A theoretical framework for quantum image representation and data loading scheme

BenQiong HU, XuDong HUANG, RiGui ZHOU, YanYu WEI, Qun WAN and ChaoYang PANG

Citation: SCIENCE CHINA Information Sciences **57**, 032108(11) (2014); doi: 10.1007/s11432-013-4866-x

SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

March 2014, Vol. 57 032108:1–032108:11
doi: 10.1007/s11432-013-4866-x

A theoretical framework for quantum image representation and data loading scheme

HU BenQiong[1], HUANG XuDong[2], ZHOU RiGui[3], WEI YanYu[4], WAN Qun[5] & PANG ChaoYang[6*]

[1]*College of Management Science, Chengdu University of Technology, Chengdu* 610059*, China;*
[2]*Neurochemistry Laboratory, Department of Psychiatry,*
Massachusetts General Hospital and Harvard Medical School, Charlestown, MA 02129*, USA;*
[3]*College of Information Engineering, East China JiaoTong University, Nanchang* 330013*, China;*
[4]*Vacuum Electronics National Laboratory, School of Physical Electronics,*
University of Electronic Science and Technology of China, Chengdu 610054*, China;*
[5]*Department of Electronic Engineering, University of Electronic Science and Technology of China,*
Chengdu 610054*, China;*
[6]*Group of Gene Computation, College of Computer Science, Sichuan Normal University, Chengdu* 610066*, China*

Received July 4, 2013; accepted December 16, 2013

Abstract Two fundamental problems exist in the use of quantum computation to process an image or signal. The first one is how to represent giant data, such as image data, using quantum state without losing information. The second one is how to load a colossal volume of data into the quantum registers of a quantum CPU from classical electronic memory. Researches on these two questions are rarely reported. Here an entangled state is used to represent an image (or vector) for which two entangled registers are used to store a vector component and its classical address. Using the representation, $n_1 + n_2 + 8$ qubits are used to store the whole information of the gray image that has a $2^{n_1} \times 2^{n_2}$ size at a superposition of states, a feat is not possible with a classic computer. The way of designing a unitary operation to load data, such as a vector (or image), into the quantum registers of a quantum CPU from electronic memory is defined herein as a quantum loading scheme (QLS). In this paper, the QLS with time complexity $O(\log_2 N)$ is presented where N denotes the number of vector components, a solution that would break through the efficiency bottleneck of loading data. QLS would enable a quantum CPU to be compatible with electronic memory and make possible quantum image compression and quantum signal processing that has classical input and output.

Keywords quantum, image representation, path interference, entangled state, quantum loading scheme

Citation Hu B Q, Huang X D, Zhou R G, et al. A theoretical framework for quantum image representation and data loading scheme. Sci China Inf Sci, 2014, 57: 032108(11), doi: 10.1007/s11432-013-4866-x

1 Introduction

Image storing and processing are the foundations of modern multimedia communications such as a JPEG picture file and DVD film. A digital image is often represented as a matrix or vector [1]. An original monochromatic image with size $N \times N$ is stored in classical memory (i.e., electronic memory) one byte

*Corresponding author (email: cypang@live.com)

by one byte. In total, $O(N \times N)$ bytes are needed to store the whole image. For example, 256 kb are required to store a gray picture with a size of 512×512. That size represents a large amount of data. If we need to process an image with size $N \times N$, a classical computer (i.e., electronic computer) has to load the image data, one byte by one byte into the central processing unit (CPU) from classical memory. Thus, $N \times N$ steps of the loading scheme are needed [2]. This loading scheme decreases the efficiency of computation. In terms of arbitrary classical algorithms, N data needs N steps of loading operation for a single processor [2], and this case decreases the efficiency of computation and results in the bottleneck of electronic computer. Furthermore, frequently the efficiency bottleneck will be encountered and there is no way to evade it because electronic computer processing image has to load a large amount of data into CPU one datum by one datum with low efficiency. In other words, only a different kind of computer can break through the efficiency bottleneck. Which kind of computer can break through the bottleneck? We propose that quantum computation can. Fortunately, in the last decades, quantum computation is well studied, and many special computation properties have been discovered so that quantum computation has become one of the most active research areas in computational science and engineering.

In 1982, Feynman argued that simulating quantum mechanics inherently required an exponential amount of overhead, and first presented quantum computation [3]. In 1985, Deutsch presented Deutsch's algorithm [4]. In 1992, Deutsch and Jozsa presented the Deutsch-Jozsa algorithm [5], which was then improved in 1997 by Cleve et al. [6]. The quantum computation for contrived-appearing problems [4–6] shows that a quantum computer seems to work better than classical computers [7]. In 1994, Simon presented Simon's algorithm [8], and Shor said that, based on that work, he figured out how to design the factoring algorithm [7]. In 1994, a milestone in quantum computation appeared when Shor presented Shor's algorithm for factoring an integer number with polynomial computation steps, an effort believed to be classically impossible [9]. In 1996, another famous algorithm appeared when Grover presented Grover's algorithm for finding an item in an unsorted database with $O(\sqrt{N})$ computation steps. Compared to that, a classical algorithm needs $O(N)$ steps [10]. Quantum Fourier transform (QFT) plays an important role in quantum algorithm. Coppersmith [11], Robert et al. [12], Kitaev [13], and Cleve et al. [6] studied QFT very well. Quantum phase estimation and eigenvalue estimation play another important role in quantum algorithm. In 1995, Kitaev presented quantum phase estimation and Mosca presented eigenvalue estimation [14]. Other quantum algorithms, such as quantum walk algorithms, adiabatic algorithms, topological algorithms, are also well studied [15]. One wonderful conclusion was that many quantum algorithms can be classified into the framework of hidden subgroup problems [16]. Recently, Mosca presented a review of the quantum algorithm [15].

Quantum algorithm depends on a quantum circuit model for it to be realized. In 1993, Yao presented the quantum Turing machine model (QTM) [17] that was subsquently thoroughly studied in 1997 by Bernstein et al. [18]. The theory of QTM establishes a rigorous mathematical model for a quantum computer. Recently, Watrous presented a review for quantum computational complexity [19].

Quantum computation not only has its own rigorous mathematical model, but also is supported by some experiments. For example, Shor's algorithm has been used on an NMR-based quantum computer (Vandersypen et al. [20]) and optical quantum computer [21] to factorize the number [19] so as to act as a proof of the quantum computation principle. In addition, entanglement plays a profound and useful role for quantum computation. Many famous scientists have studied entanglement. For these great contributions, please refer to [22–28].

Many scientists have been constructing a good foundation for quantum computation, the use of which may help solve the complicated classical problems. In fact, a plethora of other classically challenging problems also exist, such as pattern recognition signal processing, image compression, and image processing. All of these issues may possibly benefit from the application of quantum algorithms. For example, quantum information and quantum computation can be applied to data compression at quantum state [29–33], image compression [34–38], radar signal processing [39] and pattern recognition [40–42], all of which generally have a huge amount of input data stored in classical electronic memory.

In quantum image compression [34–38], quantum radar signal processing [39], pattern recognition [40–42], and input data are classical data not quantum state, so the output should be classical data but not quantum state, and only the processing of computation is quantum computation. Since huge input data is stored in electronic memory and has to be processed by a quantum CPU, a scheme should exist to put both electronic memory and quantum CPU together well to efficiently load data into the superposition of states. Furthermore, the scheme has to obey the principle of quantum mechanics. In this paper, such a scheme is called quantum loading scheme (QLS). Nielsen et al. [43] pointed out that QLS should exist. Giovannetti et al. [44,45] presented an interesting quantum method to access memory by which classical bit 0 and 1 are loaded into an entangled state (single bit is entangled with its physical storing address). It is interesting to design QLS to access a memory cell stored in an arbitrary array of real number (i.e., vector) by which all numbers (i.e., components of a vector) can be loaded into the entangled state (component is entangled with its logical address or its subscript).

2 Introduction of quantum computation

Like the classical bit that has a state—either 0 or 1, a qubit also has a state. Two possible states for a qubit are the states $|0\rangle$ and $|1\rangle$, which correspond to the states 0 and 1 for a classical bit. Notation like "$|\rangle$" is called the Dirac notation. It is possible to form linear combinations of states, often called superposition $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where numbers α and β are complex numbers. We can examine a bit to determine whether it is in the state 0 or 1 in a classical computer. By contrast, when we measure a qubit, we get either the result 0, with probability $|\alpha|^2$, or the result 1, with probability $|\beta|^2$, where $|\alpha|^2 + |\beta|^2 = 1$ [43].

Quantum parallelism allows quantum computers to evaluate a function $f(x)$ for many different values of x simultaneously. The power of quantum computation is due to the fact that the state of a quantum computer can be a superposition of basis states, and we can perform an operation on multiple quantum states simultaneously. For example, suppose $f(x) : \{0, 1\} \to \{0, 1\}$ is a function with a one-bit domain and range. An electronic computer needs at least two times of calculation to obtain the values $f(0)$ and $f(1)$. For a function $f(x)$, there is quantum circuit U_f such that

$$\frac{|0\rangle|y\rangle + |1\rangle|y\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{|0\rangle|y \oplus f(0)\rangle + |1\rangle|y \oplus f(1)\rangle}{\sqrt{2}}$$

is named the black box of computation (i.e., oracle). The U_f makes $|0, y\rangle$ and $|1, y\rangle$ converted into $|0, y \oplus f(0)\rangle$ and $|1, y \oplus f(1)\rangle$ at the same time, which is a parallel operation [43,46].

In addition, giant molecules, such as charcoal C_{60}, exhibit the quantum interference [47], which is the essential characteristic of quantum computation. A giant molecule has many degrees of freedom, and information can be represented by these degrees. Thus, giant molecules can preserve much information, making them advantageous for quantum computation. In 2006, Hosten et al. [48] presented a method of using path interference and a Zeno effect to improve the solution precision of Grover's algorithm. As is well known, the unitary operation is the basic requirement of quantum computation. The motivation of this paper is to present a scheme to load image data into quantum state from electronic memory, and make a switch between electronic data and quantum state so as to make electronic device and quantum computer compatible.

3 Quantum representation of vector (or image) and the model of quantum loading scheme (QLS)

Image data is often represented as a matrix or vector and stored in classical memory, where a matrix can also be characterized as a vector equivalently. A vector is the general format for the input data of an algorithm. In general, each component of a vector is stored sequentially in classical memory. Thus, two questions arise about the general quantum algorithm. The first one: which state is suitable to represent all information of the vector for further quantum computation? The second one: how can all of the information of the vector be loaded into quantum registers without losing information? Loading data into the electronic registers of classical CPUs from electronic memory is called classical loading scheme (CLS). Similar to CLS, designing unitary operations to load all information holden by a vector into the quantum registers of the quantum CPU from electronic memory is called a quantum loading scheme (QLS). CLS or QLS assembles electronic memory and CPU as a whole computer. QLS makes a quantum CPU compatible with electronic memory. Therefore the study of QLS is likely an important step for the future quantum computer.

3.1 Quantum representation of vector (or image)

In computer science, a binary is often used to represent data. For example, the real number 3 can be represented as 2 bits binary sequence $(11)_{\text{binary}}$, and saving the two bits is equivalent to saving the real number. The storing address of the first bit of binary sequence $(11)_{\text{binary}}$ represents the storing address of real number 3, and the CPU accesses this binary sequence according to its address. For vectors $\{1, 3\}$, they have two binary sequences $(01)_{\text{binary}}$ and $(11)_{\text{binary}}$ respectively, the sequential preservation of the two binary sequences is the saving manner of the vector, and the address of the first bit is the accessing address of CPU [49].

Consider an N-dimensional vector

$$\boldsymbol{a} = \{a_0, a_1, \ldots, a_{N-1}\}, \tag{1}$$

where the components $a_0, a_1, \ldots, a_{N-1}$ are integer numbers.

In formula (1), let $N = 2^n$, and if $N \neq 2^n$, and add some zero to make $N = 2^n$.

Let $m = \lceil \log_2(\max\{|a_0|, |a_1|, \ldots, |a_{N-1}|\}) \rceil + 1$. In general, at most m bits can save the value of every component a_i, in which one bit is used to represent the plus-minus of data. All corresponding binary sequences of components $a_0, a_1, \ldots, a_{N-1}$ are stored in electronic memory sequentially. $m \times n$ bits are needed to save vector $\boldsymbol{a}$.

An electronic CPU can access an arbitrary component a_i from electronic memory according to the corresponding subscript i because the subscript is corresponding to its storing address, which is analogous to the case of accessing somebody a_i according to the number of room ith. Thus, the one-to-one mapping relationship between subscript and component is the primary property for the electronic CPU accessing the vector. Therefore, the whole information of vector $\boldsymbol{a}$ not only includes the value of component a_i and the value of subscript i, but also includes the one-to-one mapping relationship between component a_i and subscript i. Thus, the whole information of the vector consist of three parts, component a_i, subscript i, and the one-to-one mapping relationship between them, and any storing manner should include the three parts completely. In this paper, we will study how to use the quantum state to represent all of the information held by vector and how to load all of the information of the vector into the quantum state. And quantum representation of the vector (or image) is defined as the superposition of states, by which the whole information of the vector is represented.

Let

$$|\text{qVector}\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle_{q_1 q_2 \ldots q_n} |a_i\rangle_{p_1 p_2 \ldots p_m} |\text{ancilla}\rangle, \tag{2}$$

where the n qubits $q_1, q_2, \ldots, q_n$ and the m qubits $p_1, p_2, \ldots, p_m$ are used to save subscript i and the corresponding value a_i respectively, and the ancillary state $|\text{ancilla}\rangle$ is known. Some degrees of freedom of wave particle, such as duality of C_{60} molecules [47], can be used as these $m + n$ qubits that preserve the information on the vector.

In (2), register $|i\rangle_{q_1 q_2 \ldots q_n}$ is entangled with register $|a_i\rangle_{p_1 p_2 \ldots p_m}$. Furthermore, the entanglement represents the one-to-one mapping relationship between subscript i and the corresponding component a_i. Thus, the whole information on vector $\boldsymbol{a}$ is preserved in the entangled state $|\text{qVector}\rangle$. Therefore, the state $|\text{qVector}\rangle$ is suitable to represent the vector (or image). The whole information of the vector is hidden in the superposition of states.

To understand the relationship between the superposition of state $|\text{qVector}\rangle$ and the information contained by it, the following discussion is given: 1) According to state collapse theory of quantum measurement, doing one time measurement generates one observed result. That is, only one component of

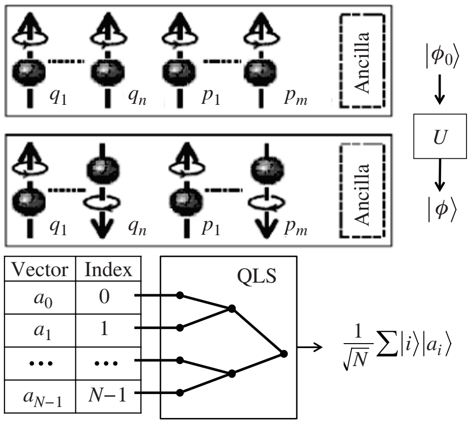

Figure 1 The illustration of the model of QLS: The focus of this paper is to design the unitary operation $U_{(0,1,\ldots,N-1)}$ (i.e., QLS) that satisfies formula (4). The function of QLS has efficiently loaded all of the information of vector into two entangled registers (components are entangled with their subscripts) from electronic memory.

the vector is obtained from the superposition of states after executing a measurement. In addition, the probability of abtaining the ith component a_i is $1/N$. Thus, all information of the vector is merely hidden in the state $|\text{qVector}\rangle$, and is not completely accessible under the orthogonal basis $|a_i\rangle_{p_1p_2\ldots p_m}$. 2) To extract wanted information from the superposition of states with high probability, unitary operation is needed. For example, as in the famous quantum search algorithm, Grovers algorithm, all data is represented by a superposition of states, and then a given unitary operation acts on the superposition. After doing several times unitary transformation, the superposition collapses on a wanted answer with nearly 100% probability. The intrinsic characteristic of quantum computation is to make a superposition of states to contain information and then design reasonable unitary operation to extract wanted answer. The aim of this paper is to make a superposition of states and load the information of digital image data into it for further image processing by quantum computation.

Since the information of vector is presented by $|\text{qVector}\rangle$, a unitary operation acting on $|\text{qVector}\rangle$ is equivalent to operating the vector information, and measuring the last state leads to a last classical result (i.e., the answer of a question). That is, the representation in formula (2) is one of the quantum representations of vector information, operating it is equivalent to operating vector information and executing computation. For example, based on this representation, the discrete Fourier transform using quantum computation (QDFT) is presented which generates classical output [39].

Although the information hidden in state $|\text{qVector}\rangle$ is not completely accessible, the representation and two-dimensional QDFT improve the running speed from classical running time $O(N^2 \log_2 N)$ to quantum running time $O(N)$ [39]. That is, the quantum representation $|\text{qVector}\rangle$ helps improve running speed compared with classical computation.

3.2 The model of quantum loading scheme (QLS)

Let the initial state $|\phi_0\rangle$ be

$$|\phi_0\rangle = |0\rangle_{q_1q_2\ldots q_n}|0\rangle_{p_1p_2\ldots p_m}|\text{ancilla}_1\rangle. \tag{3}$$

The focus of this paper is to design the unitary operation $U_{(0,1,\ldots,N-1)}$ such that

$$|\phi_0\rangle \xrightarrow{U_{(0,1,\ldots,N-1)}} |\phi\rangle = \frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle_{q_1q_2\ldots q_n}|a_i\rangle_{p_1p_2\ldots p_m}\right)|\text{ancilla}_2\rangle, \tag{4}$$

where $N = 2^n$ and ancillary state $|\text{ancilla}_2\rangle$ is known. Figure 1 illustrates the model of formula (4).

4 The design of QLS

4.1 Loading 2D vector into quantum registers from electronic memory

The design of the unitary operation $U_{(0,1)}$ that loads the 2D vector is described conceptually as follows (see Figure 2).

Step 1: The switch S_1 applies rotation on the initial ancilla state and transforms $|\mathrm{Off}_0\rangle$ into

$$|\mathrm{Off}_0\rangle \xrightarrow{S_1} \frac{|\mathrm{Off}_1\rangle + |\mathrm{On}_1\rangle}{\sqrt{2}}$$

to generate the following state $|\phi_1\rangle$:

$$|\phi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2\cdots q_n}|0\rangle_{p_1p_2\cdots p_m}|\mathrm{On}_1\rangle + \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2\cdots q_n}|0\rangle_{p_1p_2\cdots p_m}|\mathrm{Off}_1\rangle. \quad (5)$$

Step 2: Perform unitary operations I_0 and A_0 along "On_1" path, and perform unitary operations I_1 and A_1 along "Off_1" path simultaneously.

$$\begin{cases} |0\rangle_{q_1q_2\ldots q_n} \xrightarrow{I_0} |0\rangle_{q_1q_2\ldots q_n}, \\ |0\rangle_{p_1p_2\ldots p_m} \xrightarrow{A_0} |a_0\rangle_{p_1p_2\ldots p_m}, \end{cases} \quad \begin{cases} |0\rangle_{q_1q_2\ldots q_n} \xrightarrow{I_1} |1\rangle_{q_1q_2\ldots q_n}, \\ |0\rangle_{p_1p_2\ldots p_m} \xrightarrow{A_1} |a_1\rangle_{p_1p_2\ldots p_m}. \end{cases}$$

We can set a timer to ensure that the operations of different pathes are performed at the same time such that the outputs of two paths are simutaneous.The following state $|\phi_2\rangle$ is generated according to Figure 3:

$$\begin{cases} \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2\ldots q_n}|0\rangle_{p_1p_2\ldots p_m}|\mathrm{On}_1\rangle \xrightarrow{A_0I_0} \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\ldots p_m}|\mathrm{On}_1\rangle, \\ \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2\ldots q_n}|0\rangle_{p_1p_2\ldots p_m}|\mathrm{Off}_1\rangle \xrightarrow{A_1I_1} \frac{1}{\sqrt{2}}|1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\ldots p_m}|\mathrm{Off}_1\rangle, \end{cases}$$

$$\Longrightarrow |\phi_2\rangle = \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\ldots p_m}|\mathrm{On}_1\rangle + \frac{1}{\sqrt{2}}|1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\ldots p_m}|\mathrm{Off}_1\rangle. \quad (6)$$

Step 3: The switch S_2 applies rotation on the initial ancilla state as

$$\begin{cases} |\mathrm{Off}_1\rangle \xrightarrow{S_2} \frac{|\mathrm{Off}_2\rangle - |\mathrm{On}_2\rangle}{\sqrt{2}}, \\ |\mathrm{On}_1\rangle \xrightarrow{S_2} \frac{|\mathrm{Off}_2\rangle + |\mathrm{On}_2\rangle}{\sqrt{2}}. \end{cases}$$

The switch S_2 generates the following state $|\phi_3\rangle$:

$$\begin{aligned} |\phi_3\rangle = {} & \frac{1}{2}\Big(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\ldots p_m} + |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\ldots p_m}\Big)|\mathrm{Off}_2\rangle \\ & + \frac{1}{2}\Big(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\ldots p_m} - |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\ldots p_m}\Big)|\mathrm{On}_2\rangle. \end{aligned} \quad (7)$$

Step 4: Apply phase transformation B along "On_2" path.

$$B = |0\rangle\,|a_0\rangle\langle a_0|\langle 0| - |1\rangle\,|a_1\rangle\langle a_1|\langle 1|, \quad (8)$$

where the subscript of state is ignored. Operation B is a very fast operation, and it generates the following state $|\phi_4\rangle$:

$$|\phi_4\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\ldots p_m} + |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\ldots p_m}\right)\frac{|\mathrm{Off}_2\rangle + |\mathrm{On}_2\rangle}{\sqrt{2}}. \quad (9)$$

Step 5: The switch S_3 applies rotation on the initial ancilla state as

$$\begin{cases} |\mathrm{Off}_2\rangle \xrightarrow{S_3} \frac{|\mathrm{Off}_3\rangle + |\mathrm{On}_3\rangle}{\sqrt{2}}, \\ |\mathrm{On}_2\rangle \xrightarrow{S_3} \frac{|\mathrm{Off}_3\rangle - |\mathrm{On}_3\rangle}{\sqrt{2}}, \end{cases}$$

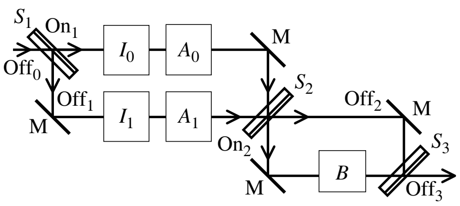

Figure 2 The illustration of the processing of unitary operation $U_{(0,1)}$ that transforms state from $|0\rangle|0\rangle|\mathrm{Off}_0\rangle$ into $\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)|\mathrm{Off}_3\rangle$. M: Mirror

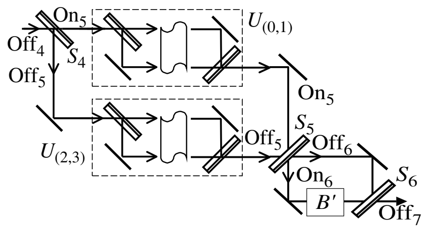

Figure 3 The illustration of unitary operation $U_{(0,1,2,3)}$ that transforms state from $|0\rangle|0\rangle|\mathrm{Off}_4\rangle$ into $(\sum_{i=0}^{3}\frac{1}{2}|i\rangle|a_i\rangle)|\mathrm{Off}_7\rangle$.

and the switch S_3 generates the final state $|\phi\rangle$:

$$|\phi\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\ldots p_m} + |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\ldots p_m}\big)|\mathrm{Off}_3\rangle. \tag{10}$$

The realization of the unitary operations I_0, I_1, A_0 and A_1 in the above algorithm is possible. For example, NMR device can flip the spin states of particles. Let the state of the spin up denote the binary data 0 and the state of spin down denote binary data 1. We can use the binary data of vector component a_i or subscript i as a control signal. According to the control signal, the physical device flips the spin states. Thus, the value of the vector component or subscript is loaded into the quantum state, and the unitary operations I_0, I_1, A_0 and A_1 are realized.

Figure 4 illustrates the physical process of the unitary operations I_0, I_1, A_0 and A_1. Figure 2 and (11) illustrate the structure of operation $U_{(0,1)}$.

$$\begin{aligned}|0\rangle|0\rangle|\mathrm{Off}_0\rangle &\xrightarrow{S_1} \left\langle\begin{array}{c}\frac{1}{\sqrt{2}}|0\rangle|0\rangle|\mathrm{On}_1\rangle \xrightarrow{A_0I_0} \frac{1}{\sqrt{2}}|0\rangle|a_0\rangle|\mathrm{On}_1\rangle\\ \frac{1}{\sqrt{2}}|0\rangle|0\rangle|\mathrm{Off}_1\rangle \xrightarrow{A_1I_1} \frac{1}{\sqrt{2}}|1\rangle|a_1\rangle|\mathrm{Off}_1\rangle\end{array}\right\rangle\\ &\xrightarrow{S_2} \left\langle\begin{array}{c}\frac{1}{2}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)|\mathrm{Off}_2\rangle\\ \frac{1}{2}(|0\rangle|a_0\rangle - |1\rangle|a_1\rangle)|\mathrm{On}_2\rangle \xrightarrow{B} \frac{1}{2}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)|\mathrm{On}_2\rangle\end{array}\right\rangle \xrightarrow{S_3} |\phi\rangle.\end{aligned} \tag{11}$$

In (11), as well as in the following discussions, all subscripts of registers are ignored.

Clearly operation $U_{(0,1)}$ is unitary, and it can be used as a quantum computation.

4.2 Loading 4D vector or multi-dimensional into quantum registers

The design of the unitary operation $U_{(0,1,2,3)}$ is described conceptually as below (see Figure 3):

Step 1: Construct unitary operation S_4, S_5, S_6, B':

$$\left\{\begin{array}{l}|\mathrm{Off}_i\rangle \xrightarrow{S_i} \frac{|\mathrm{Off}_{i+1}\rangle+|\mathrm{On}_{i+1}\rangle}{\sqrt{2}},\\ |\mathrm{On}_i\rangle \xrightarrow{S_i} \frac{|\mathrm{Off}_{i+1}\rangle-|\mathrm{On}_{i+1}\rangle}{\sqrt{2}},\end{array}\right. \quad \left\{\begin{array}{l}|\mathrm{Off}_5\rangle \xrightarrow{S_5} \frac{|\mathrm{Off}_{i+1}\rangle-|\mathrm{On}_{i+1}\rangle}{\sqrt{2}},\\ |\mathrm{On}_5\rangle \xrightarrow{S_5} \frac{|\mathrm{Off}_{i+1}\rangle+|\mathrm{On}_{i+1}\rangle}{\sqrt{2}},\end{array}\right. \quad B' = |\alpha\rangle\langle\alpha| - |\beta\rangle\langle\beta|,$$

where $i = 4, 6$, $|\alpha\rangle = \frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)$ and $|\beta\rangle = \frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle + |3\rangle|a_3\rangle)$.

Step 2: Assemble unitary operations S_4, S_5, S_6, B', $U_{(0,1)}$ and $U_{(2,3)}$ according to Figure 5 to form unitary operations $U_{(0,1,2,3)}$.

(12) illustrates the processing of operation $U_{(0,1,2,3)}$.

$$|0\rangle|0\rangle|\mathrm{Off}_4\rangle \xrightarrow{S_4} \left\langle\begin{array}{c}\frac{1}{\sqrt{2}}|0\rangle|0\rangle|\mathrm{On}_5\rangle \xrightarrow{U_{(0,1)}} \frac{1}{2}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)|\mathrm{On}_5\rangle\\ \frac{1}{\sqrt{2}}|0\rangle|0\rangle|\mathrm{Off}_5\rangle \xrightarrow{U_{(2,3)}} \frac{1}{2}(|2\rangle|a_2\rangle + |3\rangle|a_3\rangle)|\mathrm{Off}_5\rangle\end{array}\right\rangle$$

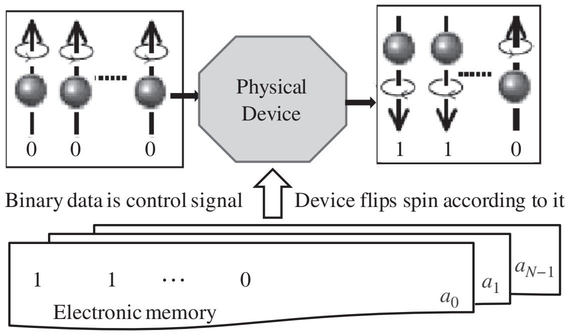

Figure 4 The illustration of the realization I_0, I_1, A_0 and A_1 in step 2. The binary sequences of each component of the vector stored in electronic memory is a control signal. If the bit (i.e., control signal) is 1, let particle spin down. Otherwise, spin up.

$$\xrightarrow{S_5}\left\langle\begin{array}{l}\frac{1}{2}[\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle+|1\rangle|a_1\rangle)+\frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle+|3\rangle|a_3\rangle)]|\mathrm{Off}_6\rangle\\ \frac{1}{2}[\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle+|1\rangle|a_1\rangle)-\frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle+|3\rangle|a_3\rangle)]|\mathrm{On}_6\rangle\xrightarrow{B'}\\ \frac{1}{2}[\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle+|1\rangle|a_1\rangle)+\frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle+|3\rangle|a_3\rangle)]|\mathrm{On}_6\rangle\end{array}\right\rangle\xrightarrow{S_6}|\phi\rangle. \quad (12)$$

If the unitary operations $U_{(0,1)}$ and $U_{(2,3)}$ embedded in Figure 3 are replaced by $U_{(0,1,2,3)}$ and $U_{(4,5,6,7)}$ respectively, then $U_{(0,1,\ldots,7)}$ is constructed. Similar to Figure 3, we can apply the same method to construct unitary operation $U_{(0,1,\ldots,2^n)}$. If $N \neq 2^n$, we could add extra zero components to create a 2^n-dimensional vector.

Clearly, $U_{(0,1,\ldots,2^n)}$ has time complexity $O(\log_2 N)$ (unit time: phase transform and flipping the qubits of registers).

4.3 Loading vector into state $\frac{1}{\sqrt{N}}(\sum_{i=0}^{N-1}|i\rangle|0\rangle)$ to form entangled state $\frac{1}{\sqrt{N}}(\sum_{i=0}^{N-1}|i\rangle|a_i\rangle)$

A database is a set of many records, and each record has an integer as its unique index. In classical computer science, a record is accessed according to its index. Grover's algorithm [10,46] has the function that finds the index i_0 of a record (record_{i_0}) from the index superposition of state $\frac{1}{\sqrt{N}}(\sum_{i=0}^{N-1}|i\rangle)$. And record_{i_0} but rather than index i_0 is the genuine answer wanted by us. However, record_{i_0} cannot be measured out unless the one-to-one mapping relationship between index i and the corresponding record record_i is bound on the entangled state $\frac{1}{\sqrt{N}}(\sum_{i=0}^{N-1}|i\rangle|\mathrm{record}_i\rangle)$. That is, we need a unitaryoperation U_L such that

$$\frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle|0\rangle\right)|\mathrm{ancilla}_4\rangle \xrightarrow{U_L} \frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}|i\rangle|a_i\rangle\right)|\mathrm{ancilla}_3\rangle. \quad (13)$$

Using the same method shown in Figures 2 and 3, U_L can be designed. Figure 5 shows the design of the inverse unitary operation $(U_L)^\dagger$ for $N = 2$. Operation U_L has time complexity $O(\log_2 N)$ too.

The Grover iteration is defined as $G = (2|\xi\rangle\langle\xi| - 1)O_f$, where $|\xi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle$ and O_f denote the oracle that flips the phase of state in Grover iteration [39].

The general Grover iteration (GGI) is gived by $G' = (2|\xi\rangle\langle\xi|-1)(U_L)^\dagger(O_c)^\dagger O_f O_c U_L$, where O_c denotes another computation oracle [35–40]. In GGI, U_L is included, which can load the content of the record into the register to be entangled with its index. That is, U_L entangles index with its corresponding record so that the index and its corresponding record both can be measured out as a last answer.

5 The advantages of QLS for image compression and signal processing

The advantages of QLS are explained briefly below.

1) QLS is the bridge between electronic memory and the quantum computer, enabling the use of quantum computation to process a classical signal and image (see Figure 6).

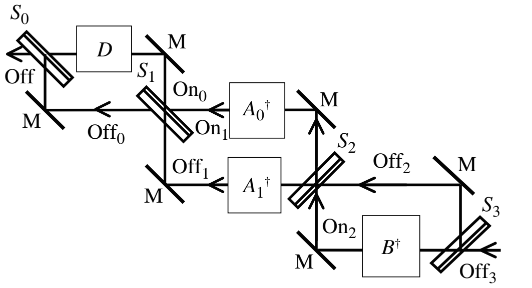

Figure 5 The illustration of unitary operation $(U_L)^\dagger$: $\frac{1}{\sqrt{2}}(\sum_{i=0}^{1}|i\rangle|a_i\rangle)|\mathrm{Off}_3\rangle \rightarrow \frac{1}{\sqrt{2}}(\sum_{i=0}^{1}|i\rangle|0\rangle)|\mathrm{Off}\rangle$. Operation U_L can be designed using the same method shown in Figures 3 and 5. S_0 : $|\mathrm{Off}_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|\mathrm{Off}\rangle + |\mathrm{On}\rangle)$, $|\mathrm{On}_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|\mathrm{Off}\rangle - |\mathrm{On}\rangle)$. Phase transformation $D = |i_1\rangle|0\rangle\langle 0|\langle i_1| - |i_0\rangle|0\rangle\langle 0|\langle i_0|$, where $i_0 = 0,\ i_1 = 1$.

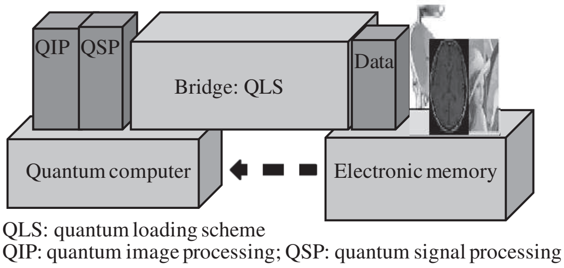

Figure 6 The role of QLS: QLS is the bridge between electronic memory and quantum computer, making possible the use of quantum computation to process an image and a classical signal.

2) QLS breaks through the efficiency bottleneck of loading data. $O(N)$ computation steps are needed to load data into the registers of the electronic CPU from the electronic memory. In contrast, if the QLS is adopted, only $O(\log_2 N)$ steps are needed for the quantum computer.

3) The advantages of the image data after loading the data information to the quantum entangled state: One advantage is that the information of huge data can be preserved in the entangled state. For example, for a given gray image with size $2^{n_1} \times 2^{n_2}$ (suppose that 8 bits save a datum), electronic computer costs space $2^{n_1} \times 2^{n_2} \times 8$ bits to save all information of this image, while quantum computer only costs $n_1 + n_2 + 8$ qubits if the entangled state is used. Another advantage is that quantum image processing holds high running speed, while electronic image processing seems to be too slow.

6 Conclusion

Two fundamental questions exist in the use of quantum computation to process an image or signal. They are: how to represent giant data such as image data using a quantum state without losing information, and how to load a huge amount of data into the quantum registers of a quantum CPU from electronic memory. Researches on these two questions are rare. This work was motivated by a desire to compress images and process classical signals using quantum computation. In this paper, an entangled state is used to represent an image (or vector) whose component is entangled with its subscript (or logical address). Using this proposal representation, few qubits can store a whole image without losing information. For example, $n_1 + n_2 + 8$ qubits can store the whole information on the gray image that has a $2^{n_1} \times 2^{n_2}$ size. An electronic computer does not have this ability.

A method of designing a unitary operation to load data, such as loading for a vector (or image) into the quantum registers of a quantum CPU from electronic memory, is advanced in this paper as a quantum loading scheme (QLS). In this paper, the QLS with time complexity $O(\log_2 N)$ is presented, while the classical loading scheme (CLS) has time complexity $O(N)$. QLS makes a quantum CPU compatible with electronic memory. Using QLS, all of the information for a digital image can be loaded into a quantum register at one time, a feat that is impossible with the electronic computer.

Acknowledgements

The work was supported by Aviation Science Foundation (Grant No. 2012ZD11) and Education Department of Sichuan Province (Grant Nos. 12ZA134, 09ZZ028). The authors would like to thank Ms. Kimberly Lawson at the Radiology Department of Brigham and Women's Hospital for her wonderful editing.

References

1 Castlman K R. Digital Image Processing (in Chinese). Beijing: Publishing House of Electronics Industry, 2002
2 Tang S F. Computer Organization and Architecture (in Chinese). Beijing: Higher Education Press, 2000
3 Feynman R. Simulating physics with computers. Int J Theor Phys, 1982, 21: 467–488
4 Deutsch D. Quantum theory: the Church-Turing principle and the universal quantum computer. Proc R Soc A-Math Phys Eng Sci, 1985, 400: 97–117
5 Deutsch D, Jozsa R. Rapid solution of problems by quantum computation. Proc R Soc A-Math Phys Eng Sci, 1992, 439: 553–558
6 Cleve R, Ekert A, Macchiavello C, et al. Quantum algorithms revisited. arXiv: quant-ph/9708016v1, 1997
7 Shor P W. Introduction to quantum algorithms. arXiv: quant-ph/0005003v2, 2000
8 Simon D R. On the power of quantum computation. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, 1994. 116–123
9 Shor P W. Algorithms for quantum computation discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on the Foundations of Computer Science, Santa Fe, 1994. 124–134
10 Grover L K. A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on the Theory of Computing. Philadelphia: ACM Press, 1996. 212–218
11 Coppersmith D. An approximate fourier transform useful in quantum factoring. arXiv:quant-ph/0201067v1, 2002
12 Griffiths R B, Niu C S. Semiclassical Fourier transform for quantum computation. arXiv:quant-ph/9511007v1, 1995
13 Kitaev A Y. Quantum measurements and the Abelian stabilizer problem. arXiv:quant-ph/9511026v1, 1995
14 Mosca M. Quantum computer algorithms. Dissertation for the Doctoral Degree. University of Oxford, 1999
15 Mosca M. Quantum algorithms. arXiv: quant-ph/0808.0369v1, 2008
16 Kaye P, Laflamme R, Mosca M. An Introduction to Quantum Computing. Oxford University Press, 2007
17 Yao A. Quantum circuit complexity. In: Proceedings of the 34th Annual Symposium on Foundations of Computer Science. Los Alamitos: IEEE Computer Society Press, 1993. 352–361
18 Bernstein E, Vazirani U. Quantum complexity theory. SIAM J Comput, 1997, 26: 1411–1473
19 Watrous J. Quantum computational complexity. In: Meyers R A, ed. Encyclopedia of Complexity and System Science. Berlin: Springer, 2009. 7174–7201
20 Vandersypen L M K, Steffen M, Breyta1 G, et al. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature, 2001, 414: 883–887
21 Lu C Y, Browne D E, Yang T, et al. Demonstration of Shor's quantum factoring algorithm using photonic qubits. Phys Rev Lett, 2007, 99: 250504
22 Einstein A, Podolsky B, Rosen N. Can quantum-mechanical description of physical reality be considered complete? Phys Rev, 1935, 47: 777–780
23 Bennett C H, Bernstein H J, Popescu S, et al. Concentrating partial entanglement by local operations. arXiv:quant-ph/9511030v1, 1995
24 Hill S, Wootters W K. Entanglement of a pair of quantum bits. arXiv:quant-ph/9703041v2, 1997
25 Cirac J I, Zoller P, Kimble H J, et al. Quantum state transfer and entanglement distribution among distant nodes in a quantum network. arXiv:quant-ph/9611017v1, 1996
26 Bennett C H, DiVincenzo D P, Smolin J A, et al. Mixed state entanglement and quantum error correction. arXiv:quant-ph/9604024v2, 1996
27 Zukowski M, Zeilinger A, Horne M A, et al. Event-ready-detectors Bell experiment via entanglement swapping. Phys Rev Lett, 1993, 71: 4287–4290
28 Vidal G, Latorre J I, Rico E, et al. Entanglement in quantum critical phenomena. arXiv:quant-ph/0211074v1, 2002
29 Braunstein S L, Fuchs C A, Gottesman D, et al. A quantum analog of Huffman coding. IEEE Trans Inf Theory, 2000, 46: 1644–1649
30 Devetak I, Winter A. Classical data compression with quantum side information. arXiv:quant-ph/0209029v4, 2003
31 Winter A. Compression of sources of probability distributions and density operators. arXiv:quant-ph/0208131v1, 2002
32 Hayden P, Jozsa R, Winter A. Trading quantum for classical resources in quantum data compression. arXiv:quant-ph/0204038v1, 2002
33 Ahn C, Doherty A C, Hayden P, et al. On the distributed compression of quantum information. IEEE Trans Inf Theory, 2006, 52: 4349–4357
34 Latorre J I. Image compression with entanglement. arXiv:quant-ph/0510031, 2005
35 Pang C Y. Quantum image compression. Postdoctoral Report. Hefei: Key Laboratory of Quantum Information, University of Science and Technology of China, 2006
36 Pang C Y, Zhou Z W, Chen P X, et al. Design of quantum VQ iteration and quantum VQ encoding algorithm taking $O(\sqrt{n})$ steps for data compression. Chin Phys B, 2006, 15: 618–623
37 Pang C Y, Zhou Z W, Guo G C. A hybrid quantum encoding algorithm of vector quantization for image compression.

Chin Phys B, 2006, 15: 3039–3043
38 Pang C Y, Zhou Z W, Guo G C. Quantum discrete cosine transform for image compression. arXiv:quant-ph/0601043, 2006
39 Pang C Y, Hu B Q. A quantum search algorithm of two entangled registers to realize quantum discrete Fourier transform of signal processing. Chin Phys B, 2008, 17: 3220–3226
40 Pang C Y, Ding C B, Hu B Q. Quantum pattern recognition of classical signal. http://arxiv.org/abs/0707.0936, 2007
41 Schutzhold R. Pattern recognition on a quantum computer. Phys Rev A, 2003, 67: 062311
42 Schaller G, Schutzhold R. Quantum algorithm for optical-template recognition with noise filtering. Phys Rev A, 2006, 74: 012303
43 Nielsen M A, Chuang I L. Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2002
44 Giovannetti V, Lloyd S, Maccone L. Quantum random access memory. Phys Rev Lett, 2008, 100: 160501
45 Giovannetti V, Lloyd S, Maccone L. Architectures for a quantum random access memory. arXiv:quant-ph/0807.4994v1, 2008
46 Galindo A, Martin-Delgado M A. Information and computation: Classical and quantum aspects. Rev Mod Phys, 2002, 74: 347–423
47 Arndt M, Nairz O, Vos-Andreae J, et al. Wave particle duality of C60 molecules. Nature, 1999, 401: 680–682
48 Hosten O, Rakher M T, Barreiro J T, et al. Counterfactual quantum computation through quantum interrogation. Nature, 2006, 439: 949–952
49 Yan W M, Wu W M. Data Structure (in Chinese). Beijing: Tsinghua University Press, 1992