Proceedings of the 8th
World Congress on Intelligent Control and Automation
July 6-9 2010, Jinan, China

2-D Wavelet Transforms in the Form of Matrices and Application in Compressed Sensing *

Huiyuan Wang [a, b] and José Vieira [a]

[a.] *IEETA, Departamento de Electrónica, Telecomunicações e Informática*
Universidade de Aveiro
3810-193 Aveiro, Portugal
jnvieira@ua.pt

[b.] *School of Information Science and Engineering*
Shandong University
Jinan, Shandong 250100, China
huiyuan@ua.pt, hywang@sdu.edu.cn

Abstract **– As a signal analysis and processing method, wavelet transform (WT) plays an important role in almost all the areas in engineering today. However, compared to other traditional orthogonal transforms, such as DFT and DCT, The usually used fast wavelet transform (FWT) has its inconvenience in application. One frequently met problem is that FWT is rarely realized in the form of linear transformation by matrix and vector multiplication, which is the form that almost all the other existing orthogonal transforms take. That is because FWT dose not usually have an explicit transform matrix. As a result, FWT cannot be used in some cases where an explicit transform matrix is required. In this paper, we explore the matrix forms of 2-D discrete wavelet transform (DWT) and apply one of them in compressed sensing (CS). Our contribution is in two aspects: we give the equivalent 2-D DWT matrix that can be used to perform the 2-D DWT in the matrix form of 1-D DWT; meanwhile, we propose a separable 2-D DWT that is different from the traditional one and has some good properties.**

Index Terms – Wavelet transform; DWT; Transform matrix; Separable transform; Compressed sensing.

I. INTRODUCTION

Wavelet transform (WT) or Wavelet analysis has become an important mathematical tool in many areas of engineering. The main implementation scheme used in signal and image analysis and processing is the fast wavelet transform (FWT) [1]. Fig. 1 shows the 1-D FWT block diagram. The inverse transform is shown in Fig. 2.

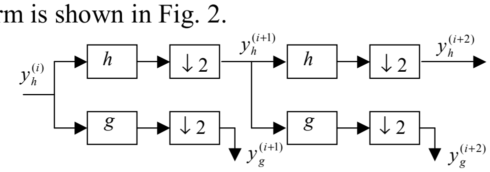

Fig. 1 A 2-scale 1-D fast wavelet transform block diagram.

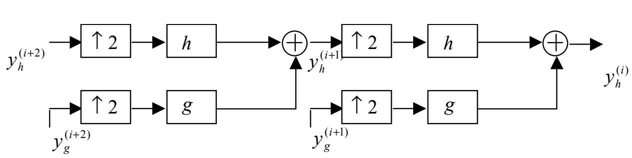

Fig. 2 A 2-scale 1-D fast wavelet inverse transform block diagram.

It is seen from Fig. 1 and Fig. 2 that, different from other existing orthogonal linear transforms like DFT and DCT, FWT is realized by iteratively filtering and down sampling to complete multi-scale decomposition. In this process, only the output of the low-pass filter becomes the input of the next scale and the output of the high-pass filter becomes a part of the final output directly. Another point that should be noted is that, to ensure completely invertible transform, the output of each stage is usually longer than the input of the same stage because of the convolution property of the filtering process. All above factors make FWT very different from ordinary linear transforms, such as that there is not usually an explicit transform matrix and the number of output elements is not equal to that of the input.

In some applications, a transform in the form of linear transformation by matrix and vector multiplication is required. For instance, in the recently appeared compressed sensing (CS) technique [2-4], DWT is preferred as a good sparsifying transform because of its good energy compacting characteristics. However, some l_1-norm minimization solvers, such as those using linear programming algorithms [5] frequently used to reconstruct sparse signals in CS can not be implemented without an explicit transform matrix.

In this paper, we present the linear transformation form of the 2-D FWT and, at the same time, define a kind of separable 2-D wavelet transform based on the work of [6,7]. We compare the characteristics of the two kind of 2-D wavelet transforms and give the experimental results.

II. 1-D DISCRETE WAVELET TRANSFORM AND ITS MATRIX FORM

Before give the matrix form of 2-D DWT in the next section and present a new separable 2-D DWT in section IV, we first discuss in this section the linear transform formulation of 1-D DWT given in [6].

A. 1-D DWT

Given a signal x of length $N = 2^n$ and a wavelet filter (low-pass filter) h, the i^th^ scale wavelet decomposition is defined as [1,6]

* This work is supported in part by FCT, Portugal, under Ciência 2007 program, the National Natural Science Foundation of China under grant no. 60872119 and the Natural Science Foundation of Shandong Province under grant no. 2009ZRB01675.

978-1-4244-6712-9/10/$26.00 ©2010 IEEE

$$y_h^{(i)}(j)=\sum_{k=1}^{2^{n-i+1}} h(k-2j)y_h^{(i-1)}(k) \tag{1}$$
$$j=1,2,\cdots,2^{n-i}$$

and

$$y_g^{(i)}(j)=\sum_{k=1}^{2^{n-i+1}} g(k-2j)y_h^{(i-1)}(k) \tag{2}$$
$$j=1,2,\cdots,2^{n-i}$$

where, g is the high-pass filter corresponding to h and can be formed by multiplying (-1) with every even sample point of h; both filters are usually zero padded to become periodical ones with the length of current signal to be decomposed as the period; K is the number of non-zero points in h and g, and $y_h^0 = x$.

The reconstruction of y_h^{i-1} from y_h^i and y_g^i can be obtained by

$$y_h^{(i-1)}(j)=\sum_{k=1}^{2^{n-i}} h(j-2k)y_h^{(i)}(k)+\sum_{k=1}^{2^{n-i}} g(j-2k)y_g^{(i)}(k) \tag{3}$$
$$j=1,2,\cdots,2^{n-i+1}$$

B. Matrix Form of 1-D DWT

It is readily seen that Eq. (1) and (2) can be represented in matrix forms as [6,7]

$$\mathbf{y}_h^{(i)}=\mathbf{H}^{(i)}\mathbf{y}_h^{(i-1)} \tag{4}$$

and

$$\mathbf{y}_g^{(i)}=\mathbf{G}^{(i)}\mathbf{y}_h^{(i-1)} \tag{5}$$

where, $\mathbf{y}_h^{(i)}$ is the 2^{n-i} dimensional low-pass vector in the i[th] scale and $\mathbf{y}_g^{(i)}$ the high-pass one, while $\mathbf{y}_h^{(i-1)}$ is the 2^{n-i+1} dimensional low-pass vector in the (i-1)[th] scale. The two 2^{n-i} by 2^{n-i+1} filter matrices are respectively

$$\mathbf{H}^{(i)}=\begin{bmatrix} h(-1) & 0 & 0 & 0 & \cdots & h(-3) & h(-2) \\ h(-3) & h(-2) & h(-1) & 0 & \cdots & h(-5) & h(-4) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & & \cdots & h(-1) & 0 \end{bmatrix} \tag{6}$$

and

$$\mathbf{G}^{(i)}=\begin{bmatrix} g(-1) & 0 & 0 & 0 & \cdots & g(-3) & g(-2) \\ g(-3) & g(-2) & g(-1) & 0 & \cdots & g(-5) & g(-4) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & & \cdots & g(-1) & 0 \end{bmatrix} \tag{7}$$

Therefore, the transform in the i[th] scale can be rewritten as

$$\begin{bmatrix} \mathbf{y}_h^{(i)} \\ \mathbf{y}_g^{(i)} \end{bmatrix}=\begin{bmatrix} \mathbf{H}^{(i)} \\ \mathbf{G}^{(i)} \end{bmatrix}\mathbf{y}_h^{(i-1)} \tag{8}$$

This is only the matrix formulation for a single scale. We have to consider that the current scale is the result of decomposing from the original signal through to the maximum scale. This can be achieved by using the following relation.

$$\mathbf{y}_h^{(i-1)}=\mathbf{H}^{(i-1)}\mathbf{y}_h^{i-2} \tag{9}$$

By continuous substitution, we have

$$\begin{bmatrix} \mathbf{y}_h^{(i)} \\ \mathbf{y}_g^{(i)} \\ \mathbf{y}_g^{(i-1)} \\ \vdots \\ \mathbf{y}_g^{(2)} \\ \mathbf{y}_g^{(1)} \end{bmatrix}=\begin{bmatrix} \mathbf{H}^{(i)}\mathbf{H}^{(i-1)}\cdots\mathbf{H}^{(1)} \\ \mathbf{G}^{(i)}\mathbf{H}^{(i-1)}\cdots\mathbf{H}^{(1)} \\ \mathbf{G}^{(i-1)}\mathbf{H}^{(i-2)}\cdots\mathbf{H}^{(1)} \\ \vdots \\ \mathbf{G}^{(2)}\mathbf{H}^{(1)} \\ \mathbf{G}^{(1)} \end{bmatrix}\mathbf{x} \tag{10}$$

where, in the left, the 2^n dimensional vector consists of the low-pass part of the i[th] scale and the high-pass wavelet coefficients from all the scales since the first.

It should be pointed out that, due to the length of the filter h, the maximum number of levels that the 2^n long signal can be decomposed is limited to $2^{n-i+1} \geq l$, where, l is the length of the filter.

The transform matrix is thus given by

$$\mathbf{W}=\begin{bmatrix} \mathbf{H}^{(i)}\mathbf{H}^{(i-1)}\cdots\mathbf{H}^{(1)} \\ \mathbf{G}^{(i)}\mathbf{H}^{(i-1)}\cdots\mathbf{H}^{(1)} \\ \mathbf{G}^{(i-1)}\mathbf{H}^{(i-2)}\cdots\mathbf{H}^{(1)} \\ \vdots \\ \mathbf{G}^{(2)}\mathbf{H}^{(1)} \\ \mathbf{G}^{(1)} \end{bmatrix} \tag{11}$$

The transpose of $\mathbf{W}$ will give the inverse transform matrix.

III. 2-D Discrete Wavelet Transform and Its Matrix Form

A. 2-D DWT

The 2-D wavelet transform is achieved by applying the 1-D transform in the vertical direction and then in the horizontal direction respectively for the same scale. Specifically, given an image x of size $M \times N = 2^m \times 2^n$, the i[th] scale decomposition can be achieved by Eq. (12) through (15)

$$y_{hh}^{(i)}(j_1,j_2)=\sum_{l=1}^{2^{n-i+1}}\left[\sum_{k=1}^{2^{m-i+1}} h(k-2j_1)y_{hh}^{(i-1)}(k,l)\right]h(l-2j_2) \tag{12}$$
$$j_1=1,2,\cdots,2^{m-i};\ j_2=1,2,\cdots,2^{n-i}$$

$$y_{hg}^{(i)}(j_1,j_2)=\sum_{l=1}^{2^{n-i+1}}\left[\sum_{k=1}^{2^{m-i+1}} h(k-2j_1)y_{hh}^{(i-1)}(k,l)\right]g(l-2j_2) \tag{13}$$
$$j_1=1,2,\cdots,2^{m-i};\ j_2=1,2,\cdots,2^{n-i}$$

$$y_{gh}^{(i)}(j_1,j_2)=\sum_{l=1}^{2^{n-i+1}}\left[\sum_{k=1}^{2^{m-i+1}} g(k-2j_1)y_{hh}^{(i-1)}(k,l)\right]h(l-2j_2) \tag{14}$$
$$j_1=1,2,\cdots,2^{m-i};\ j_2=1,2,\cdots,2^{n-i}$$

and

$$y_{gg}^{(i)}(j_1,j_2)=\sum_{l=1}^{2^{n-i+1}}\left[\sum_{k=1}^{2^{m-i+1}} g(k-2j_1)y_{hh}^{(i-1)}(k,l)\right]g(l-2j_2) \tag{15}$$
$$j_1=1,2,\cdots,2^{m-i};\ j_2=1,2,\cdots,2^{n-i}$$

Similarly to the 1-D case, the reconstruction of y_{hh}^{i-1} from y_{hh}^i, y_{hg}^i, y_{gh}^i and y_{gg}^i can be obtained by

$$
\begin{aligned}
y_{hh}^{(i-1)}(j_1,j_2) &= \sum_{l=1}^{2^{n-i}}\left[\sum_{k=1}^{2^{m-i}} h(j_1-2k)y_{hh}^{(i)}(k,l)\right]h(j_2-2l) \\
&+\sum_{l=1}^{2^{n-i}}\left[\sum_{k=1}^{2^{m-i}} h(j_1-2k)y_{hg}^{(i)}(k,l)\right]g(j_2-2l) \\
&+\sum_{l=1}^{2^{n-i}}\left[\sum_{k=1}^{2^{m-i}} g(j_1-2k)y_{gh}^{(i)}(k,l)\right]h(j_2-2l) \\
&+\sum_{l=1}^{2^{n-i}}\left[\sum_{k=1}^{2^{m-i}} g(j_1-2k)y_{gg}^{(i)}(k,l)\right]g(j_2-2l) \\
j_1 &= 1,2,\cdots,2^{m-i+1};\ j_2=1,2,\cdots,2^{n-i+1}
\end{aligned} \tag{16}
$$

B. Matrix Form of 2-D DWT

Reordering each of the sub-matrices in $\mathbf{Y}$ in Eq. (12-15) , we obtain

$$
\begin{bmatrix} \mathbf{Y}_{hh} \\ \mathbf{Y}_{hg} \\ \mathbf{Y}_{gh} \\ \mathbf{Y}_{gg} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_x\mathbf{X}\mathbf{H}_y^t \\ \mathbf{H}_x\mathbf{X}\mathbf{G}_y^t \\ \mathbf{G}_x\mathbf{X}\mathbf{H}_y^t \\ \mathbf{G}_x\mathbf{X}\mathbf{G}_y^t \end{bmatrix} \tag{17}
$$

Representing $\mathbf{X}$ and $\mathbf{Y}$ in vector form by row or column expansion, Eq. (17) becomes

$$
\begin{bmatrix} \mathbf{y}_{hh} \\ \mathbf{y}_{hg} \\ \mathbf{y}_{gh} \\ \mathbf{y}_{gg} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_x\otimes\mathbf{H}_y \\ \mathbf{H}_x\otimes\mathbf{G}_y \\ \mathbf{G}_x\otimes\mathbf{H}_y \\ \mathbf{G}_x\otimes\mathbf{G}_y \end{bmatrix}\mathbf{x} \tag{18}
$$

or

$$
\mathbf{y} = \mathbf{W}\mathbf{x} \tag{19}
$$

where, $\otimes$ is the Kronecker product operator. For 2-scale 2-D wavelet decomposition, Eq. (18) will become

$$
\begin{bmatrix} \mathbf{y}_{hh}^{(2)} \\ \mathbf{y}_{hg}^{(2)} \\ \mathbf{y}_{gh}^{(2)} \\ \mathbf{y}_{gg}^{(2)} \\ \mathbf{y}_{hg}^{(1)} \\ \mathbf{y}_{gh}^{(1)} \\ \mathbf{y}_{gg}^{(1)} \end{bmatrix} = \begin{bmatrix} (\mathbf{H}_x^{(2)}\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(2)}\mathbf{H}_y^{(1)}) \\ (\mathbf{H}_x^{(2)}\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(2)}\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(2)}\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(2)}\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(2)}\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(2)}\mathbf{H}_y^{(1)}) \\ (\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(1)}) \\ (\mathbf{G}_x^{(1)})\otimes(\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(1)})\otimes(\mathbf{G}_y^{(1)}) \end{bmatrix}\mathbf{x} \tag{20}
$$

In the same way, for i-scale decomposition, the above Equation will become

$$
\begin{bmatrix} \mathbf{y}_{hh}^{(i)} \\ \mathbf{y}_{hg}^{(i)} \\ \mathbf{y}_{gh}^{(i)} \\ \mathbf{y}_{gg}^{(i)} \\ \mathbf{y}_{hg}^{(i-1)} \\ \mathbf{y}_{gh}^{(i-1)} \\ \vdots \\ \mathbf{y}_{gh}^{(1)} \\ \mathbf{y}_{gg}^{(1)} \end{bmatrix} = \begin{bmatrix} (\mathbf{H}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{H}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ \vdots \\ (\mathbf{G}_x^{(1)})\otimes(\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(1)})\otimes(\mathbf{G}_y^{(1)}) \end{bmatrix}\mathbf{x} \tag{21}
$$

As a result, the wavelet transform matrix for 2-D wavelet transform in vector representation will be

$$
\mathbf{W} = \begin{bmatrix} (\mathbf{H}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{H}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(i)}\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(i)}\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{H}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{G}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(i-1)}\mathbf{H}_x^{(i-2)}\cdots\mathbf{H}_x^{(1)})\otimes(\mathbf{H}_y^{(i-1)}\mathbf{H}_y^{(i-2)}\cdots\mathbf{H}_y^{(1)}) \\ \vdots \\ (\mathbf{G}_x^{(1)})\otimes(\mathbf{H}_y^{(1)}) \\ (\mathbf{G}_x^{(1)})\otimes(\mathbf{G}_y^{(1)}) \end{bmatrix} \tag{22}
$$

IV. Separable 2-D Discrete Wavelet Transform

A. Formulation

It is natural to consider the possibility of achieving the 2-D formulation of 2-D wavelet transform by using the 1-D transform first to the vertical direction and then to the horizontal direction respectively just like that in DFT.

This is correct for the case of single scale 2-D wavelet decomposition.

Given an image matrix $\mathbf{X}$ of size $M\times N = 2^m\times 2^n$ and transform matrices $\mathbf{W}_x$ and $\mathbf{W}_y$ defined in Eq.(11), we have

$$
\mathbf{Y} = \mathbf{W}_x\mathbf{X}\mathbf{W}_y^t \tag{23}
$$

or specifically,

$$
\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{hh} & \mathbf{Y}_{hg} \\ \mathbf{Y}_{gh} & \mathbf{Y}_{gg} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_x \\ \mathbf{G}_x \end{bmatrix}\mathbf{X}\begin{bmatrix} \mathbf{H}_y^t & \mathbf{G}_y^t \end{bmatrix} \tag{24}
$$

Certainly, the above formulation also applies to the decomposition from the (i-1)th scale to the ith scale.

For single scale wavelet analysis, this separable decomposition is the same as FWT. For wavelet decomposition of more than one scale, however, above formulation will be different from FWT. For instance, to decompose an image into two scales, the result will become

$$
\begin{aligned}
&\begin{bmatrix} \mathbf{H}_x^{(2)}\mathbf{H}_x^{(1)} \\ \mathbf{G}_x^{(2)}\mathbf{H}_x^{(1)} \\ \mathbf{G}_x^{(1)} \end{bmatrix}\mathbf{X}\left[\left(\mathbf{H}_y^{(2)}\mathbf{H}_y^{(1)}\right)^t \quad \left(\mathbf{G}_y^{(2)}\mathbf{H}_y^{(1)}\right)^t \quad \left(\mathbf{G}_y^{(1)}\right)^t\right] \\
&= \begin{bmatrix} \mathbf{Y}_{(hh)(hh)} & \mathbf{Y}_{(hh)(hg)} & \mathbf{Y}_{(hg)(h0)} \\ \mathbf{Y}_{(hh)(gh)} & \mathbf{Y}_{(hh)(gg)} & \mathbf{Y}_{(hg)(g0)} \\ \mathbf{Y}_{(gh)(0h)} & \mathbf{Y}_{(gh)(0g)} & \mathbf{Y}_{(gg)} \end{bmatrix}
\end{aligned} \tag{25}
$$

It is clear from above formulation, that the two coefficient sub-matrices $\mathbf{Y}_{hg}$ and $\mathbf{Y}_{gh}$ are further decomposed, which should have been remain unchanged in the standard definition of 2-D FWT. The example in Fig. 3 is an illustration of this decomposition

Therefore, this is a new kind of wavelet transform. Here we define it as separable wavelet transform (SWT).

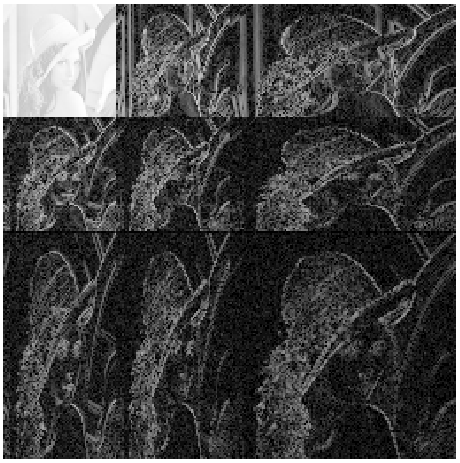

Fig. 3 An example of SWT.

B. Some Explanations

SWT is different from the traditional FWT. It has some special characteristics. By analysis and experiment, we can summarize its advantages and disadvantages as follows.

(a) Matrix formulation – easy to be realized in conventional way of matrix multiplication.

(b) Constant size before and after the transform – no memory problem is caused in processing.

(c) Separable transform – huge transform matrix (Eq. (22)) can be avoided for big images.

(d) Energy distribution of the coefficients. We observed by experimental results shown in Fig. 4 and Fig. 5 that, when the filter length is less than 3, SWT is the same as FWT in energy distribution of the coefficients. That means it has similar properties to the FWT useful for such applications as image compression, de-noising and feature extraction. However, if the filter length is increased, its energy distribution is not as good as that of FWT.

(e) Because of the boundary effect in filtering, the maximum scales that an image can be decomposed by SWT is limited by some condition mentioned in section II.

V. Application of SWT in 2-D CS

In 2-D CS [8-9], the sampling procedure can be represented by $\mathbf{Y}=\mathbf{\Phi}_1\mathbf{X}\mathbf{\Phi}_2^T=\mathbf{\Phi}_1\mathbf{\Psi}_1\mathbf{S}\mathbf{\Psi}_2^T\mathbf{\Phi}_2^T$, where, $\mathbf{Y}$ is the matrix of measurement samples, $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ are the sampling matrices, whose elements are usually drawn from an i.i.d Gaussian and $\mathbf{X}$ the 2-D image, which is sparse in the transformed domain represented by matrix $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$, and $\mathbf{S}$ is the representation of $\mathbf{X}$ in the transformed domain.

Now that we have defined the separable 2-D wavelet transform SWT, the reconstruction matrices can be found as $\mathbf{\Theta}_1=\mathbf{\Phi}_1\mathbf{\Psi}_1=\mathbf{\Phi}_1\mathbf{W}_x$ and $\mathbf{\Theta}_2=\mathbf{\Phi}_2\mathbf{\Psi}_2=\mathbf{\Phi}_2\mathbf{W}_y$, where $\mathbf{W}_x$ and $\mathbf{W}_y$ are the SWT transform matrices defined in Eq. (11).

VI. Experimental Results

We conducted the experiment in two parts. In the first part, we confirm the correctness of the proposed matrix form of 2-D DWT and observe their relevant properties by performing them on typical real images. We performed forward and inverse transforms of an image using the matrix form given in section III or the SWT proposed in section IV, and compare the results with those of FWT in image domain in terms of the error between the image got from the inverse transform and the original one. It is observed that no extra error was introduced by the proposed transforms under the condition about the maximum scales mentioned above.

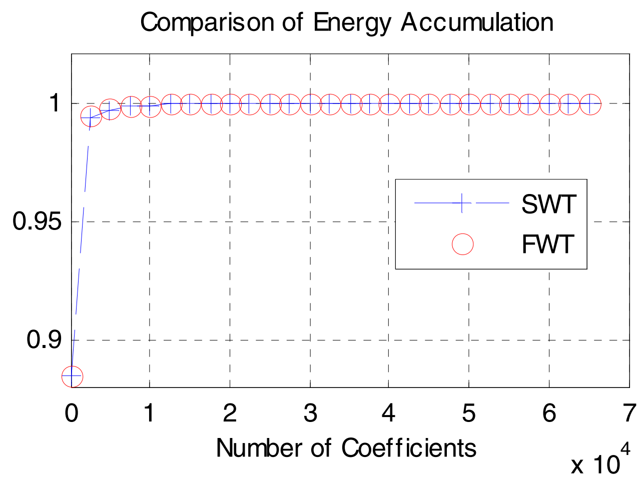

Fig. 4 Coefficient energy distribution of FWT and SWT using Haar filter.

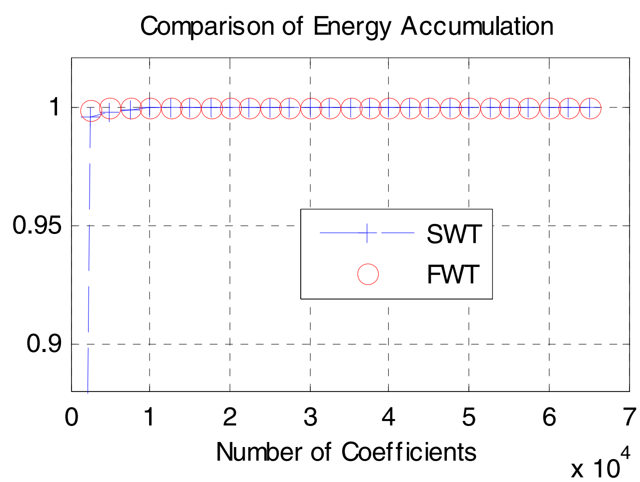

Fig. 5 Coefficient energy distribution of FWT and SWT using DB2 filter.

For SWT, we also compared its energy distribution with that of FWT to examine its characteristics in energy compactness. Fig. 4 and Fig. 5 show some examples of this comparison, where a standard 256×256 gray level image was transformed using FWT and SWT respectively and the coefficients were sorted in a descent order, and then the ratio of the energy contained in the first certain number of largest coefficients over the total energy is shown as the curve value. The difference between Fig. 4 and Fig. 5 is that Haar filter was used in Fig. 4, while DB2 filter was used in Fig. 5.By careful observation, it can be found that SWT is not as compact as FWT in Fig. 4.

In the second part, we used SWT in the 2-D CS scheme proposed in [8-9] to reconstruction the original image sparsified in SWT domain with an l_1 -norm minimization solver SPGL1 [10]. Fig. 5 (a) shows the original 512×512 Lena image; (b) shows the sparsified image by only keep the 2.5% maximum coefficients in the SWT domain; and (c) shows the reconstructed image from only 256×256 measurements using 2-D CS. The peak signal to noise ratio (PSNR) achieved is 76.7dB.

Fig. 6 Images sparsified and reconstructed in SWT domain using 2-D CS.

VII. CONCLUSIONS

We proposed the matrix forms of 2-D discrete wavelet transform (DWT) in this paper and applied the separable 2-D form in compressed sensing (CS). The matrix form of 2-D DWT is equivalent to its original FWT form if certain condition on the filter length discussed in section II is satisfied. While the separable 2-D matrix form has some special advantages discussed in section IV over the traditional 2-D FWT and the experimental results in section VI verified the effectiveness of its application in compressed sensing.

ACKNOWLEDGMENT

The authors would like to thank Qiuhui Chen and Guangbin Ren for many helpful discussions on wavelet theories.

REFERENCES

[1] Stéphane Mallat, A Wavelet Tour of Signal Processing, Academic Press, 1998, pp254, 309.

[2] D. L. Donoho, “Compressed sensing,” IEEE Trans. Inf. Theory, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] E. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Trans. Inf. Theory, vol. 52, no. 2, pp. 489–509, 2006.

[4] R. G. Baraniuk, “Compressive sensing,” IEEE Signal Processing Magazine, vol. 24, no. 4, pp. 118 - 121, 2007.

[5] D. L. Donoho and J. Tanner, “Sparse nonnegative solution of underdetermined linear equations by linear programming,” in *Proc. Nat.Academy Sci.*, 2005, pp. 9446–9451.

[6] Brani Vidakovic, Statistical Modeling By Wavelets, Wiley, 1999, pp 115-116.

[7] Brani Vidakovic, 2004, http://www2.isye.gatech.edu/~brani/wavelet.html .

[8] Y. Rivenson, A. Stern, “Compressed imaging with a separable sensing operator,” IEEE Signal Processing Letters, Vol. 16, no. 6, pp.449 – 452, June 2009.

[9] Y. Rivenson, A. Stern, “Practical compressive sensing of large images,” 2009 16th International Conference on Digital Signal Processing, pp. 1-8, July 2009.

[10] E. Berg and M.P. Friedlander, SPGL1: A solver for large-scale sparse reconstruction. Available: http://www.cs.ubc.ca/labs/scl/spgl1/, June 2007.