CHAPTER II

DECOMPOSITION AND RECONSTRUCTION BY USING WAVELET

Contents

This section we will review how wavelet decomposition work. There are two ways to decompose by wavelets. One is by using wavelet basis directly and one by usingfilter coefficients. Before doing this, we will review the concept of periodic extension first.

2.1 Periodic Extension (Newland 1994)

The purpose of the wavelet transform is to decompose any arbitrary signal $f(x)$ into an infinite summation of wavelets at different scales according to the expansion

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{j,k} \psi(2^j x - k) \qquad (2.1)$$

The interpretation of this expansion can be seen from Figure 2.1. Here the Haar wavelet is used and all wavelet amplitudes $c_{j,k}$ are taken to be unity. The diagram shows level -3, -2, -1, 0, 1, 2 corresponding to j = -3, -2, -1, 0, 1, 2.

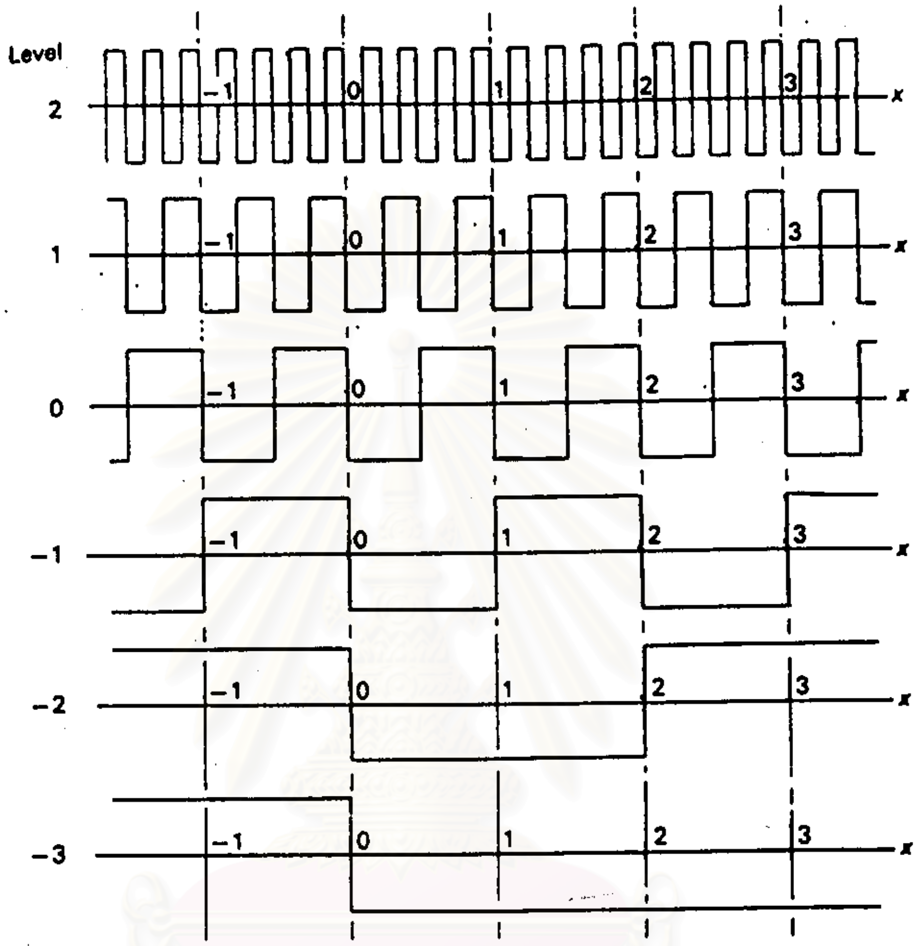


Figure 2.1 Haar wavelets with unit amplitude plotted for levels -3, -2, -1, 0, 1, 2.

At level 0 there is $2^0=1$ Haar wavelet in each unit interval, at level 1 there are two wavelets per unit interval; at level 2 there are four wavelets per unit intervals and so on. At level -1 there is one wavelet every two intervals; at level -2 there is one wavelet every four intervals and so on. We can see that for $j \leq -1$ the contribution of each wavelet is constant over unit intervals. It follows that the sum of the contributions from all these levels is also constant. Since the scaling function of the Haar wavelet is

$$\phi(x) \quad = \quad 1, \quad 0 \le x < 1$$

In this example, it means that

$$\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{j,k} \psi(2^j x - k) \quad = \quad \sum_{k=-\infty}^{\infty} c_{\phi,k} \phi(x-k) \qquad (2.2)$$

From (2.2) the expansion can be written in an alternative form:

$$f(x) \quad = \quad \sum_{k=-\infty}^{\infty} c_{\phi,k} \phi(x-k) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} c_{j,k} \psi(2^j x - k) \qquad (2.3)$$

In order to use discrete wavelet transform algorithm, it is convenient to limit the variable x to one unit interval so that $f(x)$ is assumed to be defined only for $0 \le x < 1$. Now we shall consider a wavelet series expansion that holds for the interval $0 \le x < 1$. A complication arises at the edge of this interval because some of the wavelets $\psi(2^j x - k)$ overlap the edges. To avoid this problem, it is convenient to assume that $f(x)$, $0 \le x < 1$, is one period of a periodic signal so that the signal $f(x)$ is exactly repeated in the adjacent unit intervals to give

$$F(x) \quad = \quad \sum_{k} f(x-k) \qquad (2.4)$$

where $f(x)$ is zero outside the interval $0 \le x < 1$.

Suppose that we use D4 wavelet, $\psi(x)$, which occupies three unit intervals, $0 \le x < 3$, $f(x)$ will receive contributions from the first third of $\psi(x)$, the middle third of $\psi(x+1)$ and the last third of $\psi(x+2)$. This is the same as $\psi(x)$ is "wrapped around" the unit interval. From (2.3) the wavelet expansion of $f(x)$ in $0 \le x < 1$ can be written as

$$f(x) = a_0\phi(x) + a_1\psi(x) + \begin{bmatrix} a_2 & a_3 \end{bmatrix} \begin{bmatrix} \psi(2x) \\ \psi(2x-1) \end{bmatrix}$$

$$+ \begin{bmatrix} a_4 & a_5 & a_6 & a_7 \end{bmatrix} \begin{bmatrix} \psi(4x) \\ \psi(4x-1) \\ \psi(4x-2) \\ \psi(4x-3) \end{bmatrix}$$

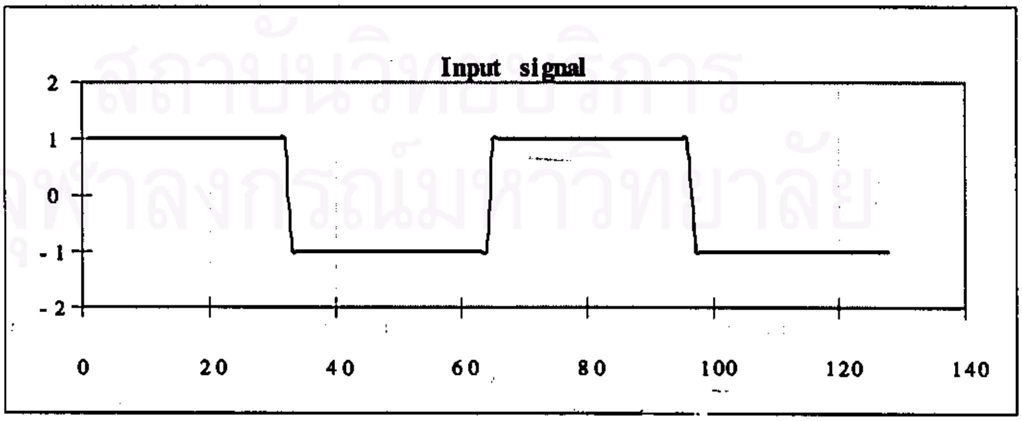
$$+ \ldots a_{2^j+k}\psi(2^j x - k) \qquad (2.5)$$

$$\vdots$$

The coefficients $a_1,\ldots,a_4$ give the amplitudes of each of the wavelet.

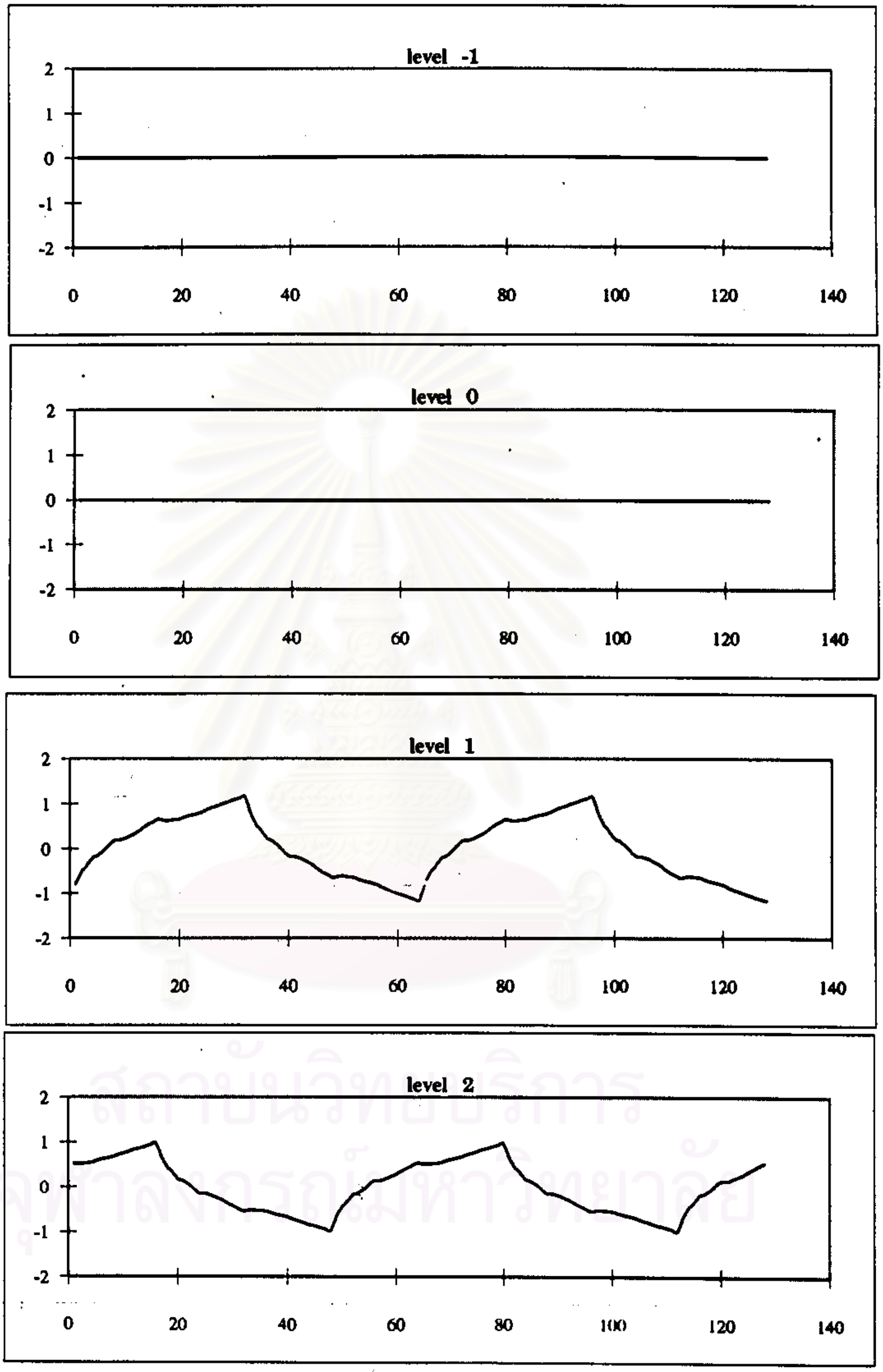
2.2 Basic Method (Newland 1994)

First we consider discrete wavelet analysis where the signal, to be analyzed is assumed to have been sampled at equally spaced intervals. If the input signal covers the range of r from 1 to N when $N = 2^n$, we have to use $n+1$ wavelet levels for the decomposition. So when $N = 128 = 2^7$, there are eight wavelet levels as shown in Figure. 2.2.



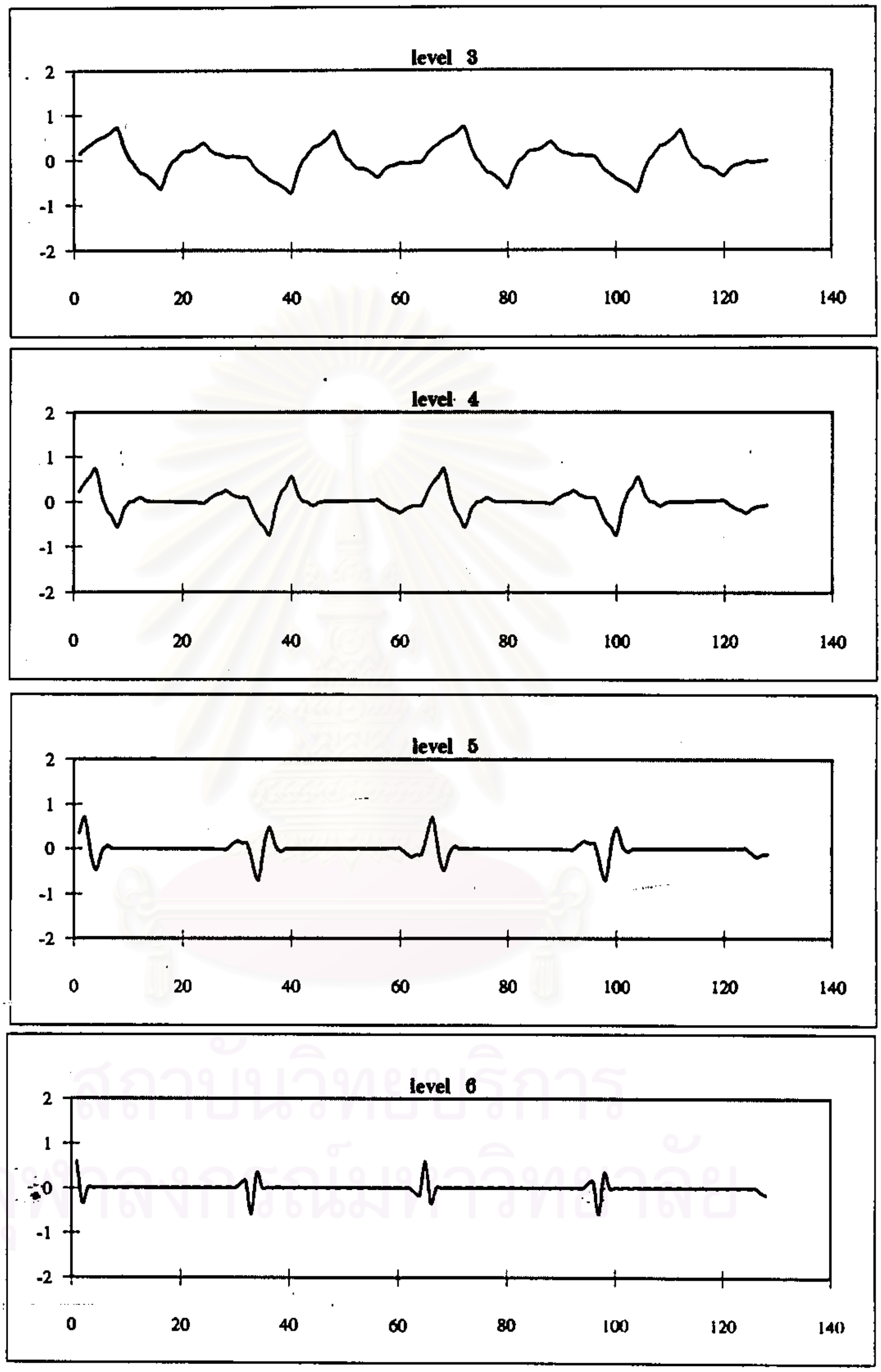
(a)

Figure 2.2 Analysis of a square wave with 128 data points, into D4 wavelet component (a) square wave, (b) and (c) wavelet in each levels



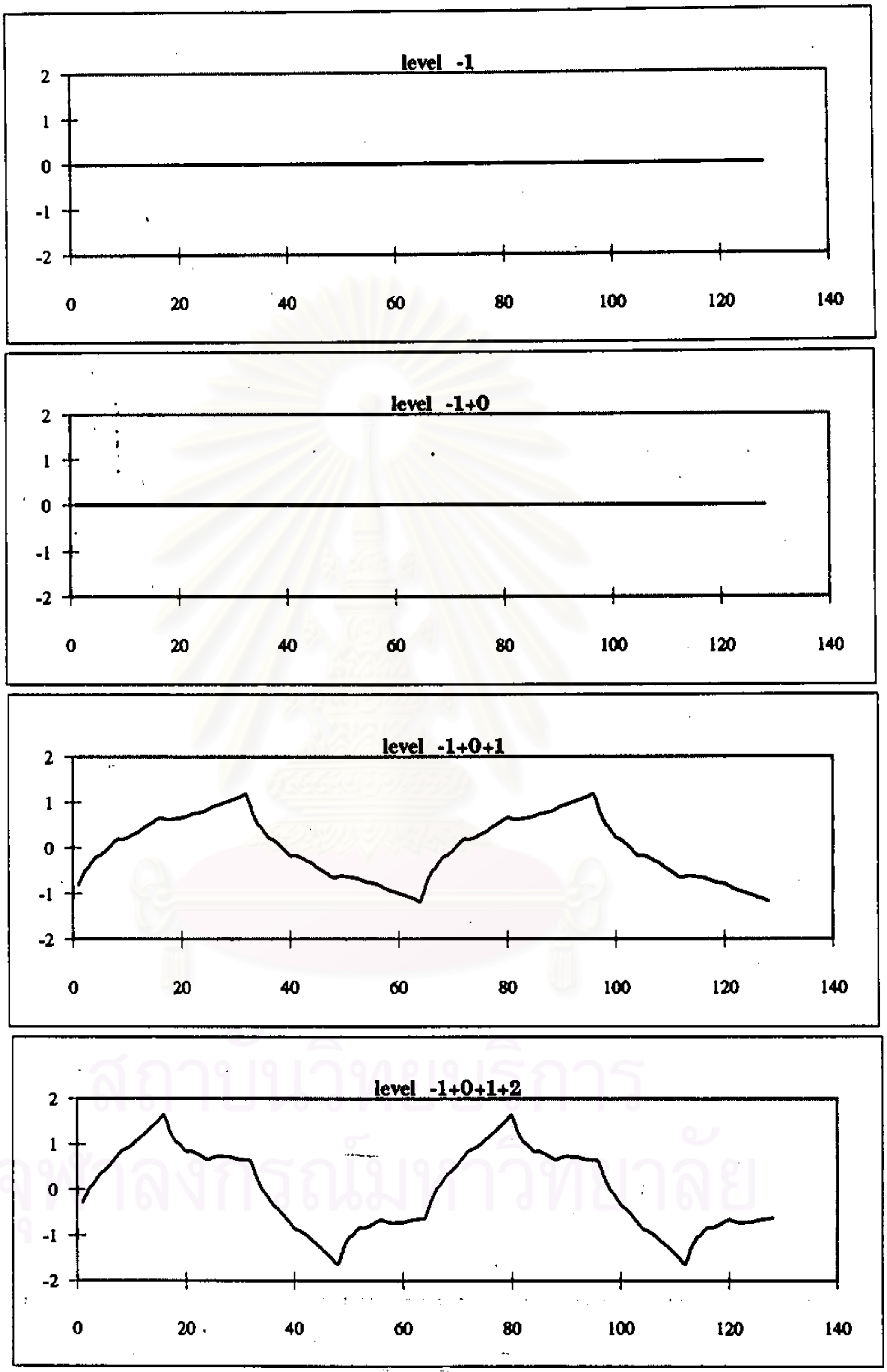
(b)

Figure 2.2 Analysis of a square wave 128 with data points, into D4 wavelet component (a) square wave, (b) and (c) wavelet in each levels



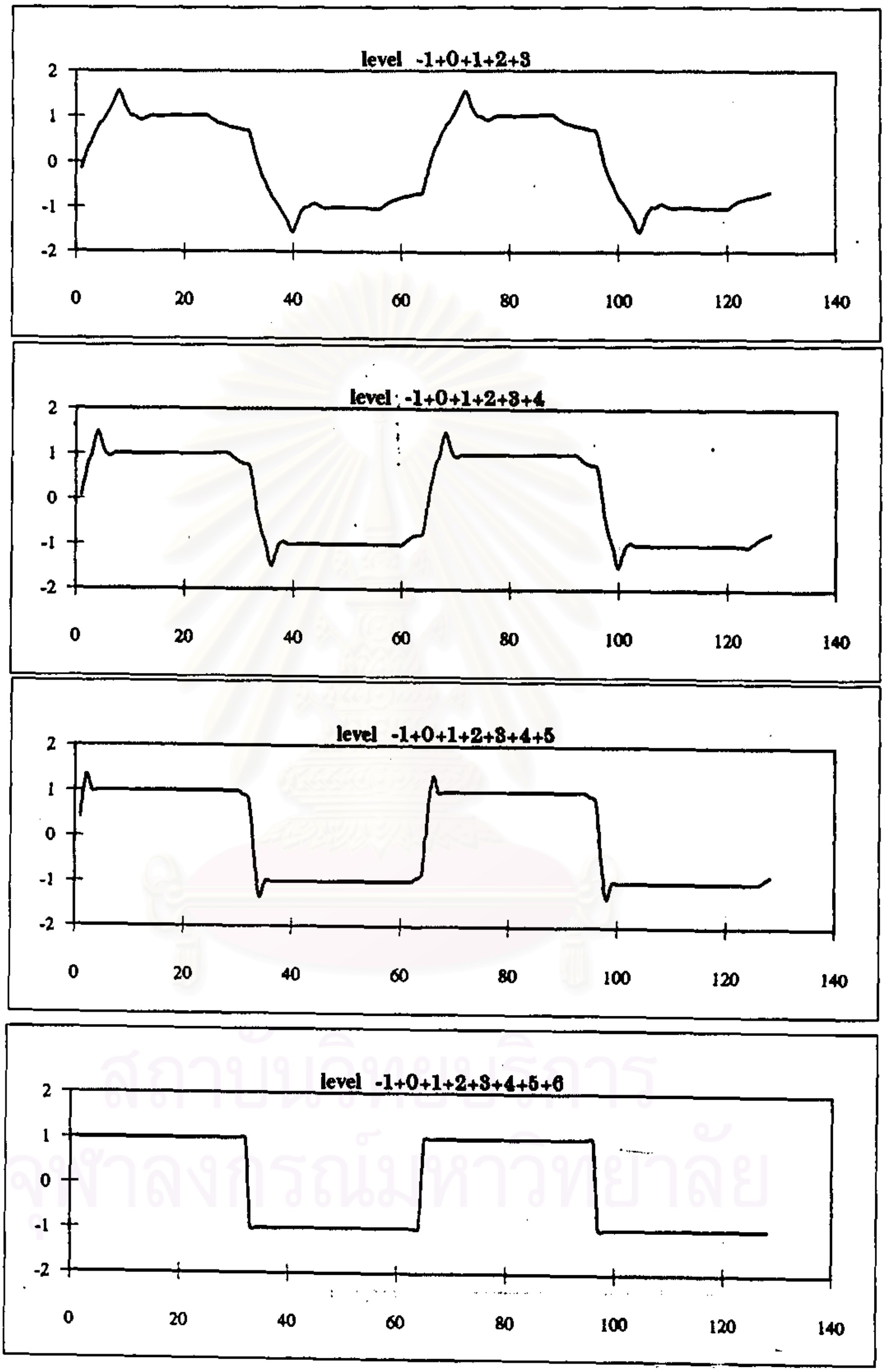
(c)

Figure 2.2 Analysis of a square wave 128 with data points, into D4 wavelet component (a) square wave, (b) and (c) wavelet in each levels



(a)

Figure 2.3 Reconstruction of the square wave from D4 wavelet components (a) and (b)



(b)

Figure 2.3 Reconstruction of the square wave from D4 wavelet components (a) and (b)

In Fig. 2.2 and Fig. 2.3, the D4 wavelet is used and the shape of this wavelet appears in the top of Fig. 2.4. It is drawn in the scale of level 3 in Fig. 2.2. Notice that in this level our wavelet basis have $3*2^4 = 48$ points which occupies only some part of the length of the signal being analyzed. To cover the whole length, additional wavelets have to be added. At level 3, there are $2^3 = 8$ wavelets along the horizontal axis. Each is placed 128/8 = 16 places with respect to its neighbor. Three adjacent wavelets are shown superimposed in the second view in Fig. 2.4, and added together in the third view. The bottom view in Fig. 2.4 shows all eight wavelets at this scale added together.

In the next higher level, at level 4, in Fig. 2.2, our wavelet basis becomes $3*2^3 = 24$ points and there are 16 wavelets placed 8 spaces apart. For level 5, the wavelet basis becomes $3*2^2$ points and there are 32 wavelets spaced 4 apart. At the highest level, with $3*2^1$ points basis, there are 64 wavelets spaced 2 apart. Level 0, has a single wavelet and level -1, has a single scaling function.

The important thing for decomposing by using D4 or DN when $N \geq 4$ wavelet is the periodic extension, which does not occur in D2 or Haar wavelet. For the above example, in the level 0, our basis have $3*2^7 = 3*128$ points which are three times of the input signal. So we have to extend the input signal three times and finally we wrapped around.

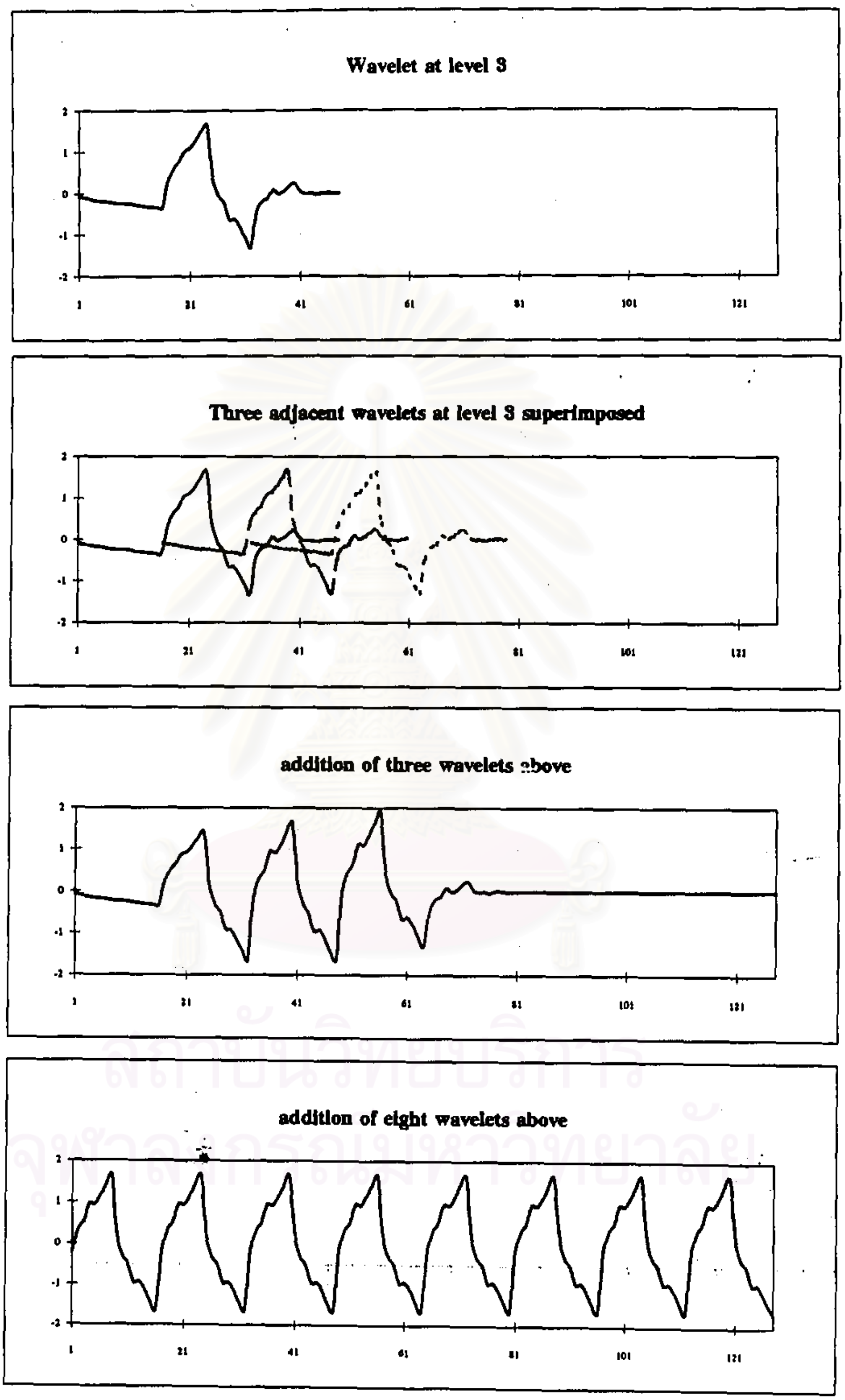


Figure 2.4 D4 wavelets at the scale of level 3 in Figure 2.2

2.3 Filter Coefficients Method

From section 1.8.1 it has been shown that the characteristic features of the scaling function and wavelet depend on the set of filter coefficients, so we can use the set of filter coefficients as a tool to decompose function like the basis function. This method was discovered by Daubechies (Daubechies, 1988) and the source code has been shown in Numerical Recipes in C by Press, W.H, et al., 1992.

The algorithm for numerically calculating the discrete wavelet transform (DWT) is similar to that of the Fast Fourier Transform (FFT) algorithm. It can be represented as a permutation of data elements followed by matrix operation. When this process is carried out iteratively, we get a corresponding transform. Similarly to Fast Fourier Transform, the data must be of length 2^n, $n \in N$, and the length of wavelet transform must also be 2^n.

2.3.1 Forward Discrete Wavelet Transform

The DWT differs from the FFT in its matrix operation. We can draw a diagram which should make the operation of the DWT clearer. The algorithm acting on a data of length 8 is represented in Figure. 2.5. The filter operation can be represented as a matrix operation and consists of the application of objects know as *quardrature mirror filter* (QMF). The elements of column vector (VI) correspond to $a_0, a_1, a_2, \ldots$which are the coefficient of the wavelet expansion.

The form of the QMF depends on the basis for the decomposition. In D4 case, the high pass QMF will be of form $[c_3, -c_2, c_1, -c_0]$ that corresponds to coefficients of the wavelet which represents the data's "detail" information, and the low pass QMF will be of from $[c_0, c_1, c_2, c_3]$ that corresponds to coefficients of the scaling function which represents the data's "smooth" information. These are mutually orthogonal. The filter matrix for a length 8 data is

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & & & & \\ c_3 & -c_2 & c_1 & -c_0 & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & \\ & & c_3 & -c_2 & c_1 & -c_0 & & \\ & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & & & & & c_0 & c_1 \\ c_1 & -c_0 & & & & & c_3 & -c_2 \end{bmatrix} \tag{2.6}$$

The very important point is that the matrix wraps around in the last two rows. This is mathematically how we extend the wavelet basis elements periodically. Operation of this matrix on column (I) return column (II) in Fig 2.5. The permutation rearranges the new data elements into "low" and "high" frequency components. This results in column (III). Operation on the lower half of column (I) is now complete. The same operation is now carried out on the top half of column(IV). In this case, the data are of length 4, and the filter operation is given by the matrix.

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \\ c_3 & -c_2 & c_1 & c_3 \\ c_2 & c_3 & c_0 & c_1 \\ c_1 & -c_0 & c_3 & c_3 \end{bmatrix} \tag{2.7}$$

Operation with this matrix on column(III) gives column(IV). A permutation carried out on column(IV) gives column(V). We then perform a filter operation on the top two elements of column(V). The filter matrix for this I also wrapped around, and using the values for the filter coefficients in the D4 case yields [see Appendix C].

$$\begin{bmatrix} c_0+c_2 & c_1+c_3 \\ c_3+c_1 & -c_2-c_0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.8}$$

This is the final operation, and column(VI) is the DWT of the original data.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix}_{(I)} \xrightarrow{\text{Filter}} \begin{bmatrix} s_0 \\ d_0 \\ s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \end{bmatrix}_{(II)} \xrightarrow{\text{Permute}} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}_{(III)} \xrightarrow{\text{Filter}} \begin{bmatrix} s'_0 \\ s'_2 \\ s'_1 \\ s'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}_{(IV)} \xrightarrow{\text{Permute}} \begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}_{(V)} \xrightarrow{\text{Filter}} \begin{bmatrix} s''_0 \\ s''_1 \\ s'_2 \\ s'_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}_{(VI)}$$

Figure 2.5 Pictorial representation of algorithm the DWT. The operation consists of the recursive application of a filter followed by permutation.

2.3.2 Inverse Discrete Wavelet Transform

We can define the *inverse discrete wavelet transform* in a similar fashion to way in which we defined the forward discret wavelet transform. In this case, the matrix or filter operation is given by the inverse of filter operation used in the forward transform. It is easy to show that the filter operation is unitary by using the properties of the filter coefficients. Then the inverse of filter operation is the transpose of forward of filter operation. Using D4 the discrete wavelet transform and the inverse wavelet transform when input data are two cycles square wave as section 2.2, has been shown in the Appendix C.

2.4 Two Dimensional Discrete Wavelet Transform (Meyer 1993)

The discrete wavelet transform can be extended to two dimensions in the same way as the one dimensional discrete Fourier Transform was extended to two dimensions. The two dimensional expansion has basis as $\phi(x)\phi(y)$, $\phi(x)\psi(y)$, $\psi(x)\phi(y)$, $\psi(x)\psi(y)$, $\phi(x)\psi(2y)$, $\phi(x)\psi(2y-1)$ and so on. Therefore a two dimensional function can be written as

$$f(x,y) = W(x)AW'(y) \tag{2.9}$$

where

$$W(x) = [\phi(x)\ \psi(x)\ \psi(2x)\ \psi(2x-1)\ \psi(4x) \ldots \psi(2^j x-k) \ldots] \tag{2.10}$$

$$W(y) = [\phi(y)\ \psi(y)\ \psi(2y)\ \psi(2y-1)\psi(4y) \ldots \psi(2^j y-k) \ldots] \quad (2.10)$$

and

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & \ldots \\ a_{10} & a_{11} & a_{12} & a_{13} & & \\ a_{20} & a_{21} & & & & \\ & & & & & \end{bmatrix} \tag{2.11}$$

If the one dimensional data lengths are 2^{n_1} and 2^{n_2}, then the order of matrix A are $2^{n_1} * 2^{n_2}$.

If the function $f(x,y)$ can be represented by a two dimensional matrix $F(2^{n_1} * 2^{n_2})$, its two dimensional wavelet transform $A(2^{n_1} * 2^{n_2})$ is calculated by repeated the one dimensional discrete wavelet transform, like the way the two dimensional discrete Fourier transform is calculated by repeated the one dimensional discrete Fourier transform. The inverse two dimensional discrete wavelet transform is calculated in the same way, beginning with the A-matrix in (2.11) and making repeated use of the one dimensional inverse discrete wavelet transform.

There is an interesting thing when we compare A matrix to Figure 2.5 in two dimensions. The matrix A can be divided into submatrices or subbands in signal analysis. Figure 2.6 shows the original image with 128*128 points and its discrete wavelet transform 128*128 matrices, A., by using D4 wavelet. Figure 2.7 shows successively higher levels of reconstruction of the image. In case that each all the elements of matrix A is set equal to zero except for a submatrix in the top left-hand corner in (2.11). The order of non zero submatrix of the element of A is progressively increased to cover the range 4*4, 8*8, 16*16, 32*32 and 64*64.

Figure 2.6 Two dimensional image with 128*128 points and its discrete wavelet transform.

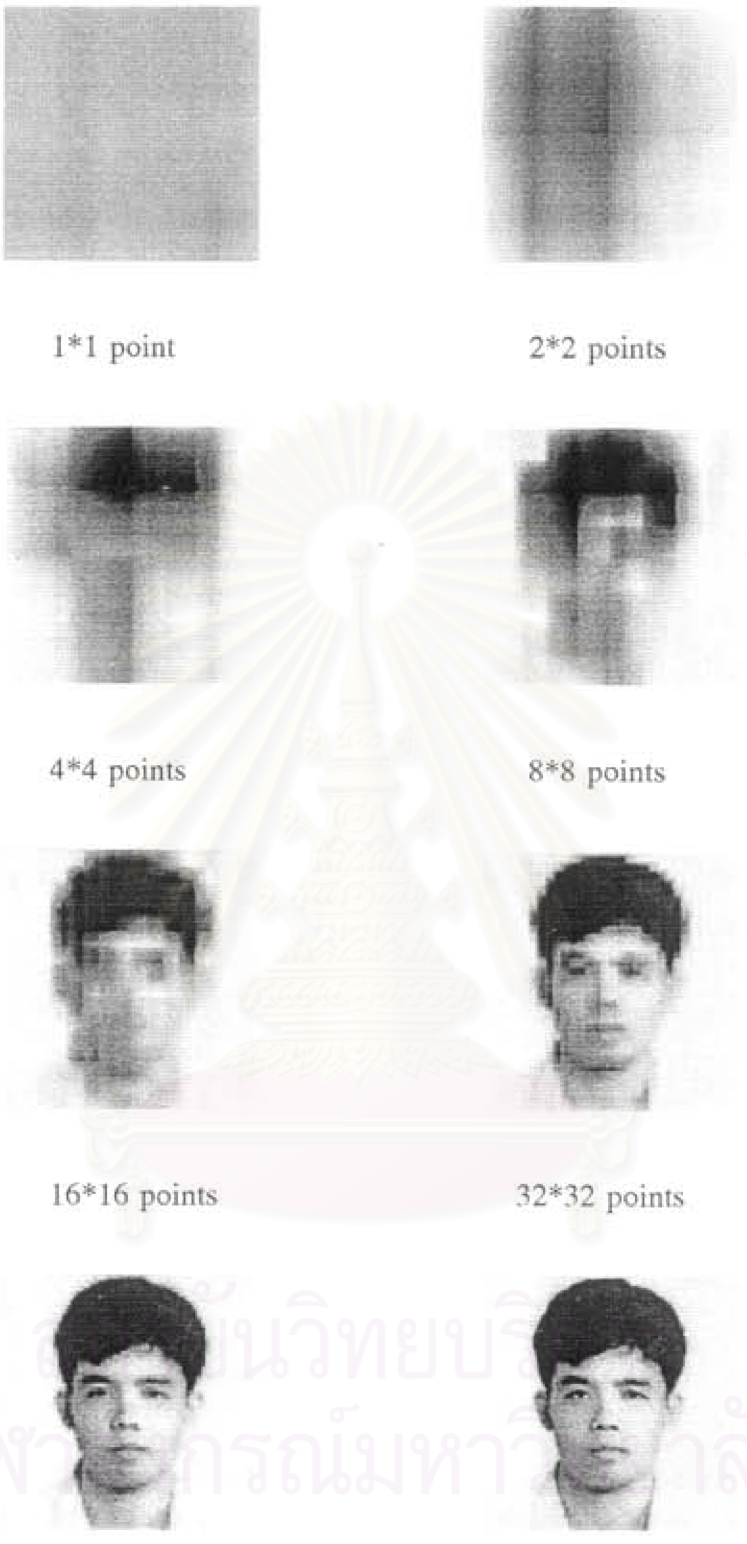


Figure 2.7 Progressive level of reconstruction of image calculated by D4 wavelet with non zero submatrix 4*4, 8*8, 16*16, 32*32, 64*64, 128*128.