International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.2 (2016), pp.369-378
http://dx.doi.org/10.14257/ijsip.2016.9.2.31

The Research and Improvement on the Fast Algorithm of Fourier Transform Image

Zhang Yong

Suzhou Vocational and Technical College
zysy1120@126.com

Abstract

The quantity of digital image is huge. Undoubtedly, it will be a huge project to complete the image processing in a limited time, and it is unrealistic to directly deal with the real-time image signal with the Discrete Fourier Transform (DFT) algorithm. There is no fundamentally change until Cooley and Tukey proposed a Fast Fourier Transform Algorithm (FFT) and it will greatly promote the application of Discrete Fourier Transform in all aspects. The following will show its superiority from the detailed study of the FFT algorithm and accordingly make the further improvement to the algorithm to improve the use efficiency of the FFT algorithm.

Keywords: *DFT; FFT; algorithm; algorithm improvement*

1. Fast Fourier Transform Algorithm

At present, the common algorithm is FFT, which is based on the radix 2 algorithm proposed by Cooley and Tukey. And then, there are a lot of people committed to further reduce the computation of DFT and proposed some improved algorithms. One of the most famous is the WFTA algorithm presented by Dr. Shmuel Winograd of IBM in 1975 (Winograd Fourier Transform Algorithm, referred to as WFTA algorithm).

This section will first mainly discusses the basic idea of Fast Fourier Transform (FFT), introducing the radix 2FFT and WFTA algorithms, and briefly introduce the radix 4 and prime factor algorithms based on radix 2.

1.1. Basic idea of FFT

In DFT, the N point of the DFT complex multiplication frequency is equal to N^2. Obviously, the N DFT is decomposed into several shorter DFT, which can make greatly reduce the times of multiplication. In addition, the twiddle factor has obvious periodicity and symmetry, and the periodicity shows as follows:

$$W_N^{m+lN} = e^{-j2\pi(m+lN)} = e^{-j\frac{2\pi}{N}m} = W_N^m \tag{1.1}$$

The performance of symmetry is

$$W_N^{-m} = W_N^{N-m} \quad \text{or} \quad [W_N^{N-m}]^* = W_N^m \tag{1.2}$$

$$W_N^{m+\frac{N}{2}} = -W_N^m \tag{1.3}$$

FFT algorithm is to keep the long sequence of DFT being decomposed into several short sequence of DFT, and use the periodicity and symmetry of W_N^{kn} to reduce the computation sequence of DFT.

ISSN: 2005-4254 IJSIP

1.2. Several Classic Algorithms of FFT

(1) Extraction radix 2FFT algorithm based on time domain -- DIT-FFT

This algorithm is to extract the order of input sequence in time domain according to the even and odd numbers. For an arbitrary $N=2^m$ point of a long sequence of DFT operations, it can be decomposed into M times and finally broken down into 2 point DFT operation, thereby reducing the amount of computation.

Suppose the length of $x(n)$ sequence is N, and meets the condition $N=2^m$ (a natural number).

The first decomposition: $x(n)$ is decomposed into two N/2 subsequences according to the parity of n

$$x_1(r)=x(2r), r=0,1,...,\frac{N}{2}-1 \quad \text{or} \quad x_2(r)=x(2r+1), r=0,1,...,\frac{N}{2}-1$$

And then the DFT of $x(n)$ is $X(k)=\sum_{r=0}^{N/2-1} x_1(r)W_N^{2kr}+W_N^k\sum_{r=0}^{N/2-1} x_2(r)W_N^{2kr}$ (1.4)

$$W_N^{2kr}=e^{-j\frac{2\pi}{N}2kr}=W_{N/2}^{kr}$$

So $X(k)=\sum_{r=0}^{N/2-1} x_1(r)W_N^{2kr}+W_N^k\sum_{r=0}^{N/2-1} x_2(r)W_N^{2kr}=X_1(k)+W_N^k X_2(k)$ k=0,1,…,N-1

(1.5)

The above $X_1(k)$ and $X_2(k)$ respectively represent the N/2 DFT of $X_1(k)$ and $X_2(k)$. Both $X_1(k)$ and $X_2(k)$ take N/2 as the cycle with the condition of $W_N^{m+\frac{N}{2}}=-W_N^m$, so they can also be expressed as

$$X(k)=X_1(k)+W_N^k X_2(k) k=0,1,...,N/2-1 \quad (1.6)$$

$$X\ (k+N/4)=X_1(k)-W_N^k X_2(k), k=0,1,...,N/2-1, \quad (1.7)$$

In this way, the DFT of N point is decomposed into two N/2 point DFT and the operations as (1.6) and (1.7).

The second decomposition: same as the first decomposition, $x_1(r)$ is decomposed into two subsequences with the length of N/4 according to its parity, $x_3(l)=x_1(2l), l=0,1,...,\frac{N}{4}-1$ and $x_4(l)=x_1(2l), l=0,1,...,\frac{N}{4}-1$, then it follows the first results,

$$X_1(k)=X_3(k)+W_{N/2}^k X_4(k), k=0,1,...,N/4-1 \quad (1.8)$$

$$X_1(k+N/4)=X_3(k)-W_{N/2}^k X_4(k), k=0,1,...,N/4-1 \quad (1.9)$$

It can be calculated with the same method.

$$X_2(k)=X_5(k)+W_{N/2}^k X_6(k), k=0,1,...,N/4-1 \quad (1.10)$$

$$X_2(k+N/4) = X_5(k) - W_{N/2}^{k} X_6(k), k = 0,1,...,N/4-1 \qquad (1.11)$$

In this way, it decomposes the N/2 point DFT into two N/4 ones after the second decomposition. And so forth, it finally divides the N point DFT into N/2 two point DFT after M—1 decomposition.

The following will take a DFT with the length of 8 as an example to illustrate the algorithm thought. When N=8, the calculation can be decomposed into three steps according to the above analysis. Figure 1-1, Figure 1-2 and figure 1-3 shows the three concrete steps, while the figure 1-4 displays the entire operation process.

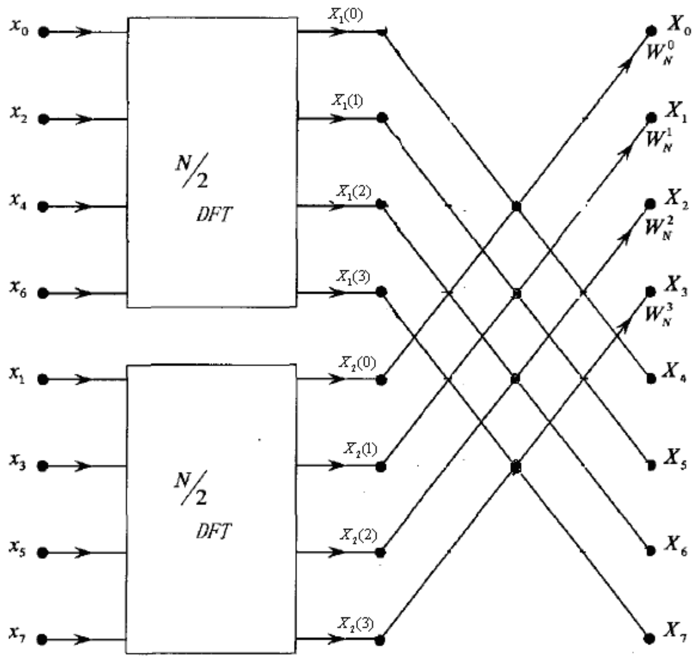

Figure (1-1). The First Time-Domain Extraction Decomposition Graph of 8 Point DFT

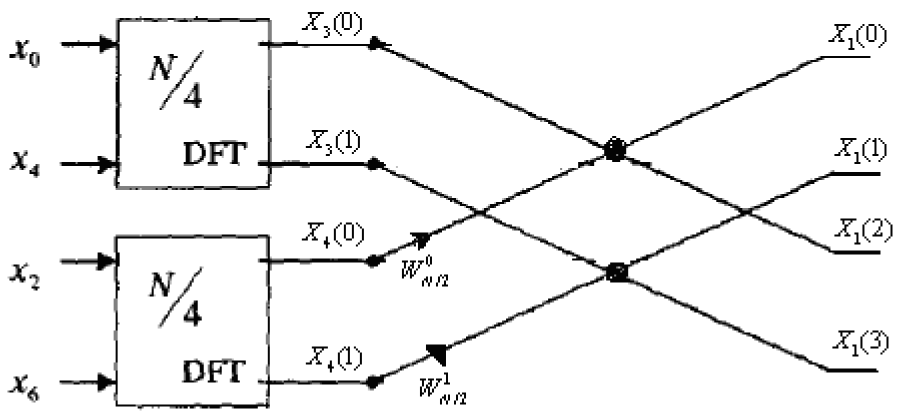

Figure (1-2). The Second Time-Domain Extraction Decomposition Graph of 8 Point DFT

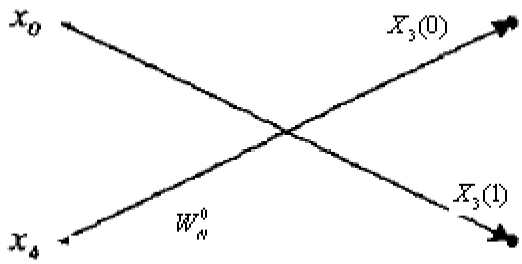

Figure (1-3). The Third Time-Domain Extraction Decomposition Graph of 8 Point DFT

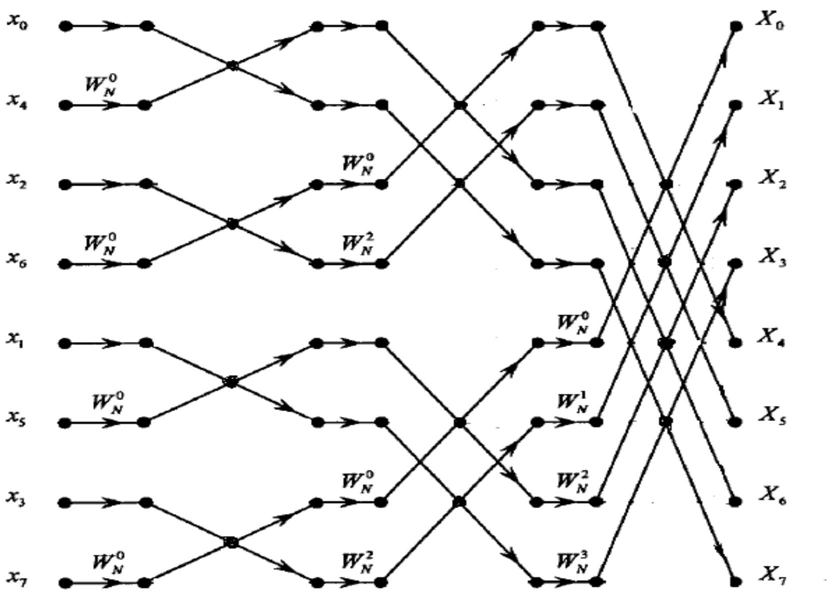

Figure (1-4). The Operation Flow Chart of 8 Point DFT

Corresponding with the DIT-FFT algorithm, the radix 2FFT algorithm (DIF-FFT) based on frequency domain is to stepwise decompose the frequency output X (k) into 2 point DFT operation according to the even or odd nature of K, whose principle is relatively even with the DIT-FFT algorithm, and the computation is also equal to the DIT-FFT algorithm, so here repeat no more.

(2) The radix 4FFT algorithm

The basic principle of radix 4FFT is about the same to that of 2FFT. First, it turns the N-length DFT into four DFTs with the length of N/4, and the required number of multiplications and additions respectively is 3N and 3N/4. Each new DFT can be decomposed into four DFTs with the length of N/16. It doesn't stop until it turns into a simple butterfly pattern.

In fact, the number of access to memory for radix 4 can also be reduced by half, which is also conducive to the improvement of operation speed. But the enumerative value of radix 4 should be half of the radix 2FFT algorithm, which limits the application of radix 4FFT, and the algorithm of radix 4FFT butterfly graph is much more complicated than 2FFT. Theoretically, the higher radix, the less total amount of computation, so radix 8 and 16 algorithm will make the number of operations further reduce, but the reductions will be less significant, and the butterfly diagram of radix 8 and 16 algorithms will be more complex. It is very inconvenient to realize through the hardware and software, so there is little research and utilization about it all the time.

(3) Prime factor algorithm (PFA)

The prime factors of both radix 2 and 4 algorithms have a very big limitation on the DFT length of N and they must be 2 or 4 times N. But it is often impossible in the practical application. For this reason, the later DFT algorithm developed a very good solution to this problem. While the composite number N can be decomposed into several coprime factor according to the Good mapping, its FFT transform can avoid the influence of the rotation factor. PFA algorithm is the use of Good mapping, turning the one-dimensional DFT' with the length of $N = N_1 * N_2$ into a two-dimensional DFT' with the size of $N_1 \times N_2$, and then calculate it along each dimension with the most efficient algorithm.

(4) WFTA algorithm

WFTA is a novel algorithm constructed on the basis of subscript mapping and number theory. Its main idea is to decompose an N ($N = N_1 \times N_2$) length Discrete Fourier Transform into two coprime small N factor DFT. It first calculates the length of N_2 transform results, and then uses the results to calculate the length of N_1. In practical application, if N is larger, N can also be divided into the product of a plurality of Coprime small Ns. "Small N" factor DF I' refers to 2-9 and 16 point DFT.

2. Comparison of Several Algorithms

In order to have a more intuitive comparison of various algorithms, the following table summarizes the various algorithms introduced in front of the. Performance and application scope

Table One: Comparison of Several Algorithms

Algorithm	Range of application	Operand	Stock
Radix 2FFT Radix 4FFT	N is an integer power of 2	The highest number of multiplication. The least number of addition. The less the greater number of N in theory.	The least
PFA	N is not an integer power of 2	The number of multiplication and addition times is almost between radix 4FFT and WFTA, depending	The less

		on the group factors of N.	
WFTA	Especially suitable for the calculation of real data.	The least number of multiplication and the largest number of addition.	The least

In addition to the summary of the above table, we list the following two interpretation about the table:

(1) If take the realization of ultra large scale integrated circuit as the target, the radix 2FFT and 4FFT will be a better choice. It has the advantages of simple structure and convenient storage mode, using for the treatment of vector machine;

(2) For the arithmetic unit with slow multiplication and fast addition speed, employing the WFTA algorithm can obtain obvious effect, and the WFTA is especially suitable for computing real input DFT.

Each algorithm has its own different characteristics. We can choose the applicable algorithms according to the specific circumstances or even combined with other methods in the practical application, and make improvements to them.

3. Improvement of Algorithms

The algorithm of the previously mentioned radix 2FFT is simple and the WFTA has less computation, so people will naturally ask if it is possible to finish the partial arithmetic through the simple radix 2FFT while the other operations by using the WETA in order to get a better FFT algorithm. The answer is yes. The following will be combined with the radix 2FFT and WFTA algorithms improving FFT, which will make the new FFT computation small and simple.

3.1. The Principle of the Improved Algorithm

First, let's recall the method of radix 2-FFT, it gradually decomposes a longer DFT into shorter DFT, and the length of the DFT will be the half of the former layer in each iteration. The result of iteration is to decompose the DFT of N length into N/2 DFT, which is the method of radix 2FFT algorithm with the decomposition process of $N\log_2 N$ step. However, our goal is to get the DFT with the length of N in actual calculation. Therefore, we must start from the bottom of the method in the process of calculation. We first calculate the N/2 DFT computing two points in the final layer; then combine the calculated results in accordance with the reverse process of method to get the 4 length DFT, which will be processed into the8 length DFT in the same way. By analogy, when the algorithm comes to the last step, the two N/2 length DFT will become the DET of N length by multiplication and addition and this is what we want to get.

The improved algorithm is to insert the WFTA into the process of radix 2-FFT algorithm, that is to say, the WFTA algorithm will take the place of a partial radix 2-FFT. The basic idea is as follows:

The decomposition process of radix 2-FFT algorithm consists of $\log_2 N$ steps. It has N/16 DFT with the length of 16 while decomposing into $\log_2 N-4$ steps in radix 2-FFT algorithm. Each group DETs iterated and decomposed from the radix 2-FFT is independent, so we can introduce the WFTA algorithm to compute these 16 length DFT. In view of the fact that the calculation process and method analysis thought is just reciprocal, we will first employ the 16 point WFTA algorithm to calculate, and then combine the results according to the radix 2FFT to get the N length DFT.

The reason for the improved algorithm adopting the combination mode is if the N's length is larger, the WFTA algorithm will become quite complex, so we here only combine the 16 bit WFTA algorithm with the radix-2FFT. In this way, it can not only keep the simple algorithm structure, but also have access to the advantage of WFTA in multiplication.

The following takes N=32 as an example to illustrate that the improved algorithm. The 32 data are divided into odd and even groups according to their subscript. These two groups respectively precede the WFTA calculation of 16 points, and the results are sent back to the site for radix 2FET. The calculation process as shown in Figure 1-5, the right part is actually the butterfly diagram of the last layer of radix 2FFT, while the left side is the use of WFTA algorithm, whose specific process as shown in Figure 3-5.

To sum up, the whole algorithm can be divided into the following steps:

The first step, input the data into an array DATA, and determine whether the number of data N is an integer power of 2. If not, it needs zero-fill to make the calculation length an integer power of 2.

The second step, extract the data according to the frequency of N/16 to make up of N/16 arrays with 16 numbers each.

The third steps, develop 16 point AFTA calculation for these arrays, and put the results into the DATA.

The fourth step, precede radix 2FFT calculation for the DATA array.

The fifth step, output the results

Figure 1-6 The flow chart of the whole algorithm

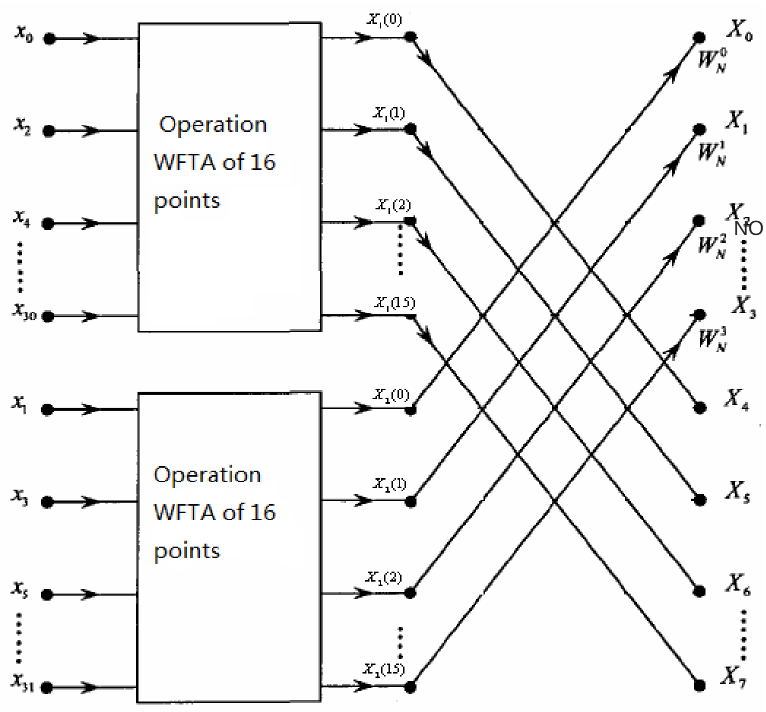

Figure (1-5). The Butterfly Diagram of the Improved Algorithm N=32

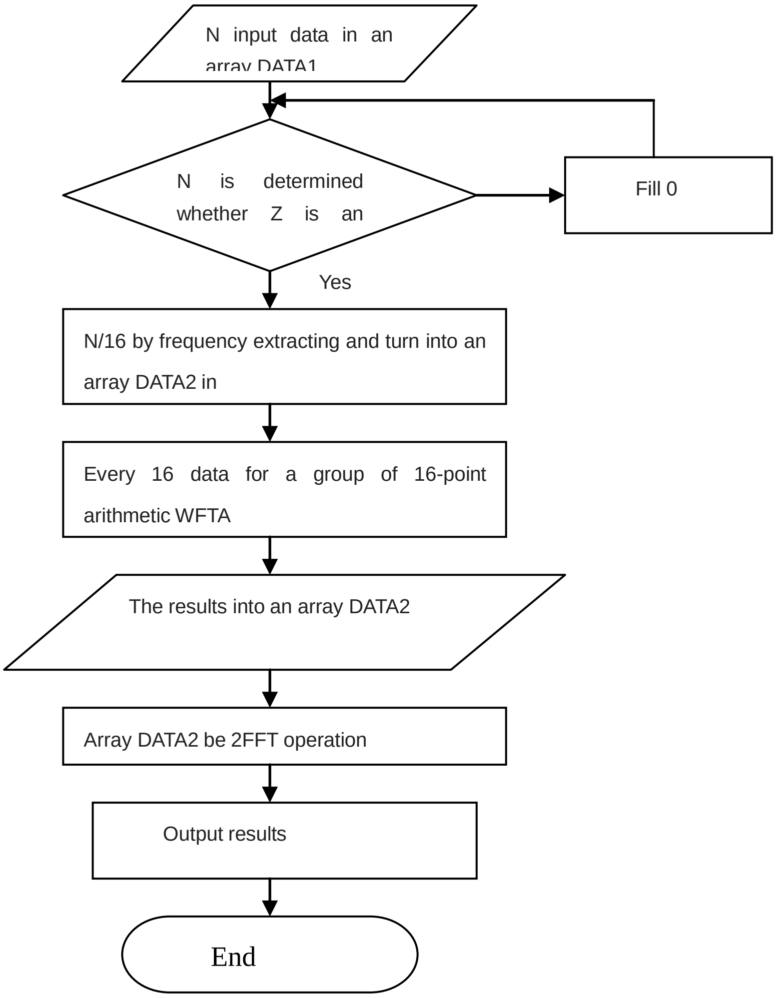

Figure 1-6. The Flow Chart of the Improved Algorithm

The improved algorithm is only applicable to the case of $N = 2^m$. The algorithm is a combination of radix 2FFT and WFTA, so the computation still can meet the radix 2FFT computation, while the times of multiplication in the last two steps of iterative process decreases. But when N is very large, such as N=600, if use the above algorithm, it needs zero-fill to extend to 1024. If discard the latter 424 data after the calculations, thus it brings a lot of redundant operations, only to increase the computation quantity. On the other hand, it also produces an error in the spectrum to a certain extent. The more zero-fills, the greater relative errors.

3.2. The Remained Problems in the Algorithm

It can be found that the new algorithm reduces the computing amount of DFT in a certain extent, from the above analysis, but this improved algorithm can not appropriately solve the problem that N is not the whole power of 2. In order to solve this problem, we may introduce prime factor algorithm (PFA) in the arithmetic. For a DFT with the length of $N = 2^m \times N_1$, the specific method is as follows: first use PFA algorithm to decompose the DFT($N = 2^m \times N_1$), and extract according to the interval of N_1. It will get 2^m groups of data and calculate, and then extract every other 2^m space, and calculate again, the calculation results will be returned to the original site. Second, use the improved algorithm to calculate. If N_1 is the product of several coprime factors, it can continue the decomposition computation according to PFA.

Combining with radix 2FFT, PFA and WFTA, it not only improve the FFT operational efficiency but also increase the application scope of the algorithm. However, this method will make the algorithm structure complex and it is hard to achieve.

References

[1] C. Shu-hai and F. Luxiang, "Practical Digital Image Processing [M]", Beijing: Science Press, **(2005)**.
[2] Y. Fan, "Digital Image Processing and Analysis [M]", Second Edition. Beijing: Beihang University Press, **(2010)**.
[3] X. Xiao-rong, "Fourier Transform and Its Application [J]", News World Paper Edition., First, **(2009)**, pp. 138-140.
[4] (American) R. N. Bracewell (Ronald N. Bracewell), "Fourier Transform and Its Application [M]", Y. Jianye and Y. Jianguo translated, "Xi'an: Xi'an Jiao Tong University Press", **(2005)**.
[5] E. M. Stein and R. Shakarchi, "Fourier Analysis an Inteoduction[M]", Beijing: The world book publishing company in Beijing, **(2013)**.
[6] O.Morice, D.Nassie, G.Gallice, etal, Laser Pulse Propagation Calculations using the Miro software, Inertial Fusion Sciences and applications99,(1999)
[7] An Introduction to MATLAB. SIMULINK. And the MATLAB Application Toolboxes. The Math Works. Inc. April (1993)
[8] Evapart-Enander. AndersSjoberg: The MATLAB5 Handbook, China Machine Press, Beijing, (2000)
[9] Gonzalez R C, Woods R E. Digital Image. Beijing: Publishing House of Electronics Industry, (2002)
[10] Leng Jian-Hua. Fourier transform [M]. Beijing: Tsinghua university press, (2004)
[11] Cheng Pei-qing. Digital signal processing (second edition) [M]. Beijing: Tsinghua university press, (2006)
[12] Yang Li-juan, Zhang bai-hua, Xuzhen. Fast Fourier transform FFT and its application [J]. Journal of electrical engineering. Volume on December 31, 2004 8, 1003-501 - x (2004) Sup - 0001-03, 1 ~ 2
[13] Xu Xiao-rong. Fourier transform and its application [J]. Journal of news paper version of heaven and earth. (2009)
[14] Zhao Jun-fang. The application of Fourier transform in image processing [J]. Journal of electronic measurement technology abroad. (2004)
[15] Ge Shi-ming. Study on the method of digital image restoration [D]. Dissertation. University of science and technology of China, in April, 2008
[16] Li Su-li. The research and application of digital image restoration technology [D]. Master thesis, xi 'an university of science and technology building, (2010).
[17] Yu Cheng-bo. Digital image processing and MATLAB [M] .Chongqing University press, (2003)

Author

Zhang Yong, he received his Master of Software Engineering (MSE)(2009) from University of Electronic Science and Technology of China. Now he is Associate Professor of Computer, Suzhou Vocational and Technical College. His current research interests include Software engineering and image processing technology.