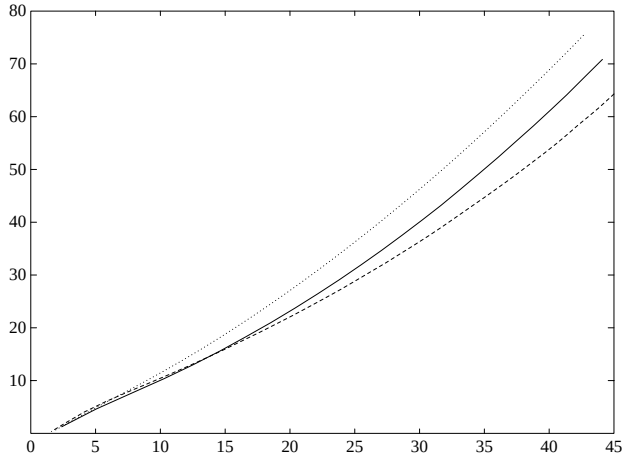

Figure 10: MSE for Lena.
(JPEG = dotted, biorthogonal = full, equal parity = dashed)

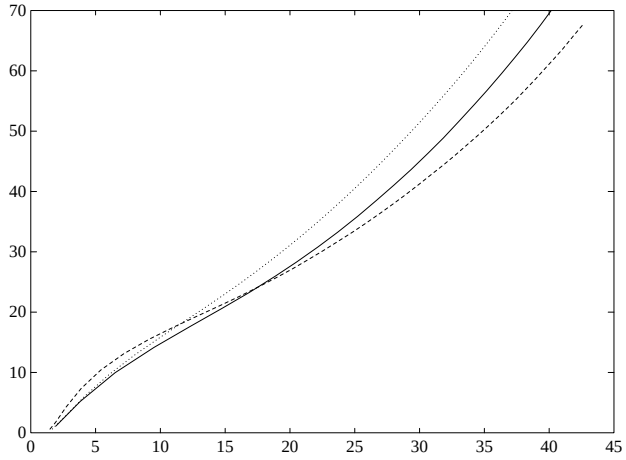

Figure 11: MSE for Peppers.
(JPEG = dotted, biorthogonal = full, equal parity = dashed)

[7] P. Auscher, G. Weiss, and V. Wickerhauser. Local sine and cosine bases of Coifman and Meyer and the construction of smooth wavelets. In [29], pages 237–256.

[8] I. Daubechies, S. Jaffard, and J.-L. Journé. A simple Wilson orthonormal basis with exponential decay. *SIAM J. Math. Anal.*, 22:554–572, 1991.

[9] P. Auscher. Remarks on the local Fourier bases. In [30], pages 203–218.

[10] R. R. Coifman and M. L. Wickerhauser. Entropy based algorithms for best basis selection. *IEEE Trans. Inform. Theory*, 38(2):713–718, 1992.

[11] M. V. Wickerhauser. Lectures on wavelet packets algorithms. 1991.

[12] X. Fang and E. Séré. Adaptive multiple folding local trigonometric transforms and wavelet packets. *Appl. Comput. Harmon. Anal.*, 1(2), 1994.

[13] B. Jawerth and W. Sweldens. Biorthogonal smooth local trigonometric bases. *J. Fourier Anal. Appl.*, 2(1), 1995. (to appear), (`ftp://ftp.math.scarolina.edu/pub/imi_94/imi94_5.ps`).

[14] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes.* Cambridge University Press, 2nd edition, 1993.

[15] J. S. Walker. *Fast Fourier transforms.* Studies in Advanced Mathematics. CRC Press, Boca Raton, 1991.

[16] I. Daubechies. *Ten Lectures on Wavelets.* CBMS-NSF Regional Conf. Series in Appl. Math., Vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[17] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36(3):377–412, 1994.

[18] Y. Meyer. *Ondelettes et Opérateurs,* I: *Ondelettes,* II: *Opérateurs de Calderón-Zygmund,* III: (with R. Coifman), *Opérateurs multilinéaires.* Hermann, Paris, 1990. English translation of first volume, *Wavelets and Operators*, is published by Cambridge University Press, 1993.

[19] R. R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser. Signal processing and compression with wave packets. In Y. Meyer, editor, *Proceedings of the International Conference on Wavelets, Marseille, 1989.* Masson, Paris, 1992.

[20] R. R. Coifman, Y. Meyer, and V. Wickerhauser. Size properties of wavelet packets. In [31], pages 453–470.

[21] H. Szu, B. Telfer, and S. Kadambe. Neural network adaptive wavelets for signal representation and classification. *Opt. Eng.*, 31(9):1907–1916, 1992.

[22] H. Szu, X.-Y. Yang, B. Telfer, and Y. Sheng. Neural network and wavelet transform for scale-invariant data classification. *Phys. Rev. E*, 48(2):1497–1501, 1993.

[23] H. Szu, Y. Sheng, and J. Chen. Wavelet transform as a bank of matched filters. *Appl. Opt.*, 31(17):3267–3277, 1992.

[24] Y. Sheng, D. Roberge, and H. Szu. Optical wavelet matched filter for shift-invariant pattern recognition. *Opt. Letters*, 18(4):299–301, 1993.

[25] I. Andonovic and D. Uttamchandani. *Modern optical systems.* Artech House, Norwood MA, 1989.

[26] D. G. Feitelson. *Optical Computing.* MIT Press, 1988.

[27] A. D. McAulay. *Optical Computer Architectures.* John Wiley & Sons, 1991.

[28] A. Arneodo, F. Argoul, J. F. Muzy, B. Pouligny, and E. Freysz. The optical wavelet transform. In [31], pages 241–273.

[29] C. K. Chui, editor. *Wavelets: A Tutorial in Theory and Applications.* Academic Press, San Diego, CA, 1992.

[30] J. Benedetto and M. Frazier, editors. *Wavelets: Mathematics and Applications.* CRC Press, Boca Raton, 1993.

[31] M. B. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael, editors. *Wavelets and their Applications.* Jones and Bartlett, Boston, 1992.

Björn Jawerth
University of South Carolina
Department of Mathematics
Columbia SC 29208
bj@math.scarolina.edu

Yi Liu
University of South Carolina
Department of Mathematics
Columbia SC 29208
yliu@math.scarolina.edu

Wim Sweldens
University of South Carolina
Department of Mathematics
Columbia SC 29208
and
Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A
B 3001 Leuven Heverlee
Belgium
sweldens@math.scarolina.edu

Figure 12: Original and compressed images for Lena.
(top left: original, top right: JPEG,
bottom left: equal parity, bottom right: biorthogonal)

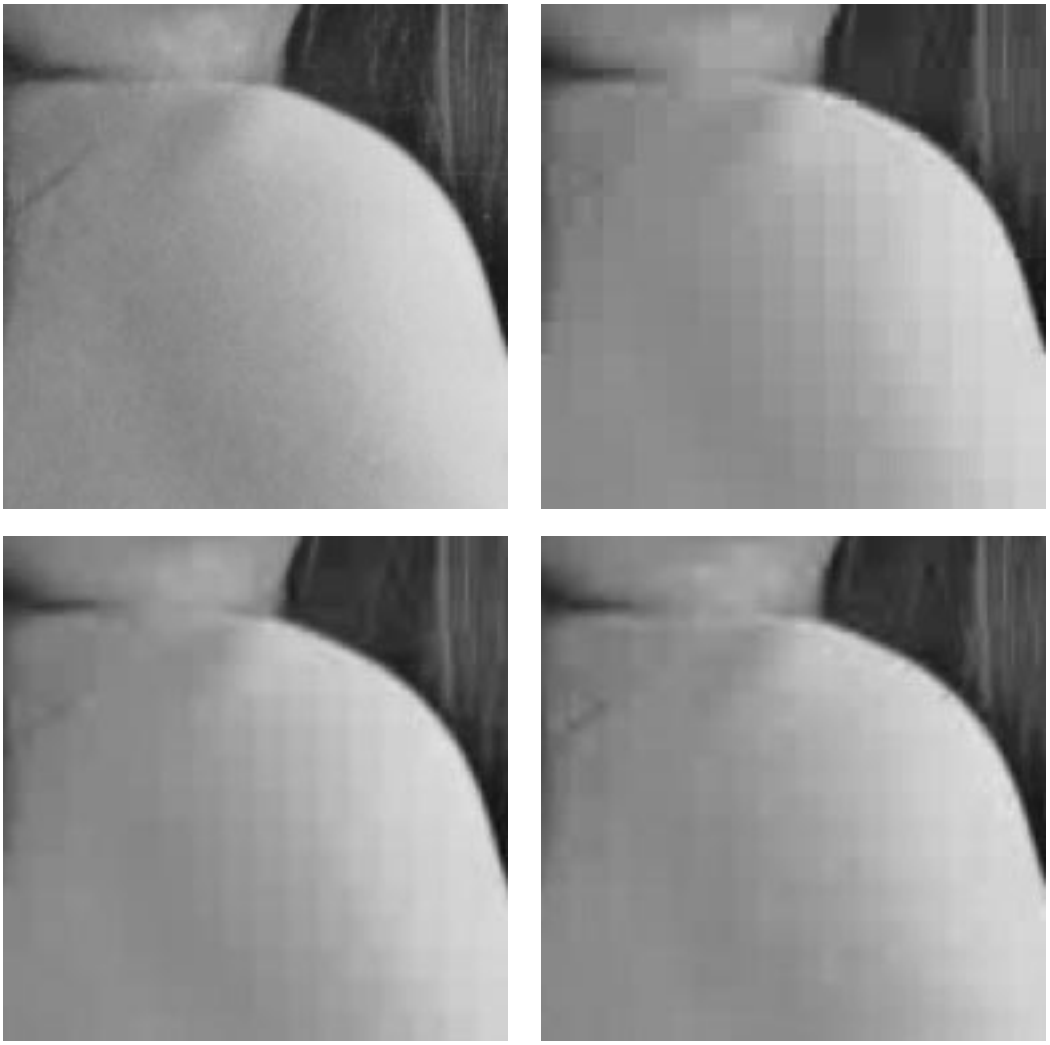

Figure 13: Original and compressed images for Lena (detail of the shoulder).
(top left: original, top right: JPEG,
bottom left: equal parity, bottom right: biorthogonal)