Introduction to the Discrete Wavelet Transform (DWT)

(last edited 02/15/2004)

1 Introduction

This is meant to be a brief, practical introduction to the *discrete wavelet transform (DWT)*, which augments the well written tutorial paper by Amara Graps [1]. Therefore, this document is not meant to be comprehensive, but does include a discussion on the following topics:

1. Qualitative discussion on the DWT decomposition of a signal;
2. Procedure for computing the forward and inverse DWT; and
3. The 2D DWT.

2 DWT decomposition

In Fourier analysis, the Discrete Fourier Transform (DFT) decompose a signal into sinusoidal basis functions of different frequencies. No information is lost in this transformation; in other words, we can completely recover the original signal from its DFT (FFT) representation.

In wavelet analysis, the Discrete Wavelet Transform (DWT) decomposes a signal into a set of mutually orthogonal *wavelet* basis functions. These functions differ from sinusoidal basis functions in that they are spatially localized – that is, nonzero over only part of the total signal length. Furthermore, wavelet functions are *dilated*, *translated* and *scaled* versions of a a common function ϕ, known as the *mother wavelet*. As is the case in Fourier analysis, the DWT is invertible, so that the original signal can be completely recovered from its DWT representation.

Unlike the DFT, the DWT, in fact, refers not just to a single transform, but rather a set of transforms, each with a different set of wavelet basis functions. Two of the most common are the *Haar* wavelets and the *Daubechies* set of wavelets. For example, Figures 1 and 2 illustrate the complete set of 64 Haar and Daubechies-4 wavelet functions (for signals of length 64), respectively. Here, we will not delve into the details of how these were derived; however, it is important to note the following important properties:

1. Wavelet functions are spatially localized;
2. Wavelet functions are dilated, translated and scaled versions of a common mother wavelet; and
3. Each set of wavelet functions forms an orthogonal set of basis functions.

3 DWT in one dimension

In this section, we describe the algorithm for computing the one-dimensional DWT and its inverse.

3.1 Forward DWT

The (one-dimensional) DWT operates on a real-valued vector x of length 2^n, $n \in \{2, 3, \dots\}$, and results in a transformed vector w of equal length. Figure 3(a) and (b) illustrate the first two steps of the DWT for a vector of length 16. First, the vector x is filtered with some discrete-time, low-pass filter (LPF) h of given length (in the Figures, we use length four for illustration purposes) at intervals of two, and the resulting

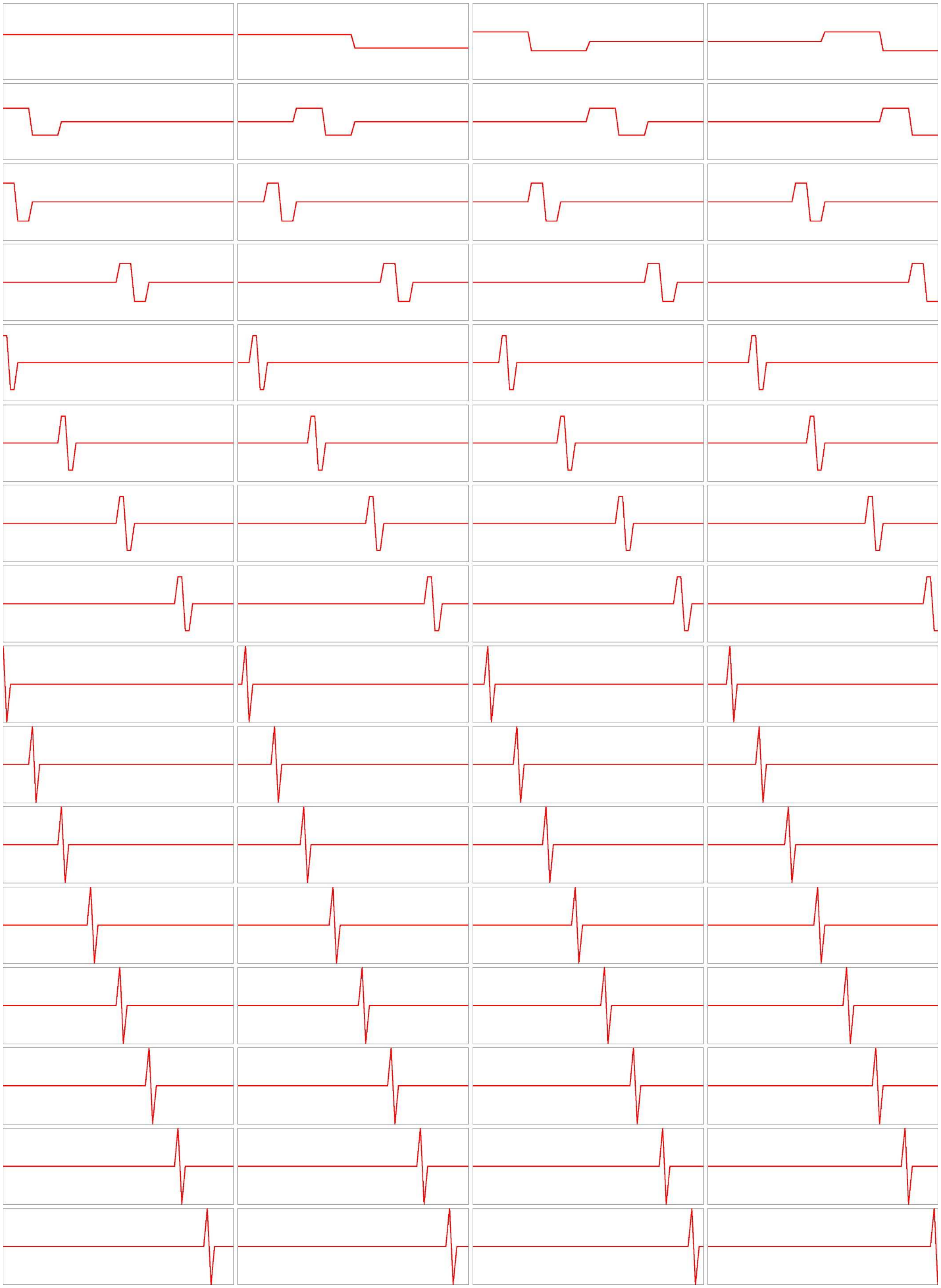

Figure 1: Haar wavelet basis functions (length 64).

Figure 2: Daubechies-4 wavelet basis functions (length 64).

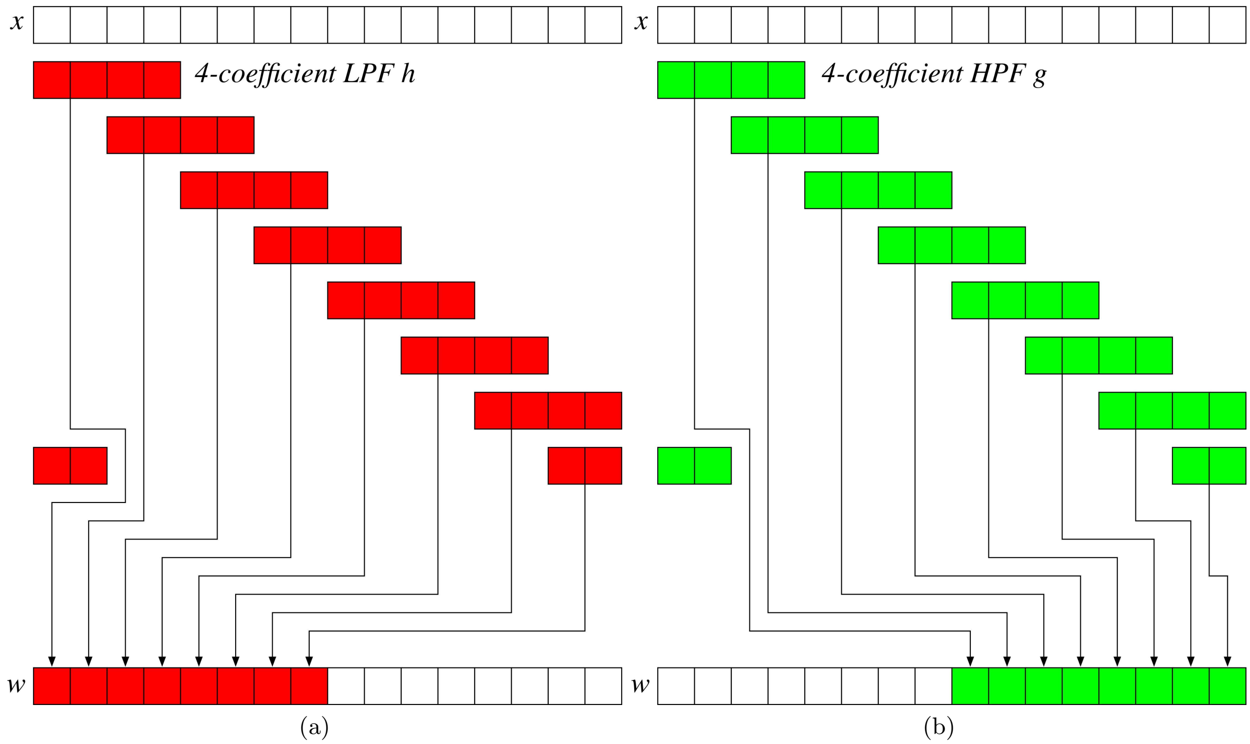

Figure 3: (a) First step of the DWT for a signal of length 16: The original signal is low-pass filtered in increments of two, and the resulting coefficients are grouped as the first eight elements of the vector. (b) Second step of the DWT: The original signal is high-pass filtered in increments of two, and the resulting coefficients are grouped as the last eight elements of the vector.

values are stored in the first eight elements of w. This step is illustrated in Figure 3(a). Second, the vector x is filtered with some discrete-time, high-pass filter (HPF) g of given length (again, for illustration purposes, we use a filter of length four) at intervals of two, and the resulting high-pass values are stored in the last eight elements of w. This step is illustrated in Figure 3(b).

Note, qualitatively, how this procedure transforms the vector x. The low-pass part of the vector w is essentially a down-sampled version (down-sampled by a factor of two) of the original signal x, while the high-pass part of the vector w detects and localizes high frequencies in x. If we were to stop here, the vector w would be a *one-level* wavelet transform of x. We need not, however, stop here; the low-pass filtered part of w (first eight elements for this example) can be further transformed using the identical procedure as outlined above and shown in Figure 3. Figure 4, for example, illustrates a *three-level*, one-dimensional DWT.

Note that in the final transform w_3, values L_3 are the result of three consecutive low-pass filters, values H_3 are the result of two consecutive low-pass filtering operations followed by a high-pass filter, values H_2 are the result of a low-pass filter followed by a high-pass filter, and values H_1 are the result of one high-pass filter. Therefore, the highest frequencies will be isolated and localized in values H_1 of w_3, intermediate frequencies will be isolated and localized in values H_2 of w_3, etc. Note that for lower frequencies, the resolution is decimated by half for each level of the wavelet transform. Thus, the DWT operation implicitly recognizes that lower frequencies cannot be localized at the same resolution as higher frequencies (see Heisenberg Uncertainty Principle). In summary, the one-dimensional DWT is a multi-resolutional frequency decomposition and localization of a one-dimensional, discrete-time signal.

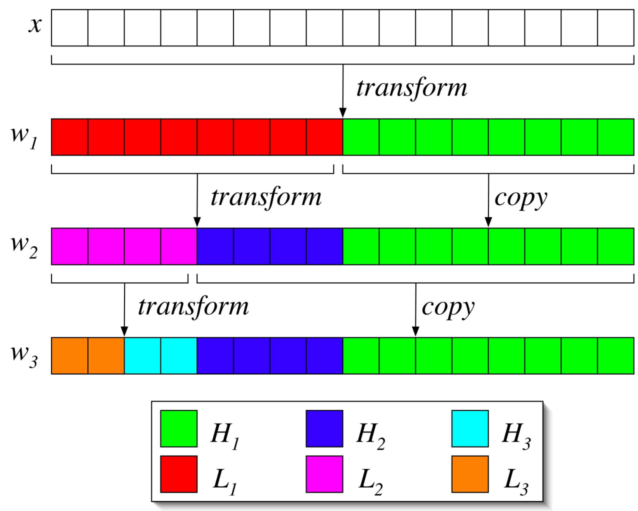

Figure 4: Three-level wavelet transform on signal x of length 16. Note that from w_1 to w_2, coefficients H_1 remain unchanged, while from w_2 to w_3, coefficients H_1 and H_2 remain unchanged.

3.2 Filter coefficients

Thus far, we have remained silent on a very important detail of the DWT – namely, the construction of the low-pass filter h, and the high-pass filter g. Obviously, the filter coefficients for h and g cannot assume arbitrary values, but rather have to be selected carefully in order to lead to basis functions, such as those in Figures 1 and 2, with the necessary properties of *compactness* (i.e. spatial localization) and *orthogonality*.

The derivation of filter coefficients is beyond the scope of this introduction; for an understandable discussion and derivation see [2]. However, we note that filter coefficients for h and g should be related as follows:

$$g_k = (-1)^k h_{n-k-1},\ k \in \{0, \ldots, n-1\}, \tag{1}$$

where n denotes the length of the filter. For example, for filter lengths 2, 4 and 6:

$$h = \begin{bmatrix} c_0 & c_1 \end{bmatrix} \implies g = \begin{bmatrix} c_1 & -c_0 \end{bmatrix} \tag{2}$$

$$h = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix} \implies g = \begin{bmatrix} c_3 & -c_2 & c_1 & -c_0 \end{bmatrix} \tag{3}$$

$$h = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix} \implies g = \begin{bmatrix} c_5 & -c_4 & c_3 & -c_2 & c_1 & -c_0 \end{bmatrix} \tag{4}$$

The simplest wavelet filter is the *Haar* filter, where h is given by,

$$h = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{5}$$

This filter gives rise to basis functions of the type shown in Figure 1. Another very popular set of wavelet filters is due to *Daubechies*. The most compact of these has four coefficients (Daubechies-4), where h is given by,

$$h = \begin{bmatrix} \frac{(1+\sqrt{3})}{4\sqrt{2}} & \frac{(3+\sqrt{3})}{4\sqrt{2}} & \frac{(3-\sqrt{3})}{4\sqrt{2}} & \frac{(1-\sqrt{3})}{4\sqrt{2}} \end{bmatrix}. \tag{6}$$

This filter gives rise to basis functions of the type shown in Figure 2; they will, of course, differ depending on the length of the signal x. Other Daubechies filters of length n, $n \in \{6, 8, 10, \ldots\}$ are also derivable; again, for a discussion and derivation of these filter coefficients, see [2].

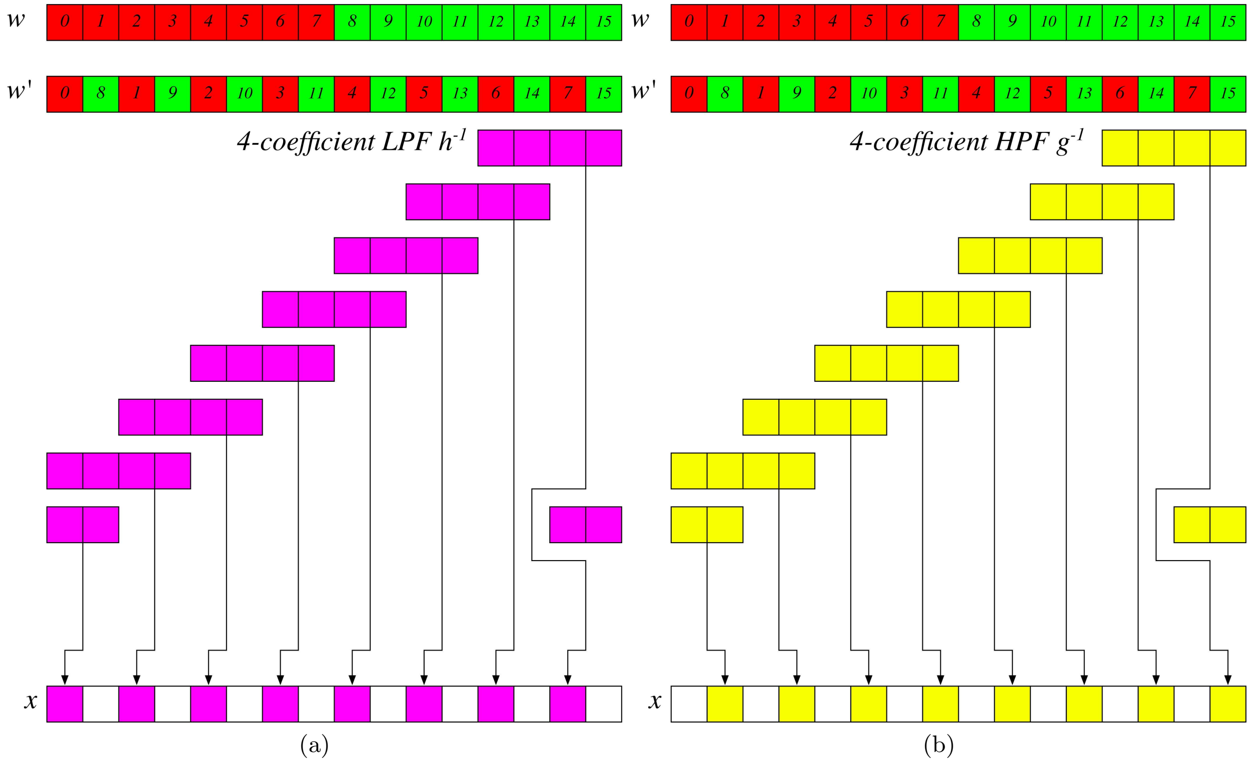

Figure 5: Illustration of the inverse DWT for a one-level DWT w of length 16. First, the low-pass and high-pass elements of w are interleaved. Then, (a) the inverse low-pass filter h^{-1} is applied in increments of two, and (b) the inverse high-pass filter g^{-1} is applied in increments of two.

3.3 Inverse DWT

To understand the procedure for computing the one-dimensional inverse DWT, consider Figure 5, where we illustrate the inverse DWT for a one-level DWT of length 16 (assuming filters of length four). Note that the two filters are now h^{-1} and g^{-1} where,

$$h_k^{-1} = \begin{cases} h_k & k \in \{1, 3, \dots\} \\ h_{n-k-1} & k \in \{0, 2, \dots\} \end{cases} \tag{7}$$

and g^{-1} is determined from h^{-1} using equation (1).

To understand how to compute the one-dimensional inverse DWT for multi-level DWTs, consider Figure 4. First, to compute w_2 from w_3, the procedure in Figure 5 is applied only to values L_3 and H_3. Second, to compute w_1 from w_2, the procedure in Figure 5 is applied to values L_2 and H_2. Finally, to compute x from w_1, the procedure in Figure 5 is applied to all of w_1 – namely, L_1 and H_1.

4 DWT in two dimensions

In this section, we describe the algorithm for computing the two-dimensional DWT through repeated application of the one-dimensional DWT. The two-dimensional DWT is of particular interest for image processing and computer vision applications, and is a relatively straightforward extension of the one-dimensional DWT discussed above.

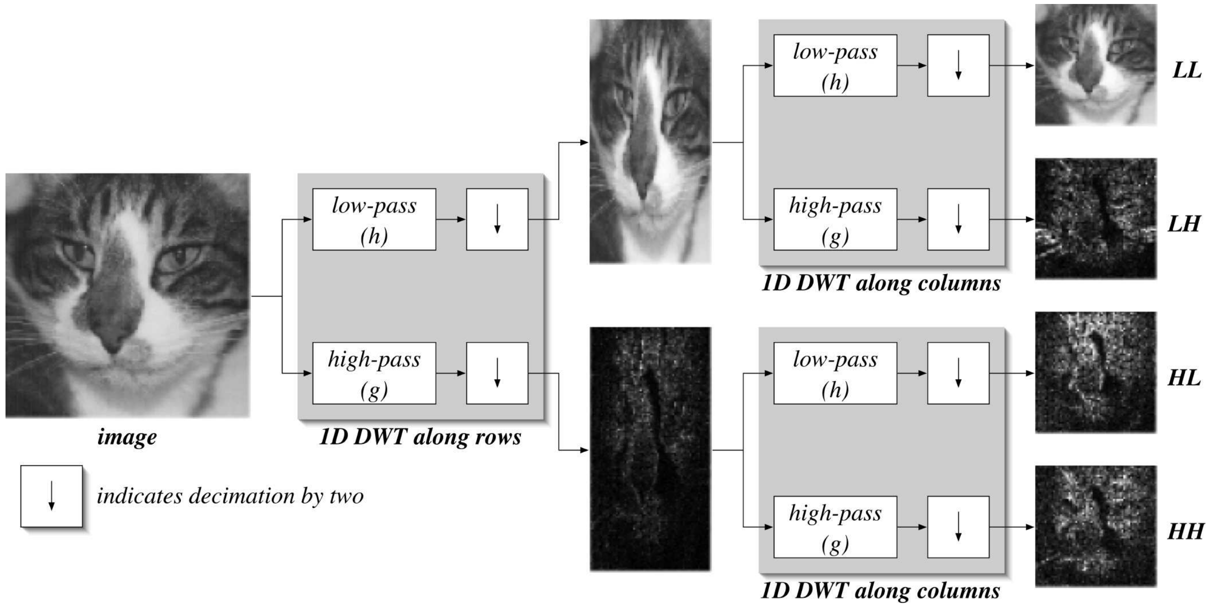

Figure 6: One-level, two-dimensional DWT. First, the one-dimensional DWT is applied along the rows; second, the one-dimensional DWT is applied along the columns of the first-stage result, generating four sub-band regions in the transformed space: LL, LH, HL and HH.

Figure 6 illustrates the basic, one-level, two-dimensional DWT procedure. First, we apply a one-level, one-dimensional DWT along the rows of the image. Second, we apply a one-level, one-dimensional DWT along the columns of the transformed image from the first step. As depicted in Figure 7 (left), the result of these two sets of operations is a transformed image with four distinct *bands*: (1) LL, (2) LH, (3) HL and (4) HH. Here, L stands for low-pass filtering, and H stands for high-pass filtering. The LL band corresponds roughly to a down-sampled (by a factor of two) version of the original image. The LH band tends to preserve localized horizontal features, while the HL band tends to preserve localized vertical features in the original image. Finally, the HH band tends to isolate localized high-frequency point features in the image.

As in the one-dimensional case, we do not necessarily want to stop there, since the one-level, two-dimensional DWT extracts only the highest frequencies in the image. Additional levels of decomposition can extract lower frequency features in the image; these additional levels are applied only to the LL band of the transformed image at the previous level. Figure 7 (right) for example illustrates the three-level, two-dimensional DWT on a sample image.

For many examples and further exploration of the DWT, please see the materials at the following link:

`http://mil.ufl.edu/~nechyba/eel6562/course_materials.html`

References

[1] A. Graps, "An Introduction to Wavelets," *IEEE Computational Sciences and Engineering*, vol. 2, no. 2, pp 50-61, 1995.

[2] W. H. Press, *et. al.*, *Numerical Recipes in C: the Art of Scientific Computing, 2nd. ed.*, pp. 591-606, Cambridge University Press, 1992.

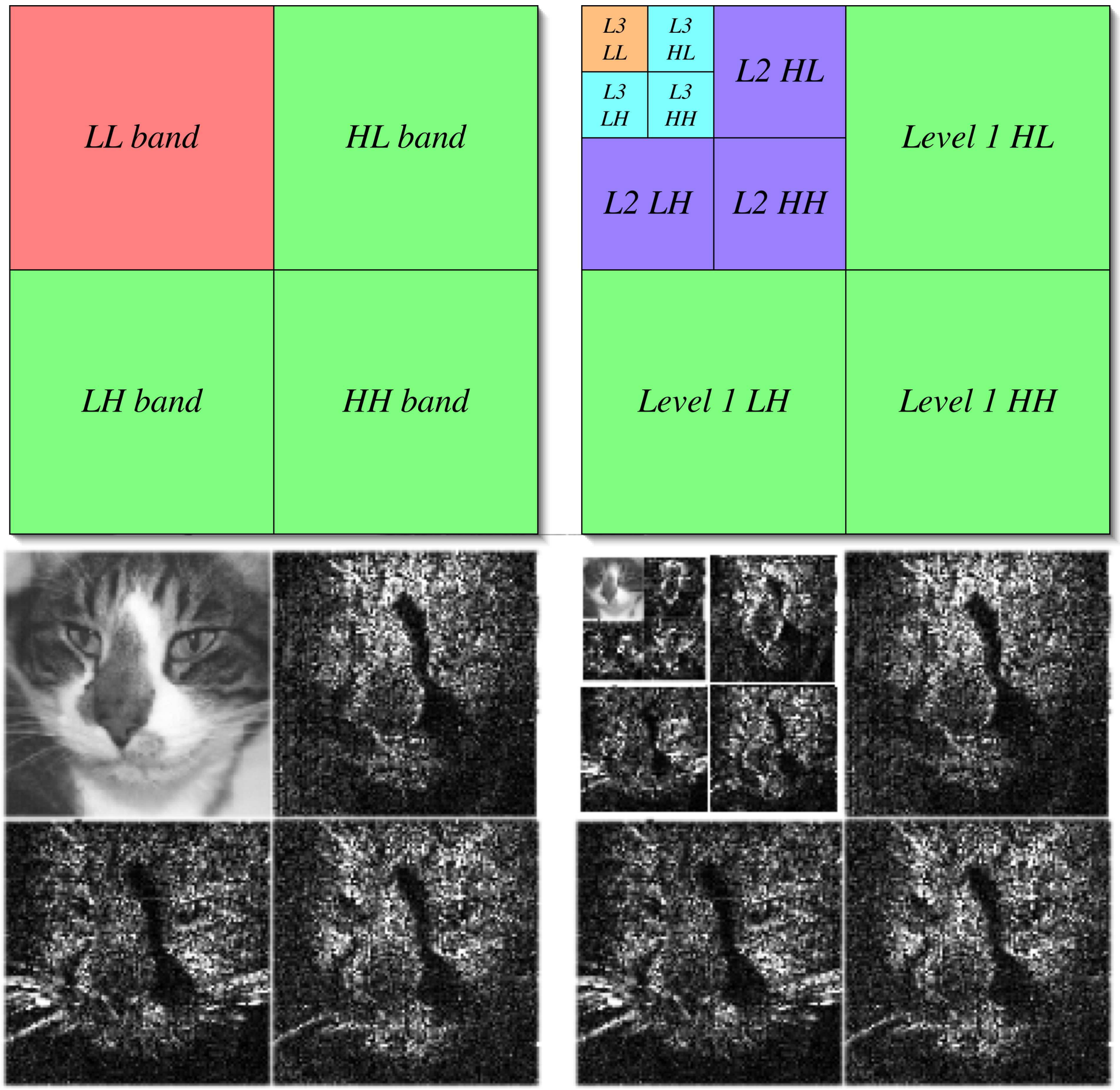

Figure 7: Two-dimensional wavelet transform: (left) one-level 2D DWT of sample image, and (right) three-level 2D DWT of the same image. Note that the LH bands tend to isolate horizontal features, while the HL band tend to isolate vertical features in the image.