Multiwavelets and Integer Transforms

Patrick J. Van Fleet*

Abstract

In many applications of image processing, the given data are integer-valued. It is therefore desirable to construct transformations that map data of this type to an integer (or rational) ring. Calderbank, Daubechies, Sweldens, and Yeo [1] devised two methods for modifying orthogonal and biorthonal wavelets so that they map integers to integers.

The first method involves appropriately scaling the transform so that data that has been transformed and truncated can be recovered via the inverse wavelet transform. In developing this method, the authors of [1] created a useful factorization of the 4-tap Daubechies orthogonal wavelet transform [2]. We have observed that this factorization can be extended to 4-tap multiwavelets of arbitrary size.

In this paper we will discuss this generalization and illustrate the factorization on two multiwavelets. In particular, the well-known Donovan, Geronimo, Hardin, and Massopust (DGHM) [3] multiwavelet transform can be scaled so that it maps integers to integers. Since this transform is (anti)symmetric in addition to orthogonal, regular, and compactly supported, the ability to modify it so that it maps integers to integers should be useful in image processing applications.

AMS(MOS) 1991 subject classification: 42A10,42A15.
Keywords and Phrases: Wavelets, multiwavelets, polyphase matrix, integer transforms.

1 Introduction

The data that comprise a digital image are often integer valued. A typical image processing algorithm (for tasks such as compression and enhancement) is as follows: (1) transform the data, (2) apply some decision rule, (3) reconstruct the image using the inverse transform. It is desirable that this algorithm return integer-valued data. In the case of many (bi)orthogonal wavelet transforms, the transformed data are not integer valued. Thus once a decision rule is applied, it is likely that the inverse transform of the resultant is not integer-valued.

A popular method for mapping integers to integers is described in [6] where the authors

*Dept. of Mathematics, University of St. Thomas, St. Paul, MN 55105, pvf@pascal.math.stthomas.edu

introduce an implicitly-defined perturbation of the data. This perturbation guarantees that the transformed data will be integer-valued. As shown in [1], this technique cannot be used with wavelet transforms. As alternatives, the authors of [1] give two methods for adapting wavelet transforms so that they map integers to integers. The first involves appropriately scaling the transform so that inverted data can be uniquely rounded/truncated to integer values and the second utilizes the lifting scheme due to Sweldens [11].

In this paper, we will generalize the first method to include 4-tap multiwavelets. In order to appropriately scale the wavelet transform, the authors of [1] devise a factorization of the transform. Of particular interest is Daubechies' orthogonal 4-tap wavelet [2]. The authors show that the matrix representation $\boldsymbol{H}$ of this transform can be factored as:

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{P}\boldsymbol{V}^T$$

where $\boldsymbol{U}$, $\boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{P}$ is a permutation matrix. Moreover the orthogonal matrices $\boldsymbol{U}$, $\boldsymbol{V}$ are block diagonal with the same 2×2 orthogonal matrices U, V comprising the diagonal. $\boldsymbol{P}$ leaves the even entries of a vector unchanged and permutes the odd ones. The advantage of this factorization is that the data can be decoupled into 2×2 blocks and the result of applying $\boldsymbol{H}$ can easily be analyzed. This in turn leads to an appropriate scaling factor that guarantees a uniquely defined integer valued inverse.

A natural question to ask is whether the same factorization exists for 4-tap multiwavelets. Since it is known that 4-tap orthogonal multiwavelets can be constructed so that they are regular and (anti)symmetric (see [3]), the ability to map integers to integers would be an additional benefit. In this paper, we will derive a factorization of the DGHM [3] multiwavelet and show how it can be used to determine an appropriate scaling value. We will apply this transform to a standard test image and compare the results to that of other wavelet transforms. We will also illustrate our factorization on a multiwavelet of size $A = 3$ due to Pan, Jiao, and Yangwang [8].

The remainder of the paper is organized as follows. The next section will contain definitions, examples, and results necessary in subsequent sections. Section 3 will illustrate the factorization of the DGHM multiwavelet and the multiwavelet from [8]. The final section will be comprised of a comparison of the integer-to-integer DGHM multiwavelet and other wavelet transformations.

2 Preliminary Results

In this section we will primarily review the results in [1] necessary to the sequel. We will also provide a brief background on multiwavelets and discuss the DGHM multiwavelet.

In the theory of wavelets, we typically start with a *scaling function* ϕ that satisfies a *refinement equation*

$$\phi(x) = \sqrt{2} \sum_{k=0}^{2N-1} h_k \phi(2x - k). \tag{1}$$

That is, ϕ can be written in terms of functions $\phi(2\cdot -k)$ on a finer level. In order to obtain an orthogonal transformation, we insist that ϕ is orthogonal to its integer translates, i.e., $(\phi(x-m),\phi(x-n))=\delta_{mn}$. The fact that the refinement equation is comprised of finitely many terms will guarantee that ϕ is compactly supported.

Let $V_0=\overline{\mathrm{span}\{\phi(\cdot-k)\}_{k\in\mathbb{Z}}}$ and $V_1=\overline{\mathrm{span}\{\sqrt{2}\phi(2\cdot-k)\}_{k\in\mathbb{Z}}}$. Using (1), we see that $V_0\subset V_1$. We define the space W_0 as the orthogonal complement of V_0 in V_1, i.e., $V_0\oplus W_0=V_1$ (see [2] for a complete treatment of multiresolution analyses in $L^2(\mathbb{R})$). It can be shown that W_0 can be realized as the closed linear span of the integer translates of the *wavelet* $\psi(x)$, where ψ also satisfies a dilation equation:

$$\psi(x)=\sum_{k=0}^{2N-1}g_k\phi(2x-k). \tag{2}$$

We continue this process and obtain a ladder of spaces $\{V_k\}$ known as a *multiresolution analysis* of $L^2(\mathbb{R})$. Please see [2] for a complete treatment.

Thus any function $f\in V_1$ can be decomposed into a course (lowpass) part from V_0 and the detail (highpass) from W_0. In the discrete setting, the matrix representation of this transformation is

$$\boldsymbol{H}=\begin{bmatrix}
h_0 & h_1 & h_2 & h_3 & 0 & 0 & \cdots & \cdots & h_{N-1} & h_N & 0 & 0 & 0 & 0 & \cdots\\
0 & 0 & h_0 & h_1 & h_2 & h_3 & \cdots & \cdots & h_{N-3} & h_{N-2} & h_{N-1} & h_N & 0 & 0 & \cdots\\
0 & 0 & 0 & 0 & h_0 & h_1 & \cdots & \cdots & h_{N-5} & h_{N-4} & h_{N-3} & h_{N-2} & h_{N-1} & h_N & \cdots\\
\vdots & & \vdots & & & & \vdots & \vdots & & & & & \vdots & \vdots & \vdots\\
g_0 & g_1 & g_2 & g_3 & 0 & 0 & \cdots & \cdots & g_{N-1} & g_N & 0 & 0 & 0 & 0 & \cdots\\
0 & 0 & g_0 & g_1 & g_2 & g_3 & \cdots & \cdots & g_{N-3} & g_{N-2} & g_{N-1} & g_N & 0 & 0 & \cdots\\
0 & 0 & 0 & 0 & g_0 & g_1 & \cdots & \cdots & g_{N-5} & g_{N-4} & g_{N-3} & g_{N-2} & g_{N-1} & g_N & \cdots\\
\vdots & & \vdots & & & & \vdots & \vdots & & & & & \vdots & \vdots & \vdots
\end{bmatrix} \tag{3}$$

As is the case in [1], an alternative form of $\boldsymbol{H}$ is used. Here the lowpass and highpass rows are intertwined. If we introduce the 2×2 matrices

$$H_k=\begin{bmatrix} h_{2k} & h_{2k+1}\\ g_{2k} & g_{2k+1}\end{bmatrix}$$

for $k=0,\ldots N-1$, we can alternatively write $\boldsymbol{H}$ as

$$\boldsymbol{H}=\begin{bmatrix}
H_0 & H_1 & H_2 & \cdots & H_{N-3} & H_{N-2} & H_{N-1} & 0 & 0 & \cdots & 0\\
0 & H_0 & H_1 & \cdots & H_{N-4} & H_{N-3} & H_{N-2} & H_{N-1} & 0 & \cdots & 0\\
\vdots & & & \vdots & & & & \vdots & & \vdots & \vdots\\
H_1 & H_2 & H_3 & \cdots & H_{N-2} & H_{N-1} & 0 & 0 & 0 & \cdots & H_0
\end{bmatrix} \tag{4}$$

Here, we have truncated (3) to account for finite length images and incorporated a circulant structure. Thus we have chosen to view the data as periodic. There are other ways to do this, but the truncation method has little to do with the factorizations that follow. Note also we have assumed the matrix $\boldsymbol{H}$ has even dimensions. We will hereafter use (4) as our wavelet transform.

We will start with the simplest orthogonal wavelet transformation and use it to show why the scaling factor is necessary.

Let $\phi(x) = \chi_{[0,1)}(x)$ be the Haar scaling function with the associated wavelet $\psi(x) = \chi_{[0,\frac{1}{2})} - \chi_{[\frac{1}{2},1)}$.

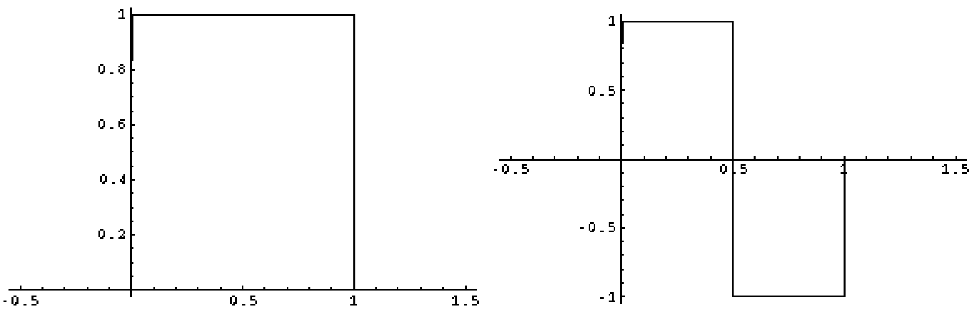

Figure 1. The Haar scaling function ϕ and wavelet ψ.

Note that both ϕ and ψ solve refinement equations

$$\begin{aligned} \phi(x) &= \frac{1}{\sqrt{2}}\phi(2x) + \frac{1}{\sqrt{2}}\phi(2x-1) \\ \psi(x) &= \frac{1}{\sqrt{2}}\phi(2x) - \frac{1}{\sqrt{2}}\phi(2x-1) \end{aligned}$$

We have only one 2×2 matrix

$$H_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

and $\boldsymbol{H}$ is a block diagonal matrix with H_0 down the diagonal. Since $\boldsymbol{H}$ decouples into 2×2 blocks, we can restrict our attention to how H_0 transforms a 2-vector.

It is clear that H_0 will not always map $(a_0, a_1)^T \in \mathbb{Z}^2$ back to $\mathbb{Z}^2$. In fact, the obvious fix is to scale H_0 (and thus $\boldsymbol{H}$) by $\sqrt{2}$ and H_0^{-1} by $\frac{1}{\sqrt{2}}$. In order to understand why this will work with longer filters and since we will floor/truncate transformed data, we will look at how H_0 acts on $[n, n+1)^2$, $n \in \mathbb{Z}$. Figure 2 below shows the effect of transforming $[n, n+1)^2$ by both H_0 and $\sqrt{2}H_0$. Note that the "gaps" in $\mathbb{R}^2$ produced by H_0 correspond to regions of overlap in $H_0^{-1} = H_0^T$ (see Figure 3). Scaling by $\sqrt{2}$ fills these gaps and removes the overlap. Thus no integer pair lies in more than one region of $\frac{1}{\sqrt{2}}H_0^{-1}$ so that we can uniquely recover the original data by inverse transformation and truncation/rounding.

Certainly we want to pick the smallest scaling factor that will do the job and in the case of longer filters, the choice is not as obvious as was the case with the Haar wavelet.

Consider the orthogonal Daubechies 4-tap filter:

$$\begin{array}{llll} h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} & h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} & h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} & h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \\ g_0 = -h_3 & g_1 = h_2 & g_2 = -h_1 & g_3 = h_0 \end{array}$$

Here, we have

$$H_0 = \begin{bmatrix} h_0 & h_1 \\ g_0 & g_1 \end{bmatrix} \qquad H_1 = \begin{bmatrix} h_2 & h_3 \\ g_2 & g_3 \end{bmatrix}$$

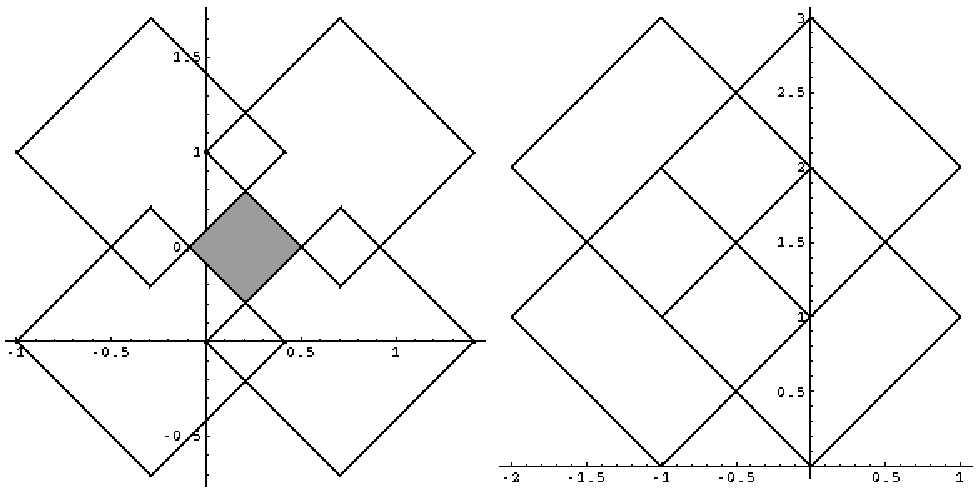

Figure 2. *The action of H_0 (left) and $\sqrt{2}H_0$ (right) on unit squares.*

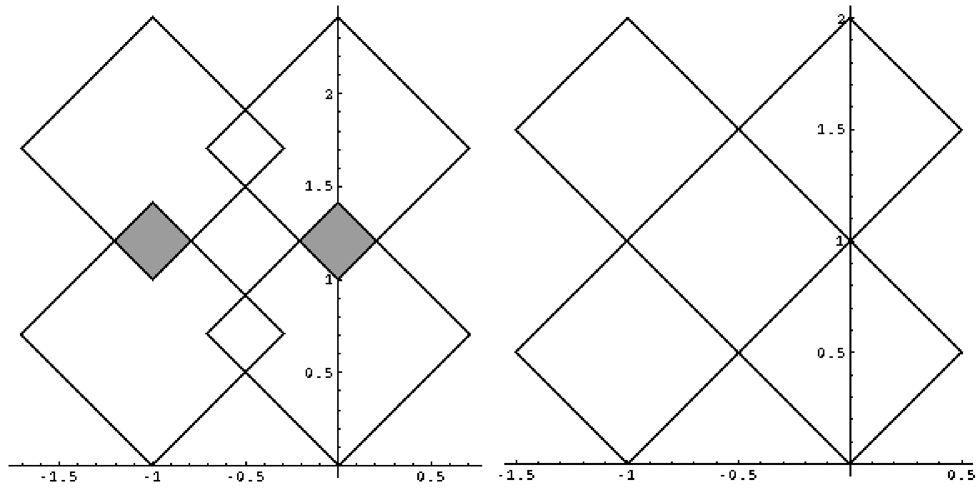

Figure 3. *The action of H_0^{-1} (left) and $\frac{1}{\sqrt{2}}H_0^{-1}$ (right) on unit squares.*

so that the wavelet transform (4) takes the form

$$\boldsymbol{H} = \begin{bmatrix} H_0 & H_1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & H_0 & H_1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & H_0 & H_1 & 0 & 0 & \cdots \\ \vdots & & & \ddots & & & \ddots \\ H_1 & 0 & 0 & \cdots & 0 & 0 & H_0 \end{bmatrix}$$

The authors of [1] introduced orthogonal matrices $U = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1+\sqrt{3} & 1-\sqrt{3} \\ \sqrt{3}-1 & 1+\sqrt{3} \end{bmatrix}$ and $V^T = \frac{1}{2} \begin{bmatrix} 1 & \sqrt{3} \\ -\sqrt{3} & 1 \end{bmatrix}$ and used them to factor H_0, H_1 as follows:

$$H_0 = U \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} V^T = U\tilde{H}_0 V^T$$

and

$$H_1 = U \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} V^T = U\tilde{H}_1 V^T$$

so that the transform H could be rewritten as

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{P}\boldsymbol{V}^T \tag{5}$$

where $\boldsymbol{U}$, $\boldsymbol{V}$ are block diagonal matrices with U, V comprising the diagonal blocks of $\boldsymbol{U}$, $\boldsymbol{V}$, respectively, and $\boldsymbol{P}$ is the block diagonal matrix

$$\boldsymbol{P} = \begin{bmatrix} \tilde{H}_0 & \tilde{H}_1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \tilde{H}_0 & \tilde{H}_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{H}_0 & \tilde{H}_1 & 0 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \ddots & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \tilde{H}_0 \\ \tilde{H}_1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \vdots & & & \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Note that $\boldsymbol{P}$ is a permutation matrix and the factorization (5) is extremely useful in determining the appropriate scaling factor.

Following Calderbank, et al [1], we note that $\boldsymbol{H}a$ is not normally integer valued for $a \in \mathbb{Z}^2$.

Note: For the remainder of this discussion, we consider $\boldsymbol{H}$ acting on only two components of the integer input vector.

We can however, floor/truncate, say by the function $f(a)$ so that $\boldsymbol{H}a + f(a) \in \mathbb{Z}^2$. Thus in order to invert we need

$$\boldsymbol{H}^{-1} f(a) \neq \boldsymbol{H}^{-1} f(b) + (b - a) \tag{6}$$

for all $a, b \in \mathbb{Z}^2$ with $a \neq b$. Alternatively, if Λ is a fundamental region in $\mathbb{R}^2$, (6) becomes

$$\boldsymbol{H}^{-1}\Lambda \cap (\boldsymbol{H}^{-1}\Lambda + n) = \emptyset \tag{7}$$

for $n \in \mathbb{Z}^2$, $n \neq 0$. Now suppose Ω to be a parallelpiped centered at the origin that contains Λ and chosen so that $U^T\Omega$ is a square S centered at the origin. We can easily do this since U is orthogonal. Then $\boldsymbol{P}^{-1}$ will leave $U^T\Omega$ unchanged. Then substituting $\boldsymbol{H}^{-1} = \boldsymbol{VPU}^T$ into (7) gives for $n \in \mathbb{Z}^2$, $n \neq 0$:

$$\begin{aligned} \boldsymbol{VPU}^T\Omega \cap \boldsymbol{VPU}^T\Omega + n &= \{0\} \\ \boldsymbol{V}S \cap (\boldsymbol{V}S + n) &= \{0\} \\ 2(V^TS)^\circ \cap \mathbb{Z}^2 &= \{0\} \end{aligned} \tag{8}$$

So as stated in [1], to find the appropriate scaling factor α, we

(C1) Find Ω so that $\alpha U^T\Omega$ is a square centered at the origin that contains a fundamental region for $\mathbb{Z}^2$.

(C2) Satisfy (8)

This is worked out for the 4-tap in [1] and both conditions are satisfied if the square S centered at the origin has sides of length $\frac{\sqrt{3}}{4}$ and $\alpha \geq \sqrt{2}$. Once α is in hand, the following algorithm is proposed in [1]:

1. Given $a \in \mathbb{Z}^L$, compute $b = \sqrt{2}\boldsymbol{H}a$.
2. Floor/round each b_n to b'_n so that $b'_n \in \mathbb{Z}$ and $b'_n - \frac{1}{2} \leq b_n \leq b'_n + \frac{1}{2}$

To invert, compute $c = \frac{1}{\sqrt{2}}\boldsymbol{H}^Tb'$ and then find unique integer vector a so that $V^T[(a_{2n} - a_{2n+1})^T - (c_{2n} - c_{2n+1})^T]$ maps into S.

Finding appropriate scaling factors for longer orthogonal filters and biorthogonal filters are discussed in [1], but we will concentrate only on multiwavelets constructed from 4 filter coefficients.

We conclude this section with a brief overview of multiwavelets. In this case, we start with A scaling functions $\phi_1, \ldots, \phi_A$ stored in vector

$$\Phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_A(x) \end{bmatrix}$$

that satisfies the *matrix refinement equation*

$$\Phi(x) = \sum_{k=0}^{2N-1} C_k\Phi(2x - k) \tag{9}$$

where now the coefficients C_k are $A \times A$ matrices. Fundamental results for multiwavelets can be found in [4, 10, 5, 7]. We can still construct the multiresolution analysis $\cdots \subset V_0 \subset V_1 \subset \cdots$ as above (see [4]) and construct a wavelet for the corresponding W_j spaces

$$\Psi(x) = \sum_{k=0}^{2N-1} D_k \Phi(2x-k) \tag{10}$$

but now we do not have the nice relationship between the C_k and the D_k as is the case with the orthogonal uni-scaling functions. While the analysis is more difficult in this setting, there are advantages. For example, the popular space of piecewise C^1 cubic polynomials cannot be spanned by the integer translates of a single function, but $A = 2$ are needed. It can be shown that no orthogonal scaling function/wavelet pair can be constructed so that it is regular, compactly supported, orthogonal, and (anti)symmetric. The DGHM wavelets [3] satisfy all four of these desirable properties. Let $A = 2$ and consider the refinement equation

$$\Phi(x) = \sum_{k=0}^{3} C_k \Phi(2x-k) \tag{11}$$

where

$$C_0 = \begin{bmatrix} \frac{3\sqrt{2}}{10} & \frac{4}{5} \\ -\frac{1}{20} & -\frac{3\sqrt{2}}{20} \end{bmatrix} \qquad C_1 = \begin{bmatrix} \frac{3\sqrt{2}}{10} & 0 \\ \frac{9}{20} & \frac{\sqrt{2}}{2} \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 0 & 0 \\ \frac{9}{20} & -\frac{3\sqrt{2}}{20} \end{bmatrix} \qquad C_3 = \begin{bmatrix} 0 & 0 \\ -\frac{1}{20} & 0 \end{bmatrix}.$$

These are the filter coefficients for the DGHM scaling functions (see Figure 4). The associated wavelet coefficients are

$$D_0 = \begin{bmatrix} \frac{\sqrt{3}}{20} & -\frac{3\sqrt{6}}{20} \\ 0 & 0 \end{bmatrix} \qquad D_1 = \begin{bmatrix} -\frac{9\sqrt{3}}{20} & \frac{\sqrt{6}}{6} \\ 0 & -\frac{\sqrt{3}}{3} \end{bmatrix},$$

$$D_2 = \begin{bmatrix} \frac{3\sqrt{3}}{20} & -\frac{\sqrt{6}}{20} \\ \frac{3\sqrt{6}}{10} & -\frac{\sqrt{3}}{5} \end{bmatrix} \qquad D_3 = \begin{bmatrix} -\frac{\sqrt{3}}{60} & 0 \\ -\frac{\sqrt{6}}{30} & 0 \end{bmatrix}.$$

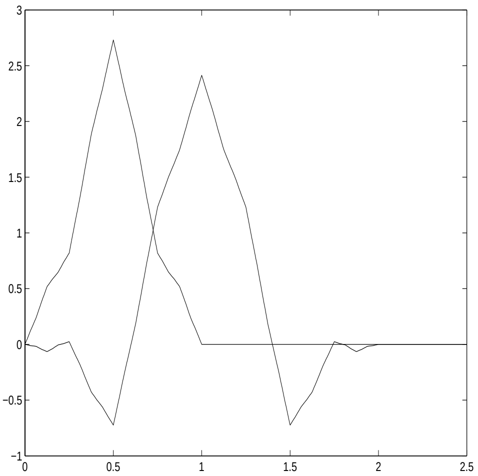

Figure 4. The DGHM scaling functions ϕ_1 and ϕ_2.

3 Factoring the 4-Tap Multiwavelet Transform

In this section, we will start by considering the DGHM multiwavelet and develop a general factorization for 4-tap multiwavelets of any number of scaling functions.

We start by defining H_0 and H_1 for the DGHM multiwavelet. In this case we have:

$$\begin{aligned} H_0 &= \begin{bmatrix} C_0 & C_1 \\ D_0 & D_1 \end{bmatrix} \\ &= \begin{bmatrix} 3\sqrt{2}/10 & 4/5 & 3\sqrt{2}/10 & 0 \\ -1/20 & -3\sqrt{2}/20 & 9/20 & \sqrt{2}/2 \\ \sqrt{3}/20 & 3\sqrt{6}/20 & -9\sqrt{3}/20 & \sqrt{6}/6 \\ 0 & 0 & 0 & -\sqrt{3}/3 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} H_1 &= \begin{bmatrix} C_2 & C_3 \\ D_2 & D_3 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 9/20 & -3\sqrt{2}/20 & -1/20 & 0 \\ 3\sqrt{3}/20 & -\sqrt{6}/20 & -\sqrt{3}/60 & 0 \\ 3\sqrt{6}/10 & -\sqrt{3}/5 & -\sqrt{6}/30 & 0 \end{bmatrix} \end{aligned}$$

Note that now H_0 and H_1 are 4×4 matrices comprised of matrices. Also, in the case of the wavelets, a certain symmetry is enjoyed by $\tilde{H}_0$ and $\tilde{H}_1$. In the case of the DGHM multiwavelets, we will not be able to obtain an analgous factorization since a straightforward computation shows that

$$r_0 := \mathrm{rank}(H_0) = 3$$

and

$$r_1 := \mathrm{rank}(H_1) = 1$$

It is encouraging that r_0 and r_1 sum to 4 - twice the number of scaling functions. This is also the case with the Daubechies' 4-tap wavelet.

We proceed then by computing the singular value decompositions (SVD) of both H_0 and H_1. Here we adopt the convention that we the singular values are nonnegative with the largest entry occupying the $(1,1)$ position of the diagonal matrix and the diagonal entries decreasing.

We have:

$$\begin{aligned}
H_0 &= U_0 D_0 V_0^T \\
&= \left[\begin{array}{cccc} \vec{u}_0^1 & \vec{u}_0^2 & \vec{u}_0^3 & \vec{u}_0^4 \end{array}\right] D_0 \left[\begin{array}{cccc} \vec{v}_0^1 & \vec{v}_0^2 & \vec{v}_0^3 & \vec{v}_0^4 \end{array}\right]^T \\
&\approx \left[\begin{array}{cccc} -.973329 & .2001360 & -.112147 & \\ .114708 & .0788920 & -.854762 & \vec{u}_0^4 \\ -.198680 & -.0934916 & .055910 & \\ 0 & .2822310 & .503665 & \end{array}\right] \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \\
&\quad \cdot \left[\begin{array}{cccc} -.43589 & 0 & 0 & \\ -.875996 & -.200136 & .112147 & \vec{v}_0^4 \\ -.206474 & .849107 & -.475801 & \\ 0 & -.488839 & -.872374 & \end{array}\right]^T
\end{aligned} \tag{12}$$

and

$$\begin{aligned}
H_1 &= U_1 D_1 V_1^T \\
&= \left[\begin{array}{cccc} \vec{u}_1^1 & \vec{u}_1^2 & \vec{u}_1^3 & \vec{u}_1^4 \end{array}\right] D_1 \left[\begin{array}{cccc} \vec{v}_1^1 & \vec{v}_1^2 & \vec{v}_1^3 & \vec{v}_1^4 \end{array}\right]^T \\
&\approx \left[\begin{array}{cccc} 0 & & & \\ -.5 & \vec{u}_1^2 & \vec{u}_1^3 & \vec{u}_1^4 \\ -.288675 & & & \\ -.816497 & & & \end{array}\right] \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{cccc} -.9 & & & \\ .424264 & \vec{v}_1^2 & \vec{v}_1^3 & \vec{v}_1^4 \\ .1 & & & \\ 0 & & & \end{array}\right]^T
\end{aligned} \tag{13}$$

We need to determine the unknown vectors. A direct computation indicates that $\vec{u}_1^1$ is orthogonal to the vectors $\vec{u}_0^j$, $j = 1,2,3$. Likewise, $\vec{v}_1^1$ is orthogonal to $\vec{v}_0^j$, $j = 1,2,3$. We can verify these orthogonality relations hold if we consider the fact that if $\boldsymbol{H}$ is not truncated, then it is an orthogonal matrix. Computing $\boldsymbol{H}\boldsymbol{H}^T = I$ immediately gives rise to the following two identities:

$$H_0^T H_1 = 0 \tag{14}$$

$$H_0^T H_0 + H_1^T H_1 = I \tag{15}$$

Inserting (12,13) into (14) gives

$$\begin{aligned} D_0 U_0^T U_1 D_1 &= 0 \\ \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} &= 0 \end{aligned} \tag{16}$$

where L is an $r_0 \times r_1$ matrix whose compontents are $(\vec{u}_0^j, \vec{u}_1^k)$, $j = 1, \ldots, r_0$ and $k = 1, \ldots, r_1$. Once we have the vectors of U_0, U_1 orthogonal, it is a straightforward task to verify that $(\vec{v}_0^j, \vec{v}_1^k) = 0$, $j = 1, \ldots, r_0$ and $k = 1, \ldots, r_1$.

We can complete the factorization then by setting $\vec{u}_0^4 = \vec{u}_1^1$ and $\vec{v}_0^4 = \vec{u}_1^4$. Furthermore, we can define

$$U = U_0 \qquad V = V_0$$

and we see that

$$H_0 = U \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} V = U\tilde{H}_0 V$$

and

$$H_1 = U \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} V = U\tilde{H}_1 V$$

In this way, we can factor the DGHM multiwavelet transform in a way analogous to the 4-tap Daubechies wavelet transform discussed in Section 2. We can generalize this factorization to all 4-tap orthogonal multiwavelet transforms. In this case it is important to note that the ranks r_0, r_1 of H_0, H_1, respectively satisfy

$$r_0 + r_1 = 2A, \tag{17}$$

where A is the number of scaling functions in the scaling vector. This result is proven in [12]. In this case we can use (16) to show that the vectors of U_0 (V_0) and U_1 (V_1) are orthogonal. Moreover, we can then define U, V as above and show that the positive singular values of D_0, D_1 are 1. If we let P be the permutation matrix that moves the r_1 positive values of D_1 to the last r_1 rows of D_1, we have:

$$\begin{aligned} I &= H_0^T H_0 + H_1^T H_1 \\ &= V D_0 U^T U D_0 V^T + V P^T D_1 U^T U P D_1 V^T \\ &= (D_0)^2 + (P^T D_1 P D_1) \\ &= (D_0)^2 + (P D_1)^2 \end{aligned}$$

Since $r_0 + r_1 = 2A$, we have the nonzero diagonal elements of D_0, D_1 equal to ± 1. Since the singular values are nonnegative, we have the desired result. An algorithm then to factor any orthogonal 4-tap multiwavelet is as follows:

1. Find the singular value decompositions of H_0, H_1.

2. Take $U = U_0$, $V = V_0$.

3. $H_0 = U\tilde{H}_0V^T$, $H_1 = U\tilde{H}_1V^T$. Where

$$H_0 = \mathrm{diag}[\underbrace{1, 1, \ldots, 1}_{r_0\text{-times}}, 0, \ldots, 0]$$

and

$$H_1 = \mathrm{diag}[0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{r_1\text{-times}}]$$

We conclude this section with one more example. In [8], the authors showed how to complete the matrix $\boldsymbol{H}$ given the lowpass coefficients C_k. They illustrated their technique (for $A = 3$) with the following matrix coefficients:

$$C_0 = \frac{\sqrt{2}}{8}\begin{bmatrix} 3 & -\sqrt{3} & 0 \\ 3 & -\sqrt{3} & 0 \\ \sqrt{3} & 2 & \sqrt{3} \end{bmatrix} \qquad C_1 = \frac{\sqrt{2}}{8}\begin{bmatrix} 1 & -\sqrt{3} & 0 \\ 1 & -\sqrt{3} & 0 \\ -\sqrt{3} & 0 & \sqrt{3} \end{bmatrix}$$

$$C_2 = \frac{\sqrt{2}}{8}\begin{bmatrix} 1 & \sqrt{3} & 0 \\ -1 & -\sqrt{3} & 0 \\ -\sqrt{3} & 0 & \sqrt{3} \end{bmatrix} \qquad C_3 = \frac{\sqrt{2}}{8}\begin{bmatrix} 3 & \sqrt{3} & 0 \\ -3 & -\sqrt{3} & 0 \\ \sqrt{3} & -2 & \sqrt{3} \end{bmatrix}$$

and

$$D_0 = \begin{bmatrix} 0 & 0 & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} \\ \frac{\sqrt{6}}{8} & \frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{8} \end{bmatrix} \qquad D_1 = \begin{bmatrix} 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & \frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} \\ -\frac{\sqrt{6}}{8} & 0 & \frac{\sqrt{6}}{8} \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\sqrt{6}}{8} & 0 & -\frac{\sqrt{6}}{8} \end{bmatrix} \qquad D_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{\sqrt{6}}{8} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{6}}{8} \end{bmatrix}$$

In this case,

$$H_0 = \begin{bmatrix} \frac{3\sqrt{2}}{8} & -\frac{\sqrt{6}}{8} & 0 & \frac{\sqrt{2}}{8} & -\frac{\sqrt{6}}{8} & 0 \\ \frac{3\sqrt{2}}{8} & -\frac{\sqrt{6}}{8} & 0 & \frac{\sqrt{2}}{8} & -\frac{\sqrt{6}}{8} & 0 \\ \frac{\sqrt{6}}{8} & \frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{8} & -\frac{\sqrt{6}}{8} & 0 & \frac{\sqrt{6}}{8} \\ 0 & 0 & -\frac{\sqrt{2}}{2} & 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} \\ \frac{\sqrt{6}}{8} & \frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{8} & -\frac{\sqrt{6}}{8} & 0 & \frac{\sqrt{6}}{8} \end{bmatrix}$$

and

$$H_1 = \begin{bmatrix} \frac{\sqrt{2}}{8} & \frac{\sqrt{6}}{8} & 0 & \frac{3\sqrt{2}}{8} & \frac{\sqrt{6}}{8} & 0 \\ -\frac{\sqrt{2}}{8} & -\frac{\sqrt{6}}{8} & 0 & -\frac{3\sqrt{2}}{8} & -\frac{\sqrt{6}}{8} & 0 \\ -\frac{\sqrt{6}}{8} & 0 & \frac{\sqrt{6}}{8} & \frac{\sqrt{6}}{8} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\sqrt{6}}{8} & 0 & -\frac{\sqrt{6}}{8} & -\frac{\sqrt{6}}{8} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{6}}{8} \end{bmatrix}$$

We have $r_0 = 4$ and $r_1 = 2$ and U, V, $\tilde{H}_0$, $\tilde{H}_1$ take the form:

$$U \approx \begin{bmatrix} -.7000006 & .0759191 & -.0650219 & 0 & -.353553 & -.612372 \\ -.7000006 & .0759191 & -.0650219 & 0 & .353553 & .612372 \\ .0879292 & .248924 & -.655977 & 0 & .612372 & -353553 \\ -.044513 & -.615616 & -.239575 & .749426 & 0 & 0 \\ -.0503848 & -.696823 & -.271178 & -.662088 & 0 & 0 \\ .0879292 & .248924 & -.655977 & 0 & -.612372 & .353553 \end{bmatrix}$$

$$V \approx \begin{bmatrix} -.688623 & .232959 & -.470668 & 0 & -.5 & 0 \\ .521694 & .556241 & -.257966 & .405445 & -.216506 & -.375 \\ .0675072 & .341377 & -.328173 & -.764008 & 0.375 & -.216506 \\ -.301335 & -.125593 & .378713 & 0 & 0 & -.866025 \\ .39781 & -.473206 & -.126244 & -.405445 & -.649519 & -.125 \\ .0045562 & -.529236 & -.666983 & .2958 & .375 & -.216506 \end{bmatrix}$$

$$\tilde{H}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \tilde{H}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4 Numerical Examples

We conclude this paper by comparing the integer DGHM multiwavelet transform to other transforms. It is a straightforward task to compute the scaling value α for the integer to integer DGHM using the conditions C1, C2 in Section 2. We find

$$\alpha \approx 15.72$$

We use the standard test image *lena* shown in the figure below. This image is 512×512 pixels in size and we perform 5 decompositions of each wavelet transform. The wavelet transforms we use are the orthogonal Daubechies 4-tap (D4), the integer to integer orthogonal Daubechies 4-tap (ID4), the DGHM tranform, and the integer DGHM transform (IDGHM).

Figure 5. The test image *lena*.

For comparison, we measure the entropy of each decomposition. The entropy formula is given by

$$E^2(v) = -\sum_k \frac{v_k^2}{||v||^2} \log_2 \frac{v_k^2}{||v||^2}$$

for vector v. We follow [1] and weight the entropy at level j of the decomposition by 4^{-j}, $j = 1, 2, 3, 4, 5$.

The table below gives our results.

	D4	ID4	DGHM	IDGHM
$E(v)$	1.2277533	1.2274470533	1.3142649337	1.3145029853

References

[1] R. Calderbank, I. Daubechies, W. Sweldens, and B.L. Yeo, "Wavelet transform that map integers to integers", *App. Comp. Harm. Anal.*, No. 3, **5**(1998), 332–369.

[2] I. Daubechies, "Orthonormal bases of compactly supported wavelets", *Comm. Pure App. Math.* **41**(1988), 909–996.

[3] G. Donovan, J. Geronimo, D. Hardin, and P. Massopust, "Construction of orthogonal wavelets using fractal interpolation functions", *SIAM J. of Math. Anal.*, No. 4**27**(1996), 1158–1192.

[4] J. Geronimo, D. Hardin, and P. Massopust, "Fractal functions and wavelet expansions based on several scaling functions", J. Approx. Theory, **78**(3) (1994), 373-401.

[5] T.N.T Goodman, and S. L. Lee, "Wavelets of multiplicity r", Trans. Amer. Math. Soc., Vol. 342, Number 1, March 1994, 307-324.

[6] R. Laroia, S.A. Tretter, and N. Farvardin, "A simple and effective precoding scheme for noise whitening on intersymbol interference channels", *IEEE Trans. Comm.*, **41**(1993), 460–463.

[7] P.R. Massopust, D.K. Ruch, and P.J. Van Fleet, "On the support properties of scaling vectors", Comp. and Appl. Harm. Anal., **3**(1996), 229–238.

[8] J. Pan, L. Jiao, and F. Yangwang, "Construction of orthogonal multiwavelets with short sequence", preprint.

[9] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, 1997.

[10] G.Strang and V. Strela, "Orthogonal multiwavelets with vanishing moments", *Proc. SPIE Conference on Mathematics of Imaging, J. Optical Eng.*, **33**(1994), 2104–2107.

[11] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets", *App. .Comp. Harm. Anal.*, No. 3 **2**(1996), 186–200.

[12] P.J. Van Fleet, "Integer to Integer Transforms from 4-Tap Multiwavelets", preprint.