Vol 17 No 9, September 2008
1674-1056/2008/17(09)/3220-07

Chinese Physics B

© 2008 Chin. Phys. Soc.
and IOP Publishing Ltd

A quantum search algorithm of two entangled registers to realize quantum discrete Fourier transform of signal processing*

Pang Chao-Yang(庞朝阳)[a)b)†] and Hu Ben-Qiong(胡本琼)[c)]

[a)] *Key Software Laboratory, Sichuan Normal University, Chengdu* 610066, *China*
[b)] *College of Mathematics and Software Science, Sichuan Normal University, Chengdu* 610066, *China*
[c)] *College of Information Management, Chengdu University of Technology, Chengdu* 610059, *China*

(Received 27 June 2007; revised manuscript received 27 February 2008)

The discrete Fourier transform (DFT) is the base of modern signal processing. 1-dimensional fast Fourier transform (1D FFT) and 2D FFT have time complexity $O(N\log N)$ and $O(N^2\log N)$ respectively. Since 1965, there has been no more essential breakthrough for the design of fast DFT algorithm. DFT has two properties. One property is that DFT is energy conservation transform. The other property is that many DFT coefficients are close to zero. The basic idea of this paper is that the generalized Grover's iteration can perform the computation of DFT which acts on the entangled states to search the big DFT coefficients until these big coefficients contain nearly all energy. One-dimensional quantum DFT (1D QDFT) and two-dimensional quantum DFT (2D QDFT) are presented in this paper. The quantum algorithm for convolution estimation is also presented in this paper. Compared with FFT, 1D and 2D QDFT have time complexity $O(\sqrt{N})$ and $O(N)$ respectively. QDFT and quantum convolution demonstrate that quantum computation to process classical signal is possible.

Keywords: Grover's algorithm, entangled state, DFT, QDFT
PACC: 0367, 8000, 4230V, 4230S

1. Introduction

1.1. Introduction of discrete Fourier transform (DFT)

DFT is the base of modern signal and information processing. Without DFT, there would be no modern signal and information processing.[1]

Let

$$\boldsymbol{W}_N=\frac{1}{\sqrt{N}}\begin{bmatrix}\omega^0 & \omega^0 & \omega^0 & \dots & \omega^0\\ \omega^0 & \omega^1 & \omega^2 & \dots & \omega^{N-1}\\ \omega^0 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)}\end{bmatrix}=\begin{pmatrix}\boldsymbol{W}_0\\ \boldsymbol{W}_1\\ \vdots\\ \boldsymbol{W}_{N-1}\end{pmatrix}, \tag{1}$$

where $\omega=\mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}}$ and $\boldsymbol{W}_0$, $\boldsymbol{W}_1,\dots,\boldsymbol{W}_{N-1}$ is the line vector of the matrix. The matrix $\boldsymbol{W}_N$ defined in Eq.(1) is called the Fourier transform matrix.[1]

The one-dimensional DFT (1D DFT)[1] is defined as

$$\boldsymbol{c}=\boldsymbol{W}_N\boldsymbol{x}=\begin{pmatrix}\boldsymbol{W}_0\bullet\boldsymbol{x}\\ \boldsymbol{W}_1\bullet\boldsymbol{x}\\ \vdots\\ \boldsymbol{W}_{N-1}\bullet\boldsymbol{x}\end{pmatrix}, \tag{2}$$

*Project supported by Sichuan Normal University, China (Grant No 06lk002).
†E-mail: cyp_900@hotmail.com, cypang@sicnu.edu.cn

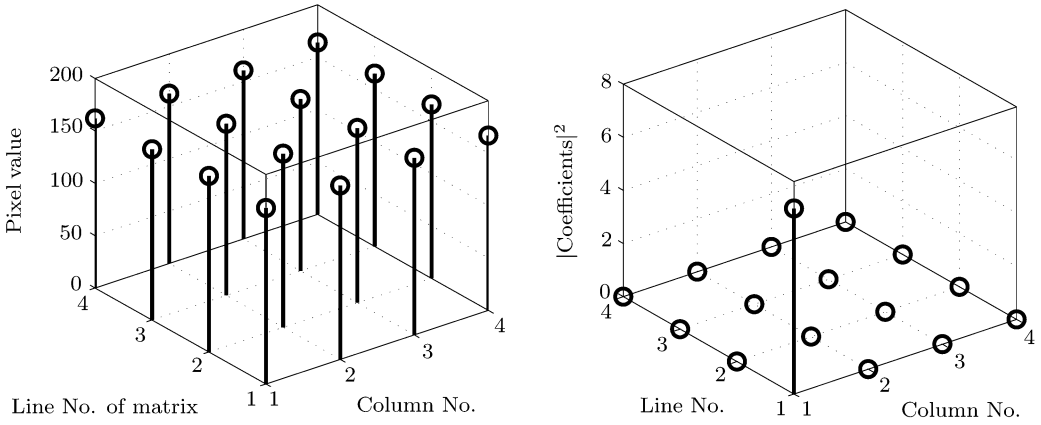
where vector $\boldsymbol{x} = (x_0, x_1, x_2, \ldots, x_{N-1})^{\mathrm{T}}$, $\boldsymbol{c} = (c_0, c_1, c_2, \ldots, c_{N-1})^{\mathrm{T}}$ (T denotes the transpose of vector), and '$\bullet$' denotes the inner product between two vectors. Each component of vector $\boldsymbol{c}$ is called Fourier coefficient.

For any input matrix $F = [f_{ij}]_{N\times N}$, 2D DFT[1] is defined as

$$C = W_N F W_N, \tag{3}$$

where C denotes the matrix of DFT coefficients.

DFT has two important properties, that will be applied to design quantum algorithm in this paper. One property is that DFT is energy conservation transform (i.e., $||\boldsymbol{x}||^2 = ||\boldsymbol{c}||^2$, or $\sum(x_i)^2 = \sum|c_i|^2$). The other property is that, typical data sequence, such as digital image, has high redundance, many of its DFT coefficients have values close to zero, and these coefficients can be discarded without seriously affecting the estimated value of $||\boldsymbol{x}||^2$ (i.e., without seriously affecting the quality of the restored image). Therefore, inverse DFT acting on the few big retained coefficients can restore the original data approximatively. Figure 1 shows the properties of DFT.



(a) the distrbution of original data

(b) the distrbution of DFT coefficients

Fig.1. Illustration of the two properties of DFT that were applied to design quantum algorithm in this paper. (a) The original data of the top-left image block with size 4×4 of image Lena.[2] (b) The 2D DFT coefficients of the image block. DFT has two properties. One property is that DFT is energy conservation transform. The other property is that, only few DFT coefficients is not close to zero, and the inverse DFT acting on these coefficients can restore the original data approximately. The two properties are applied to design quantum DFT algorithm in this paper.

Fast Fourier transforms (FFT) were presented at 1965,[1] and 1D and 2D DFT have time complexity $O(N\log N)$ and $O(N^2\log N)$ respectively. Fast DFT plays the key role of modern signal processing. However, since 1965, there is no more essential breakthrough for the design of fast DFT algorithm. The principle of electronic computation becomes the bottleneck of designing fast DFT algorithm.

1.2. Introduction of quantum computation

1.2.1. Introduction of quantum Fourier transform (QFT)

QFT[3,4] on an orthonormal basis $|0\rangle$, $|1\rangle, \ldots,$ $|N-1\rangle$ is defined to be a linear operator $\mathrm{QFT}|j\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} \mathrm{e}^{2\pi \mathrm{i} jk/N}|k\rangle$. QFT is the key to the famous Shor order-finding and factoring algorithm.[3,5] QFT has time complexity $O(\log^2 N)$. The output of QFT is the superposition of states, not the DFT coefficients that are actual digital numbers. Thus, QFT is not suitable to classical signal and information processing.[6,7]

1.2.2. Introduction of Grover's Algorithm

Grover's algorithm[8] solves the problem of searching for an element with a unique index i_0 in a list of N unsorted elements, similar to searching a database like a telephone directory when we know the number but not the person's name.[4] Grover's algorithm has time complexity $O(\sqrt{N})$.[4] Long[9] proposed an improved

Grover's algorithm, that has the probability of success 100% even for the case that the number of elements is very small. Boyer, Brassard, Hϕyer, and Tap[10] presented an improved Grover's algorithm (named BBHT algorithm) for the case that the number of solutions is unknown. The BBHT algorithm is a very smart algorithm because it saves many quantum circuits.

1.2.3. Introduction of quantum loading scheme $U_{\rm L}$[11]

Grover's algorithm has a function which finds the index i_0 of a special database record record_{i_0} from the index superposition of state $\frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle\right)$. And the record_{i_0} is the genuine answer wanted by us. However, the corresponding record_{i_0} cannot be measured out unless the 1–1 mapping relationship between index i and the corresponding record_i is bound in the entangled state $\frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle\,|\text{record}_i\rangle\right)$. That is, we need an unitary operation $U_{\rm L}$ to load all records that are stored in a classical database into quantum state. The function of unitary operation $U_{\rm L}$ can be described as

$$\frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle\,|0\rangle\right)|\text{ancilla}\rangle \xrightarrow{U_{\rm L}} \frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle\,|\text{record}_i\rangle\right)|\text{ancilla}\rangle, \tag{4}$$

where ancillary state $|\text{ancilla}\rangle$ is known.

Pang[11] presented a method to design unitary operation $U_{\rm L}$, which has time complexity $O(\log N)$ (unit time: phase transform and flipping the qubits of registers). Operator $U_{\rm L}$ is so fast that its running time can be ignored when analysing the time complexity of an algorithm.

1.2.4. Introduction of generalized Grover's algorithm[2,12−14]

Grover's algorithm can find a database recorded according to the given index. However, database search is complex in general. E.g., police often hopes to find a mug shot from the big database in which many sample photos are stored. And the similarity degree between every sample photo and the mug shot should be calculated and the closest sample photo should be found out. Clearly, Grover's algorithm is invalid for this kind of search case. Pang *et al* present a generalized Grover's algorithm,[2,12−14] which is derived from the research of quantum image compression.[14] The key of generalized Grover's algorithm is to design the following general Grover iteration $G_{\rm general}$:

$$G_{\rm general} = (2|\xi\rangle\langle\xi| - I)(U_{\rm L})^{\dagger}(O_{\rm c})^{\dagger}O_{\rm f}O_{\rm c}U_{\rm L}, \tag{5}$$

where $O_{\rm c}$ denotes computation oracle such as image matching, f denotes the judge function (i.e., if the output of $O_{\rm c}$ satisfies some conditions, let $f = 1$, else $f = 0$), $O_{\rm f}$ is the oracle of the judge function, and $|\xi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle$.

Let unitary operation $G_{\rm general}$ act on initial state $\frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle_{\text{registers1}}\right)|0\rangle_{\text{registers2}}$ $O(\sqrt{N})$ times and the optimal solution will be found. It should be noted that the first register and the second register are entangled during the computation processing.

2. 1-dimensional quantum DFT (1D QDFT)

2.1. Construct the following data structure (DS) and unitary operations

DS1 Save DFT matrix W defined in Eq.(1) in classical memory as a database. And each line vector $\boldsymbol{W}_i$ of the matrix is regarded as a record, which has index i.

DS2 Construct six registers that have data format

$$|\alpha\rangle_{\text{register1}}|\beta\rangle_{\text{register2}}|\boldsymbol{x}\rangle_{\text{register3}}|i\rangle_{\text{register4}}|\boldsymbol{W}_i\rangle_{\text{register5}}|(\boldsymbol{W}_i \bullet \boldsymbol{x})^2\rangle_{\text{register6}}.$$

That is, 1st, 2nd, 3rd, 4th, 5th, and 6th registers are used to save input parameter α, input parameter β, input vector $\boldsymbol{x}$, index i, line vector $\boldsymbol{W}_i$, and squared inner product respectively.

DS3 Design oracle O_{inner} to compute the squared inner product between vector $\boldsymbol{W}_i$ and $\boldsymbol{x}$, i.e.

$$|\alpha\rangle|\beta\rangle|\boldsymbol{x}\rangle|i\rangle|\boldsymbol{W}_i\rangle|0\rangle \xrightarrow{O_{\text{inner}}} |\alpha\rangle|\beta\rangle|\boldsymbol{x}\rangle|i\rangle|\boldsymbol{W}_i\rangle|(\boldsymbol{W}_i \bullet \boldsymbol{x})^2\rangle.$$

We can design a very simple parallel circuit to calculate squared inner product, and has time complexity $2t_{\text{m}} + [\log_2 N]t_{\text{a}}$, where t_{m} and t_{a} denote the unit running time of multiplication and addition respectively. Because addition is more fast than multiplication ($50t_{\text{a}} < t_{\text{m}}$ in general) and $N < 2^{50}$ in general, the running time of O_{inner} can be regarded as a few times of multiplication (i.e., $O(1)t_{\text{m}}$).

DS4 Design oracle O_{f}:

$$|\alpha\rangle|\beta\rangle|\boldsymbol{x}\rangle|i\rangle|\boldsymbol{W}_i\rangle|(\boldsymbol{W}_i \bullet \boldsymbol{x})^2\rangle \xrightarrow{O_{\text{f}}} (-1)^{f(i)}|\alpha\rangle|\beta\rangle|\boldsymbol{x}\rangle|i\rangle|\boldsymbol{W}_i\rangle|(\boldsymbol{W}_i \bullet \boldsymbol{x})^2\rangle,$$

where $f(i) = \begin{cases} 1 \text{ if } \alpha \le (\boldsymbol{W}_i \bullet \boldsymbol{x})^2 \le \beta, \\ 0 \qquad \text{otherwise.} \end{cases}$

Oracle O_{f} is used to mark the DFT coefficients $\boldsymbol{W}_i \bullet \boldsymbol{x}$ which satisfy the condition $\alpha \le (\boldsymbol{W}_i \bullet \boldsymbol{x})^2 \le \beta$. $\alpha \le (\boldsymbol{W}_i \bullet \boldsymbol{x})^2$ equals to $(\boldsymbol{W}_i \bullet \boldsymbol{x})^2 - \alpha \ge 0$, and the state (i.e., 0 or 1) of the highest qubit of register can be used to mark the sign '+' or '−' of a datum.

DS5 Define 1D QDFT iteration G_{1DQDFT}:

According to Eq.(5) 1D QDFT iteration G_{1DQDFT} is

$$G_{\text{1DQDFT}} = (2|\xi\rangle\langle\xi| - I)(U_{\text{L}})^{-1}(O_{\text{inner}})^{\dagger}O_{\text{f}}O_{\text{inner}}U_{\text{L}}. \tag{6}$$

2.2. Design the following subroutine 1 to find a coefficient $\boldsymbol{W}_{i_0} \bullet \boldsymbol{x}$ that satisfies the condition $\alpha \le (\boldsymbol{W}_{i_0} \bullet \boldsymbol{x})^2 \le \beta$

subroutine 1:

Step 1 Initialize $m = 1$ and set $\lambda = 6/5$. (Any value of λ strictly between 1 and 4/3 would do.)

Step 2 Choose j uniformly at random among the nonnegative integers smaller than m.

Step 3 Apply j iterations of G_{1DQDFT} acting on initial state $|\psi_0\rangle = \frac{1}{\sqrt{N}}|\alpha\rangle|\beta\rangle|\boldsymbol{x}\rangle\left(\sum_{i=0}^{N-1}|i\rangle\right)|0\rangle|0\rangle$.

Step 4 Observe the 4th register: let i_0 be the outcome.

Step 5 Calculate value $\boldsymbol{W}_{i_0} \bullet \boldsymbol{x}$ by using classical computation. If $\alpha \le (\boldsymbol{W}_{i_0} \bullet \boldsymbol{x})^2 \le \beta$, preserve i_0, and exit.

Step 6 Otherwise, set m to $\min(\lambda m, \sqrt{N})$ and go back to step 2.

Subroutine 1 is similar to BBHT algorithm,[10] and the main difference between them is that Grover iteration is replaced by 1D QDFT iteration G_{1DQDFT}. Subroutine 1 has time complexity $O\left(\sqrt{\frac{N}{M}}\right)$,[10] where M denotes the number of coefficients $\boldsymbol{W}_i \bullet \boldsymbol{x}$ that satisfy the condition $\alpha \le (\boldsymbol{W}_i \bullet \boldsymbol{x})^2 \le \beta$.

2.3. Design the following 1D QDFT algorithm

Step 0 Let $\Delta E = ||\boldsymbol{x}||^2 = \sum_{i=0}^{N-1}(x_i \times x_i)$, $\alpha = \frac{\Delta E}{N}$, $\beta = \Delta E$, $nS = 0$. We can design a very simple parallel circuit to calculate value ΔE, and the parallel circuit has computation complexity $O(1)$ (unit time: multiplication) approximately.

Step 1 Generate the initial state $|\psi_0\rangle = \frac{1}{\sqrt{N}}|\alpha\rangle|\beta\rangle|\boldsymbol{x}\rangle\left(\sum_{i=0}^{N-1}|i\rangle\right)|0\rangle|0\rangle$. This can be achieved in $O(\log_2 N)$ steps using a $[\log_2 N]-$bit Hadamard transform.

Step 2 While $\left(\frac{\Delta E}{||\boldsymbol{x}||^2} \ge \varepsilon\right)$, where ε is the given threshold.

{

Step 2.1 Apply subroutine 1 to find a coefficient $c[i_0] = \boldsymbol{W}_{i_0} \bullet \boldsymbol{x}$ that satisfies the condition $\alpha \le (c[i_0])^2 \le \beta$.

Step 2.2 If the DFT coefficient $c[i_0]$ is the result that is not found previously, preserve it, and let $\Delta E = \Delta E - (c[i_0])^2$, $nS = nS + 1$, $\alpha = \frac{\Delta E}{N - nS}$, and $\beta = \Delta E$, where α denotes the average residual energy per DFT coefficient which is not obtained still by this algorithm, β denotes the total residual energy that is preserved by the DFT coefficients which are not obtained still, and nS denotes the number of the coefficients which have been obtained.

}

It is the main idea of step 2.1 that apply subroutine 1 to find a coefficient whose energy is bigger than the average residual energy (i.e., value α) and smaller than the total residual energy (i.e., value

β). Because DFT is an energy conservation transform (i.e., $\sum_{i=0}^{N-1}(x_i)^2 = \sum_{i=0}^{N-1}(c_i)^2$), the stop criterion $\frac{\Delta E}{||\boldsymbol{x}||^2} \geq \varepsilon$ shows that the above algorithm will find all big coefficients that preserve nearly all energy and information if the threshold ε is small enough. In addition, almost signal sequences have high redundance and the main task of DFT is to find and retain big coefficients to eliminate this redundance (see Fig.1). Therefore, the 1D QDFT algorithm realizes 1D DFT computation approximately.

1D QDFT has time complex $O(\sqrt{mN})$ approximately, where m denotes the number of the big coefficients. Because signal sequences have high redundance in general, the case $m \ll N$ often occurs (see Fig.1). Thus, we say that the 1D QDFT has time complexity $O(\sqrt{N})$ (unit: multiplication and addition), while the classical 1D FFT has time complexity $O(N \log_2 N)$ (unit: multiplication and addition) and even parallel DFT computation has time complexity $O(N)$. QDFT can process a long signal sequence at a time, while the classical computer requires that a long signal sequence be divided into many small sections (such as 8×8 or 16×16 section).

3. 2D QDFT

The 2D DFT (see Eq.(3)) is a separable linear transformation. That is, the result of 2-D DFT may be obtained by first taking transforms along the columns of $\boldsymbol{F}$ and then along the rows of that result,[1] i.e., $C = W_N F W_N = (W_N F) W_N$. We define

$$G = W_N F = (\boldsymbol{W}_i \bullet \boldsymbol{f}_j)_{N \times N},$$

where $\boldsymbol{W}_i$ is the line vector of matrix W_N and $\boldsymbol{f}_j$ is the column vector of input matrix $\boldsymbol{F}$.

The main task of 2D QDFT is to calculate out matrix $\boldsymbol{G}$ and matrix GW_N (i.e., C). 2D QDFT is described as follows.

First Construct the following DS and unitary operations:

DS1 Store all of line vector $\boldsymbol{W}_i$ in a database, and each index of line vector is denoted by i. Store all of column vector $\boldsymbol{f}_j$ of matrix $\boldsymbol{F}$ in the database, and each index of column vector is denoted by j.

DS2 Construct seven registers which have data format $|\alpha\rangle|\beta\rangle|i\rangle|j\rangle|\boldsymbol{W}_i\rangle|\boldsymbol{f}_j\rangle|(\boldsymbol{W}_i \bullet \boldsymbol{f}_j)^2\rangle$.

DS3 Design two loading operation $U_{\rm L1}$ and $U_{\rm L2}$ according to Eq.(4) and Ref.[10], where $U_{\rm L1}$ and $U_{\rm L2}$ are applied to load vector $\boldsymbol{W}_i$ and $\boldsymbol{f}_j$ into registers from classical database respectively. That is,

$$\begin{cases} |\alpha\rangle|\beta\rangle|i\rangle|j\rangle|0\rangle|0\rangle|0\rangle \xrightarrow{U_{\rm L1}} |\alpha\rangle|\beta\rangle|i\rangle|j\rangle|\boldsymbol{W}_i\rangle|0\rangle|0\rangle, \\ |\alpha\rangle|\beta\rangle|i\rangle|j\rangle|0\rangle|0\rangle|0\rangle \xrightarrow{U_{\rm L2}} |\alpha\rangle|\beta\rangle|i\rangle|j\rangle|0\rangle|\boldsymbol{f}_j\rangle|0\rangle. \end{cases}$$

DS4 Design oracle $B_{\rm inner}$ to calculate squared inner product. That is,

$$|\alpha\rangle|\beta\rangle|i\rangle|j\rangle|\boldsymbol{W}_i\rangle|\boldsymbol{f}_j\rangle|0\rangle \xrightarrow{B_{\rm inner}} |\alpha\rangle|\beta\rangle|i\rangle|j\rangle|\boldsymbol{W}_i\rangle|\boldsymbol{f}_j\rangle|(\boldsymbol{W}_i \bullet \boldsymbol{f}_j)^2\rangle.$$

DS5 Design oracle $O'_{\rm f}$:

$$|\alpha\rangle|\beta\rangle|i\rangle|j\rangle|\boldsymbol{W}_i\rangle|\boldsymbol{f}_j\rangle|(\boldsymbol{W}_i \bullet \boldsymbol{f}_j)^2\rangle \xrightarrow{O'_{\rm f}} (-1)^{f'(i,j)}|\alpha\rangle|\beta\rangle|i\rangle|j\rangle|\boldsymbol{W}_i\rangle|\boldsymbol{f}_j\rangle|(\boldsymbol{W}_i \bullet \boldsymbol{f}_j)^2\rangle,$$

where $f'(i,j) = \begin{cases} 1 \text{ if } \alpha \leq (\boldsymbol{W}_i \bullet \boldsymbol{f}_j)^2 \leq \beta, \\ 0 \qquad \text{otherwise.} \end{cases}$

DS6 Design 2D QDFT iteration $G_{\rm 2DQDFT}$:

$$G_{\rm 2DQDFT} = (2|\xi'\rangle\langle\xi'| - I)(U_{\rm L1})^\dagger (U_{\rm L2})^\dagger (B_{\rm inner})^\dagger O'_{\rm f} B_{\rm inner} U_{\rm L2} U_{\rm L1},$$

where $|\xi'\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\underbrace{|i\rangle|j\rangle}$.

Second Generate the initial state $|\Psi_0\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|\alpha\rangle|\beta\rangle\underbrace{|i\rangle|j\rangle}|0\rangle|0\rangle|0\rangle$. And apply the method of 1D QDFT, we will calculate out all elements of matrix $\boldsymbol{G}$. Apply the same method again, we can calculate out all elements of matrix $\boldsymbol{C}$.

2D QDFT has time complexity $O(N)$, while classical 2D FFT has time complexity $O(N^2\log_2 N)$.

4. Using 1D QDFT to calculate convolution

For the given two N-dimensional vectors $\boldsymbol{u} = (u_0, u_1, ..., u_{N-1})$ and $\boldsymbol{v} = (v_0, v_1, ..., v_{N-1})$, let $w_k = \sum_{j=0}^{N-1} u_j v_{(k-j)\bmod N}$, where $0 \le k < N$. Then vector $\boldsymbol{w} = (w_0, w_1, ..., w_{N-1})$ is called the periodic convolution of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. We often denote the convolution by symbol $\boldsymbol{w} = \boldsymbol{u} * \boldsymbol{v}$. If $\boldsymbol{u}$ and $\boldsymbol{v}$ have different dimensions, add zero components to each vector such that both vectors have the same dimension. The periodic convolution of the new vectors can be regarded as the result. Algebraically, convolution is the same operation as multiplying the polynomials whose coefficients are the elements of $\boldsymbol{u}$ and $\boldsymbol{v}$.[1]

Holding the view of signal processing, we can regard vector $\boldsymbol{u}$ as input signal sequence, and vector $\boldsymbol{v}$ is the function of a physical apparatus. After vector $\boldsymbol{u}$ passing through the apparatus, convolution $\boldsymbol{w}$ comes out. Furthermore, the output sequence $\boldsymbol{w}$ becomes smoother than the input sequence $\boldsymbol{u}$ in general. Therefore, convolution is very important for signal or information processing.[1]

The convolution theorem[1] says, roughly, that convolving two sequences is the same as multiplying their Fourier transforms. That is,[1]

$$\mathrm{DFT}(\boldsymbol{u} * \boldsymbol{v}) = \mathrm{DFT}(\boldsymbol{u})\mathrm{DFT}(\boldsymbol{v}). \tag{7}$$

In general, the big coefficients of $\mathrm{DFT}(\boldsymbol{v})$ are few and other coefficients have little contribution to the result. That is, we can calculate only big coefficients of $\mathrm{DFT}(\boldsymbol{u})$ and $\mathrm{DFT}(\boldsymbol{v})$ and discard small coefficients to speedup the computation of convolution. According to this property, quantum algorithm for convolution estimation is described as follows:

Step 1 Calculate $\mathrm{DFT}(\boldsymbol{u})$ and $\mathrm{DFT}(\boldsymbol{v})$ using the 1D QDFT.

Step 2 Calculate $\mathrm{DFT}(\boldsymbol{u}*\boldsymbol{v})$ according to Eq.(7).

Step 3 Calculate the inverse of $\mathrm{DFT}(\boldsymbol{u} * \boldsymbol{v})$ to obtain convolution $\boldsymbol{u} * \boldsymbol{v}$.

It should be noted that the efficiency of the above algorithm depends on the redundancy (or smooth property) of the signal sequences. The higher the redundancy is, the faster the quantum algorithm runs. Fortunately, many digital signal sequences have high redundancy, which is just the existence reason of modern signal processing technique.

5. Conclusion

Discrete Fourier transform (DFT) is the base of modern signal or information processing. No DFT, no modern signal and information processing, and no modern information science. The fast DFT plays the key role of signal processing. 1-dimensional fast Fourier transform (1D FFT) and 2D FFT have time complexity $O(N\log N)$ and $O(N^2\log N)$ respectively. Since 1965, there is no more essential breakthrough for the design of fast DFT algorithm, which is based on electronic computer. Maybe it is impossible to design much more fast DFT algorithm basing on classical computation principle. Quantum computation offers the chance of designing fast DFT algorithm possibly.

Quantum Fourier transform (QFT) is the key of the famous Shor's factoring algorithm. QFT has time complexity $O(\log^2 N)$. However, QFT is not suitable to classical signal and information processing because the result of QFT is a superposition of states, not some digital numbers.

DFT have two properties, which is used to design quantum DFT algorithm (QDFT). One property is that DFT is energy conservation transform, which is used to design termination criterion of algorithm. The other property is that many DFT coefficients are close to zero, have little information, and can be discarded. The second property demonstrates that we can use quantum computation to search all big coefficients, which contain nearly all energy and nearly all information.

Grover's algorithm is an efficient quantum search algorithm. However, Grover's algorithm cannot be applied to DFT directly. The function of Grover's algorithm is to find the index of a special database record from the superposition of state that is comprised by all indices of records, while actual coefficients is the genuine answer of DFT. In addition, the computation process of every coefficient should be per-

formed. Thus, the computation of DFT coefficient and Grover's search should be coupled as a whole. And the generalized Grover iteration presented by Pang *et al* can realize this coupling. In addition, classical signal is often stored in electronic memory and its information should be loaded into quantum registers. Pang presented a quantum loading scheme that can load all information of classical signal into two entangled registers.

It is the basic idea of this paper that the generalized Grover's iteration acts on the entangled states to search big DFT coefficients until these big coefficients contain nearly all energy (i.e., nearly all information of signal sequence).

The one-dimensional quantum DFT (1D QDFT) and two-dimensional quantum DFT (2D QDFT) are presented in this paper. And quantum algorithm for convolution estimation is also presented in this paper. Compared with fast Fourier transformation, QDFT has at least two advantages. One of the advantages is that 1D and 2D QDFT have time complexity $O(\sqrt{N})$ and $O(N)$ respectively, while 1D and 2D FFT have time complexity $O(N \log N)$ and $O(N^2 \log N)$ respectively. The other advantage is that QDFT can process very long signal sequence at a time. DFT is the base of modern signal and information process. QDFT and quantum convolution demonstrate that quantum computation to process classical signal is possible.

Acknowledgments

The first author thanks Professor Guo G C and the Key Laboratory of Quantum Information, USTC, China. The first author also thanks Professor Han Z F for his encouragement and help. The first author further thanks Dr Zhou Z W help and useful discussion from 2004 to 2006, and Professor Gorbachev V N of St.-Petersburg State University of Aerospace Instrumentation for a discussion about quantum discrete cosine transform in 2006. The authors also thank Dr Cai J M, Professor Zhang J and Professor Pu Z L for their help.

References

[1] Hu G S 1997 *Digital Signal Processing* (Beijing: Tsinghua University Press) p55

[2] Pang C Y, Zhou Z W and Guo G C 2006 *Chin. Phys.* **15** 3039

[3] Shor P W 1994 *Proc. 35th Annual Symposium on the Foundations of Computer Science* (Los Alamitos, CA: IEEE Computer Society Press) p20

[4] Galindo A and Martín-Delgado M A 2002 *Rev. Mod. Phys.* **74** 347

[5] Lu C Y, Browne D E, Yang T and Pan J W 2007 *Phys. Rev. Lett.* **99** 250504

[6] Nielsen M A and Chuang I L 2002 *Quantum Computationand and Quantum Information* (Cambridge: Cambridge University Press) p216

[7] Nielsen M A and Chuang I L (translator: Zhao Q C) *Quantum Computationand and Quantum Information* (Beijing: Tsinghua University Publishers) p198

[8] Grover L K 1996 *Proc. 28th Annual ACM Symposium on the Theory of Computing* (Philadelphia, Pennsylvania: ACM Press) p212

[9] Long G L 2001 *Phys. Rev.* A **64** 022307

[10] Boyer M, Brassard G, Hoyer P and Tap A 1996 quant-ph/9605034

[11] Pang C Y 2006 quant-ph/0612061

[12] Pang C Y, Zhou Z W, Chen P X and Guo G C 2006 *Chin. Phys.* **15** 618

[13] Pang C Y, Zhou Z W and Guo G C 2006 quant-ph/0601043

[14] Pang C Y 2006 *Quantum Image Compression* Postdoctoral report, Key Laboratory of Quantum Information, University of Science and Technology of China p44