Noise Reduction Algorithms Using Fibonacci Fourier Transforms

Sos S. Agaian, Mei-Ching Chen, and C. L. Philip Chen
Department of Electrical and Computer Engineering
The University of Texas at San Antonio
San Antonio, Texas, USA
sagaian@utsa.edu, axf710@my.utsa.edu, pchen@utsa.edu

***Abstract*—This paper presents the new Fibonacci Fourier-like transforms. The proposed transforms render the relationship between Fibonacci numbers and the conventional Discrete Fourier Transform. The fast Fibonacci Fourier transforms are also introduced with the use of the Kronecker product properties. The proposed transforms are applied to the problem of noise reduction with two new algorithms, sliding double window filtering and fusion sliding window filtering. The primary concept of sliding double window filtering is to process the noisy signals with nonoverlapped windows, while the primary concept of fusion sliding window filtering is to process the noisy signals with various weighted filtering methods and overlapped signal values. The results and analysis show the noise reduction of the given noisy gray level images. The proposed methods are compared with the well-known Wiener filtering using images that contain Gaussian noise with the range of variance between 0 and 0.3. The analysis shows by visual inspection that the noisy parts are smoothened while retaining natural edges.**

***Keywords*—Fibonacci Fourier transforms, fast Fibonacci Fourier transforms, noise reduction, sliding window filters**

I. INTRODUCTION

The discrete Fourier transform (DFT) is one of the most fundamental operations in digital signal processing (DSP) [12]. The fast Fourier transform (FFT) was developed to accelerate the computation process [2]. The computational complexity is reduced from $O(N^2)$ to $O(N\log_2 N)$ for N-point DFT. In various research areas and applications, more extended or modified transform algorithms have been investigated [3,5,13]. The developed transforms have improved the performance in many areas, such as image processing, pattern recognition, data hiding, etc [1,15,16]. For instance, in spatial domain filtering techniques, the problem is the number of computation necessary for each pixel. Transforms are used to reduce the amount of additive noise to an image while reducing the computational complexity. This paper presents a new Fibonacci Fourier-like transforms and uses for the application of Gaussian noise reduction.

In this paper, the Fibonacci Fourier-like transforms are introduced. The transforms are represented by the Fibonacci numbers or the sum of the Fibonacci numbers. It also shows that the classical Fourier transform is an approximation of the Fibonacci Fourier-like transforms. By using these transforms with sliding window filtering algorithms, an application of the new methods are demonstrated for image denoising. The sliding window filtering techniques are farther expanded by developing two variations. In order to reduce the noises, sliding double window filtering applies the outer transformed window with the inner nonoverlapped spatial window, whereas the fusion sliding window filtering is designed to integrate the advantages of various filters.

The comparisons of image denoising performance have been made between the proposed methods and Wiener filters. With additive Gaussian noisy images, the proposed methods reduce more noises than the Wiener filters by visual inspection. The sections in this paper are organized as follows: Section 2 provides a brief background in Fibonacci series and the Kronecker product. Section 3 describes the new Fibonacci Fourier transform and its fast algorithm. To reduce the noisy gray level images, two sliding window filtering techniques making use of threshold filtering discussed in Section 4. Section 5 illustrates the experimental results, and Section 6 concludes the paper.

II. BACKGROUND

A. Fibonacci Series

The Fibonacci Series was introduced in a book published by Leonardo Fibonacci in 1202 [4]. It has been exploited in many areas, such as art, design, mathematics, geometry, etc [11]. The series is defined as a sequence of numbers $f_n = f_{n-1} + f_{n-2}$, where $f_0 = 1$ and $f_n = 0$ for $n < 0$. Each number in the sequence is simply the sum of the two preceding numbers, for example, sequence $x[n] = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144$, etc. From the series, many properties are found. One of the important characteristics is the golden ratio [9] used in this paper.

By analyzing the Fibonacci series, the ratio between the adjacent numbers in the sequence converges to a certain value as the index n increases. The value is named the golden ratio denoted in (1).

$$\varphi = \lim_{n\to\infty} \frac{f_n}{f_{n-1}} = 1.618033988749895\ldots \tag{1}$$

The golden ratio can also be derived from a quadratic equation $\varphi^2 - \varphi - 1 = 0$ resulting in the positive solution shown in (2).

$$\varphi = \frac{1+\sqrt{5}}{2} \tag{2}$$

1-4244-2384-2/08/$20.00

A property related to the golden ratio is that some trigonometric functions may be characterized by the ratio [9]. The function representations used in this paper include

$$\cos\frac{\pi}{5}=\frac{\varphi}{2} \text{ and } \sin\frac{\pi}{5}=\frac{\gamma}{2} \tag{3}$$

where $\gamma=\sqrt{3-\varphi}$.

B. Kronecker Product

The Kronecker product originates from group theory and has been applied in various fields of matrix theory [7,10]. It is also known as a direct product or a tensor product and denoted as $\otimes$. Suppose two matrices are given, say, $\mathbf{A}$ and $\mathbf{B}$. The dimensions are $p\times q$ and $r\times s$ respectively. Note that $\mathbf{A}$ and $\mathbf{B}$ may be different matrices and have different dimensions. The Kronecker product of two matrices is defined as

$$\mathbf{C}=\mathbf{A}\otimes\mathbf{B}=\begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1q}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2q}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}\mathbf{B} & a_{p2}\mathbf{B} & \cdots & a_{pq}\mathbf{B} \end{bmatrix} \tag{4}$$

where a_{ij} indicates an element in the matrix $\mathbf{A}$, i = 1, 2, …, p and j = 1, 2, ..., q. The operation takes all possible multiplication between the two matrices. The output matrix $\mathbf{C}$ then has the dimension of $pr\times qs$.

In this paper, two properties of Kronecker product are applied and described in the following.

Property 1. If $\mathbf{A}\in\mathbb{R}^{p\times p}$ and $\mathbf{B}\in\mathbb{R}^{r\times r}$ are orthogonal matrices, then $\mathbf{A}\otimes\mathbf{B}$ is orthogonal indicating

$$\left(\mathbf{A}_p\otimes\mathbf{B}_r\right)\left(\mathbf{A}_p\otimes\mathbf{B}_r\right)^{-1}=\mathbf{I}_{p\times r} \tag{5}$$

Property 2. Let $\mathbf{A}\in\mathbb{R}^{p\times p}$ and $\mathbf{B}\in\mathbb{R}^{r\times r}$. Then the Kronecker product of $\mathbf{A}_p$ and $\mathbf{B}_r$ can be represented as

$$\mathbf{A}_p\otimes\mathbf{B}_r=\left(\mathbf{A}_p\otimes\mathbf{I}_r\right)\left(\mathbf{I}_p\otimes\mathbf{B}_r\right) \tag{6}$$

C. Discrete Fourier Transform

Suppose a one dimensional signal is of length N and represented as a vector $\mathbf{x}$ = [x_0, x_1, …, x_{N-1}]^T. The discrete Fourier transform of the signal outputs a transformed vector $\mathbf{y}$ by multiplying a Fourier transform matrix $\mathbf{W}_N$ to $\mathbf{x}$, refer to (7).

$$\mathbf{y}=\mathbf{W}_N\mathbf{x} \tag{7}$$

The Fourier transform matrix is structured by every element as $e^{-j2\pi nk/N}$, where n = 0, 1, …, N-1 and k = 0, 1, …, N-1. Since the exponential representation can be decomposed into a complex combination of the trigonometric functions cosine and sine, the Fourier transform matrix is a complex-valued matrix defined as

$$\mathbf{W}_N=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \cos\left(2\pi\frac{1}{N}\right) & \cdots & \cos\left(2\pi\frac{N-1}{N}\right) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cos\left(2\pi\frac{N-1}{N}\right) & \cdots & \cos\left(2\pi\frac{(N-1)^2}{N}\right) \end{bmatrix} - j\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \sin\left(2\pi\frac{1}{N}\right) & \cdots & \sin\left(2\pi\frac{N-1}{N}\right) \\ 0 & \vdots & \ddots & \vdots \\ 0 & \sin\left(2\pi\frac{N-1}{N}\right) & \cdots & \sin\left(2\pi\frac{(N-1)^2}{N}\right) \end{bmatrix} \tag{8}$$

The 5-point discrete Fourier transform matrix can be written as

$$\mathbf{W}_5=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0.3090 & -0.8090 & -0.8090 & 0.3090 \\ 1 & -0.8090 & 0.3090 & 0.3090 & -0.8090 \\ 1 & -0.8090 & 0.3090 & 0.3090 & -0.8090 \\ 1 & 0.3090 & -0.8090 & -0.8090 & 0.3090 \end{bmatrix} - j\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9511 & 0.5878 & -0.5878 & -0.9511 \\ 0 & 0.5878 & -0.9511 & 0.9511 & -0.5878 \\ 0 & -0.5878 & 0.9511 & -0.9511 & 0.5878 \\ 0 & -0.9511 & -0.5878 & 0.5878 & 0.9511 \end{bmatrix} \tag{9}$$

Comparatively, the inverse discrete Fourier transform is described as

$$\mathbf{x}=\frac{1}{N}\mathbf{W}_N^*\mathbf{y} \tag{10}$$

where the elements in $\mathbf{W}_N^*$ are the complex conjugates of their corresponding elements in $\mathbf{W}_N$.

III. Fibonacci Fourier-like Transforms

This section presents the Fibonacci Fourier transforms and its fast algorithm. The new transforms are derived by rendering the relationship between the Fibonacci numbers and the classical discrete Fourier transforms. The fast algorithms are designed using the Kronecker product properties described in the previous section in order to reduce the computational complexity from $O(N^2)$ to $O(N\log_5 N)$.

A. Fibonacci Fourier Transforms

A basis of Fibonacci Fourier transform matrix $\mathbf{F}_5$ shown in (11) is derived from a 5-point discrete Fourier transform matrix represented in the form of the golden ratio using trigonometric identities [9] with (3).

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2\varphi}+j\frac{\varphi\gamma}{2} & -\frac{\varphi}{2}+j\frac{\gamma}{2} & -\frac{\varphi}{2}-j\frac{\gamma}{2} & \frac{1}{2\varphi}-j\frac{\varphi\gamma}{2} \\ 1 & -\frac{\varphi}{2}+j\frac{\gamma}{2} & \frac{1}{2\varphi}-j\frac{\varphi\gamma}{2} & \frac{1}{2\varphi}+j\frac{\varphi\gamma}{2} & -\frac{\varphi}{2}-j\frac{\gamma}{2} \\ 1 & -\frac{\varphi}{2}-j\frac{\gamma}{2} & \frac{1}{2\varphi}+j\frac{\varphi\gamma}{2} & \frac{1}{2\varphi}-j\frac{\varphi\gamma}{2} & -\frac{\varphi}{2}+j\frac{\gamma}{2} \\ 1 & \frac{1}{2\varphi}-j\frac{\varphi\gamma}{2} & -\frac{\varphi}{2}-j\frac{\gamma}{2} & -\frac{\varphi}{2}+j\frac{\gamma}{2} & \frac{1}{2\varphi}+j\frac{\varphi\gamma}{2} \end{bmatrix} \quad (11)$$

The generalized Fibonacci Fourier transform matrix $\mathbf{F}_5$ shown in (12) is derived from (9). The matrix representation in (12) combines a real part and an imaginary part matrix in which all the elements are Fibonacci numbers or the sum of Fibonacci numbers, where α and β are constants. The true value of (11) can be approximated by the generalized representation of the Fibonacci Fourier transform matrix $\mathbf{F}_5$ shown in (12) using $\alpha \approx 0.1$ and $\beta \approx 0.2$.

$$\mathbf{F}_5 = \alpha \begin{bmatrix} 11 & 11 & 11 & 11 & 11 \\ 11 & 3 & -8 & -8 & 3 \\ 11 & -8 & 3 & 3 & -8 \\ 11 & -8 & 3 & 3 & -8 \\ 11 & 3 & -8 & -8 & 3 \end{bmatrix} - j\beta \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 3 & -3 & -5 \\ 0 & 3 & -5 & 5 & -3 \\ 0 & -3 & 5 & -5 & 3 \\ 0 & -5 & -3 & 3 & 5 \end{bmatrix} \quad (12)$$

Fibonacci Fourier transforms of larger dimensions can be obtained using the Kronecker product properties described in the previous section on the transform matrix. A 5^k-point Fibonacci Fourier transform matrix can be expanded using k $\mathbf{F}_5$ matrices in (13), where $k \in \mathbb{N}$.

$$\underbrace{\mathbf{F}_5 \otimes \mathbf{F}_5 \otimes \cdots \otimes \mathbf{F}_5}_{k} = \mathbf{F}_{5^k} \quad (13)$$

The calculation can be processed recursively as

$$\mathbf{F}_{5^k} = \mathbf{F}_{5^{k-1}} \otimes \mathbf{F}_5 \quad (14)$$

Consider $k = 3$ for example, the 125-point Fibonacci Fourier transform matrix is derived from $\mathbf{F}_{5^3} = \mathbf{F}_{5^2} \otimes \mathbf{F}_5 = \mathbf{F}_5 \otimes \mathbf{F}_5 \otimes \mathbf{F}_5$.

B. Fast Fibonacci Fourier Transforms

To obtain the fast Fibonacci Fourier transforms, (6) is applied to the transform matrix in (14) recursively [8,14]. An N-point Fibonacci Fourier transform matrix can then be calculated using the Fibonacci Fourier transform matrix $\mathbf{F}_5$ and the identity matrices as

$$\mathbf{F}_{5^k} = \prod_{i=1}^{k} \left(\mathbf{I}_{5^{i-1}} \otimes \mathbf{F}_5 \otimes \mathbf{I}_{5^{k-i}} \right) \quad (15)$$

when $N = 5^k$. Applying a 125-point Fibonacci Fourier transform to a vector $\mathbf{x}$ of length 125, the output vector $\mathbf{y}$ can be derived using the fast algorithm as follows

$$\mathbf{y} = \mathbf{F}_{125}\mathbf{x} = \left(\mathbf{F}_5 \otimes \mathbf{I}_{25}\right)\left(\mathbf{I}_5 \otimes \mathbf{F}_5 \otimes \mathbf{I}_5\right)\left(\mathbf{I}_{25} \otimes \mathbf{F}_5\right)\mathbf{x} \quad (16)$$

Such representation improves the number of operations including additions and multiplications. The direct manipulation of a 5^k-point Fibonacci Fourier transform takes $2N^2$-N operations, while the fast algorithm merely takes $9N\log_5 N$ operations. Hence, the computational complexity is reduced from $O(N^2)$ to $O(N\log_5 N)$.

IV. FILTERING ALGORITHM

This section describes two proposed sliding window filtering techniques. The first is a sliding double window filter in which a selection is made between two window types, a transformed window and a spatial window. The transformed window is selected for filtering process, while the spatial window is nonoverlapped for pixel value substitution. The second technique is a fusion sliding window filter. This method is able to integrate different filters to reduce various noises, taking advantage of the filters individual strengths. For the two sliding window filters, the amount of noise reduction of an image is determined by a threshold.

A. Threshold Filtering

In the transform domain of a signal, the larger coefficients contain a majority of the information. In contrast, smaller coefficients contain less information of the signal. Therefore, if noise is added into the signal, threshold filtering allows the smaller coefficient values in the transform domain to be reduced to constants, e.g., zeros. Suppose $\mathbf{y}$ is a vector indicating the transform coefficients of length N, in which each element is denoted as y_i, where i = 0, 1, …, N-1. The values after thresholding $\hat{y}_i$ can be calculated as

$$\hat{y}_i = \begin{cases} y_i & y_i \geq \delta \\ v & y_i < \delta \end{cases} \quad (17)$$

where δ is the threshold and v is a constant.

B. Sliding Double Window Filtering

The sliding double window filter contains two window types, the transformed window and the spatial window. The former selects a block to be transformed for filtering processing. The latter decides a subblock within the former window for pixel replacement. Since the pixels in the spatial domain windows will be substituted after the filtering process, the method avoids overlapping between spatial domain windows.

Given a two dimensional signal. Fig. 1 shows the flow chart of the sliding double window filtering algorithm. The window w_M is the transformed window, while the window w_N is the spatial window. The next processing subimage block is determined by the nonoverlapping w_N window adjacent to the current block along the rows or columns. The procedure ends if every possible subimage block is processed.

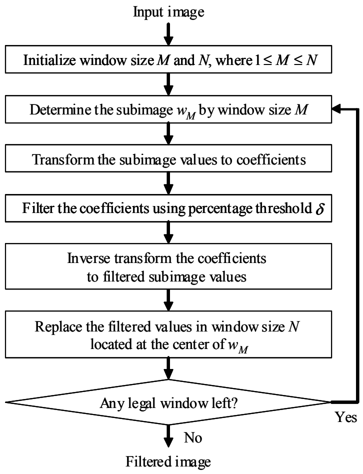


Figure 1. Sliding double window filtering algorithm.

C. Fusion Sliding Window Filtering

The primary concept of fusion sliding window filtering is to obtain the filtered signals by a weighted operation during the filtering process. For a given image, a certain operation is applied to the prefiltered images acquired from different filtering methods in order to attain the final filtered image. Fig. 2 shows the fusion sliding window filtering algorithm.

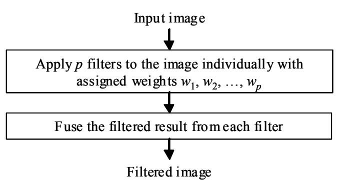


Figure 2. Fusion sliding window filtering algorithm.

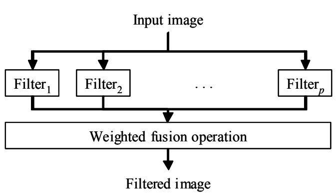


Figure 3. Fusion sliding window filtering schematic diagram.

Fig. 3 shows the fusion sliding window filtering schematic diagram. Different filters, such as threshold filters, Wiener filters, sliding double window filters, etc., can be used as individual filters for the given image. The results of pixel information from each filter $f_{s,t}^{i}$ are incorporated using weighted fusion technique as described

$$\hat{f}_{s,t} = \phi\left(w_i f_{s,t}^{i}\right) \tag{18}$$

where i = 1, 2, …, p, s and t indicate the locations of the pixel in the image is, and ϕ denotes the operation used, for example, average, summation, maximization, minimization, etc.

V. Computer Simulation

In this section, an application is demonstrated for image denoising using the new Fibonacci Fourier-like transforms. In addition to the two sliding window filtering algorithms, Wiener filtering [6] is used to make a comparison with the proposed method. The Wiener filter is a two dimensional adaptive filtering technique, often used in image denoising based on the statistics estimated from a local neighborhood of each pixel. The formula for the Wiener filter is described as

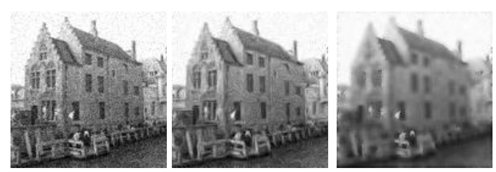
$$\hat{o}_{i,j} = \mu_{i,j} + \frac{\sigma_{i,j}^{\ 2} - s^2}{\sigma_{i,j}^{\ 2}}\left(o_{i,j} - \mu_{i,j}\right) \tag{19}$$

where i and j indicates the pixel location, $o_{i,j}$ is the observed value, $\hat{o}_{i,j}$ is the estimated value, $\mu_{i,j}$ is the local mean, $\sigma_{i,j}^{\ 2}$ is the local variance around each pixel, and s^2 is the noise variance, which indicates the average of all the local estimated variances.

By means of the presented methods, the experimental results reveal visually that the quality of the filtered images is better when the sliding window size equals to 5. Fig. 4 to Fig. 7 show examples of a selected number of images with additive Gaussian noise and corresponding filtered images with different filtering techniques.

(a) (b) (c)

Figure 4. (a) The noisy daisy image added Gaussian noise with zero mean and 0.003 variance. (b) The filtered image by exploiting the sliding double window filtering algorithm with M = 5, N = 1, and threshold value 86%. (c) The filtered image by implementing Wiener filtering with window size 5.

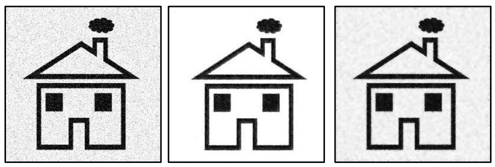


(a) (b) (c)

Figure 5. (a) The noisy building image [17] added Gaussian noise with zero mean and 0.0025 variance. (b) The filtered image by exploiting the sliding double window filtering algorithm with M = 5, N = 1, and threshold value 81%. (c) The filtered image by implementing Wiener filtering with window size 5.

(a) (b) (c)

Figure 6. (a) The noisy chess image added Gaussian noise with zero mean and 0.0008 variance. (b) The filtered image by exploiting the fusion sliding window filtering algorithm with 2 filters, i.e., sliding double window filtering with M = 5, N = 1, and threshold value 80%, and Wiener filtering with window size 5, giving average operation. (c) The filtered image by implementing Wiener filtering with window size 5.



(a) (b) (c)

Figure 7. (a) The noisy home image added Gaussian noise with zero mean and 0.02 variance. (b) The filtered image by exploiting the fusion sliding window filtering algorithm with 2 filters, i.e., sliding double window filtering with M = 5, N = 1, and threshold value 87%, and Wiener filtering with window size 5, giving average operation. (c) The filtered image by implementing Wiener filtering with window size 5.

Referring to Fig. 4 to Fig. 7, one can obtain clearer images with the proposed Fibonacci Fourier-like transforms and the filtering techniques. By altering the parameters of sliding window filtering algorithms, higher quality images is achieved visually, i.e., Fig. 4(b), 5(b), 6(b) and 7(b), can be acquired with the proposed method while the images by Wiener filtering in Fig. 4(c), 5(c), 6(c) and 7(c) appear blurred.

VI. Conclusion

In this paper, the Fibonacci Fourier transform and its fast algorithm are presented. A demonstration on image denoising is shown using the proposed algorithms when compared to Wiener filtering. For visualization, adopting the proposed filtering techniques can obtain better quality images. Not only are the noisy areas smoothed, but the edges are maintained. In future work, investigation on algorithms can help improve the quality of filtered images. The fusion techniques can also be investigated to improve the calculation of the weights in (18). Furthermore, this method can be exploited in the fields of face recognition, character recognition, generation of features for pattern recognition, etc.

Acknowledgment

A special thank goes to the Department of Electrical and Computer Engineering, The University of Texas at San Antonio for partially funding this research.

References

[1] H. A. Barger and K. R. Rao, "A comparative study of phonemic recognition by discrete orthogonal transforms," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 553–556, 1978.

[2] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling Jr., D. E. Nelson, C. M. Rader and P. D. Welch, "What is the fast Fourier transform?" Proceedings of the IEEE, vol. 55, no. 10, pp. 1664–1674, 1967.

[3] G. L. Demuth, "Algorithms for defining mixed radix FFT flow graphs," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 9, pp. 1349–1358, 1989.

[4] L. Fibonacci and L. E. Sigler, Fibonacci's Liber Abaci, Springer, 2002 [Liber Abaci, 1202].

[5] J. A. Glassman, "A generalization of the fast Fourier transform," IEEE Transactions on Computers, vol. C–19, no. 2, pp. 105–116, 1970.

[6] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed., Prentice Hall, NJ, 2008.

[7] J. Granata, M. Conner and R. Tolimieri, "Recursive fast algorithm and the role of the tensor product," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 40, no. 12, pp. 2921–2930, 1992.

[8] N. C. Hu and O. K. Ersoy, "Fast computation of real discrete Fourier transform for any number of data points," IEEE Transactions on Circuits and Systems, vol. 38, no. 11, pp. 1280–1292, 1991.

[9] T. Koshy, Fibonacci and Lucas Numbers with Applications, Wiley-Interscience, NY, 2001.

[10] A. J. Laub, "Kronecker products," Matrix Analysis for Scientists and Engineers, Chapter 13, Society for Industrial and Applied Mathematics, PA, 2004, pp. 139–147.

[11] D. R. Mack, "The magical Fibonacci number," IEEE Potentials, vol. 9, no. 3, pp. 34–35, 1990.

[12] A.V. Oppenheim and R. W. Schafer, Discrete-time signal processing, Prentice Hall, New Jersey, 1989.

[13] S. Oraintara, Y. J. Chen and T. Q. Nguyen, "Integer fast Fourier transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 50, no. 3, pp. 607–618, 2002.

[14] X. Qingbin, Z. Qing and S. Shenghe, "Iterative structure of Winograd FFT algorithm," IEEE Conference on Precision Electromagnetic Measurements Digest, pp. 57–58, 1996.

[15] K. R. Rao and N. Ahmed, "Orthogonal transforms for digital signal processing," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 136–140, 1976.

[16] S. Sridharan, E. Dawson and B. M. Goldburg, "Speech encryption using discrete orthogonal transforms," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 1647–1650, 1990.

[17] Image Database, Breaking Our Watermarking System, 2nd ed., http://bows2.gipsa-lab.inpg.fr/.

SMC 2008