A STUDY ON QAUANTUM FOURIER TRANSFORM AND IT'S APPLICATION IN REMOTE SENSING FOR IDENTIFICATION OF FEATURES

[1]N.R.SHANKER, [2] DR.S.S.RAMAKRISHANAN.
[1]Research Scholar, Anna University, India
[2]Professor in Remote sensing, Anna University

E-mail:nr_phd@yahoo.co.in

ABSTRACT

Facets of computer science, information theory, quantum mechanics combines to form Quantum computing. The repression architectural of Von Neumann computers with computational complexity of classical Algorithms which often deigh down, it makes to identity a better way to handle with skill for visual information. In classical computer the storage done in hardware is of bits. The bits are Independent to each other. The connectivity for those Independent bits is given by the software components. Inter connectivity between the bits in the memory leads to information to be lost. Each independent bit will represent some property of the associated image, like spatial, light strength. Retrieval of Image is done by the fetching the binary data from hardware if memory, by the way of independent property of bits. The reciprocal relationship between the bits of the image which is needed to understand properly about the Image is left out due to the Von Neumann architecture. An attempt has been made to improve visual information using QFT transform in the remote sensing data's for linear feature identification. Here the images are recorded using the Qubit Lattice and algorithms for an image in qubit have been done in serial methods.
Keywords: - QFT, Von Neumann architecture, transformations, qubit, Fourier transforms

1. INTRODUCTION

The most spectacular discover in quantum computing to date is that quantum computers can efficiently perform some tasks which are not feasible on a classical computer. For example, finding the prime factorization of n-bit integer is thought to require exp ((n1/3log2/3n)) operations using the best classical algorithm known at the time of writing, the so-called number field sieve(M.A. Nielsen and I.L. Chuang,2000).
This is exponential in the size of the number being factored, so factoring is generally considered to be an intractable problem on a classical computer: it quickly becomes impossible to factor even modest numbers. In contrast, a quantum algorithm can accomplish the same task using O(n^{2l}logn log log n)operations. (M.A. Nielsen and I.L. Chuang, 2000).That is, a quantum computer can factor a number exponentially faster than the best known classical algorithms. This result is important in its own right, but perhaps the most exciting aspect is the question it raises: what other problems can be done efficiently on quantum computers which are infeasible on a classical computer. In this paper we develop the quantum Fourier transform, which is the key ingredient for quantum factoring and many other interesting quantum algorithms. The quantum Fourier transforms is an efficient quantum algorithm for performing a Fourier transforms of classical data. But one important task which it does enable is phase estimation, the approximation of the eigen values of a unitary operator under certain circumstances. (M.A. Nielsen and I.L. Chuang,2000). This allows us to solve several other interesting problems, including the order-finding problem and the factoring problem. Phase estimation can also be combined with the quantum search algorithm to solve the problem of counting solutions to a search problems of how the quantum Fourier transform maybe used to solve the hidden subgroup problem, a generalization of the phase estimation and order-finding problems that has among its special cases an efficient quantum algorithm for the discrete logarithm problem, another problem thought to be intractable on a classical computer.

2. METHODOLOGY

An Alternative approach to avoid information in the Image which is carried out by Von Neumann architecture is quantum algorithm with associative memories (Feynman, 1982). By implementing quantum algorithms in the image processing leads for faster algorithms with reduced component size. The images are stored in Qubit with non-entangled array of qubits. In Quantum algorithms the pixels which are represented by the correct values of the frequency values rather than linear combination of RGB which is followed in the Von Neumann computers. A System, which is capable of detecting EM, waves for different frequencies. When an EM wave has an impact on the system it generates portionate qubits. The system works as an infective function. A: F → Ψ where F is the set of monochromatic electromagnetic waves whose frequencies can be detected by A and Ψ is the set of quantum states of the form ()

$\Psi\rangle = \cos\theta/2\ |0\rangle + \sin\theta/2\ |1\rangle$,

where $\theta/2 \in [0, \pi/2]$ ----------- (1)

$\Psi\rangle = \cos\theta/2\ [{}_0{}^1] + \sin\theta/2\ [{}^0{}_1]$, --------(2)

For each frequency value of EM wave, it is possible to find a value of theta in Equation 2. The system can generate the different states of qubits for different frequency wave. The system consists of frequency detector and storeded, and for which a magnetic field equal to the stored EM waves to spin up or spin down state. By this way quantum state is generated with real parameter Ø equal to the stored frequency (Simon, 1997).

Recording an Image in a Qubit Lattice

$Ø = \{|h\rangle_{ij}\}\ \ i = \in \{1,2,\ldots.n_1\}$; $j \in \{1,2,\ldots n_2\}$

Ø is a lattice of qubits,

Ø = 2 dimensional qubit array

$M = \{Ø_k\}$, $k \in \{1,2,\ldots.n_3\}$

M = 3D of qubit lattice

$M = \{|h\rangle_{ijk}\}$ is of $n_1 * n_2 * n_3$ qubits

Algorithm for an Image in qubit

1. Set i=0 and j=0
2. The frequency u_{ij} for which EM impact on System will generates qubit |h>ij $K \in \{1,2,\ldots n\}$
3. Update ij values for visual interpretation which is in system.

Preceded the algorithm for all frequency which has impact on the system.

This is of serial methods, and it can be done for parallel methods also.

The image stored in M

$P = \alpha_1|o\rangle\langle o| + \alpha_2|1\rangle\langle 1|$

$P(\alpha_1) = \cos^2 Ø/2 \quad p(\alpha_2) = \sin^2 Ø/2$

Ø= color information in every qubit in M

3. HIVING IMAGES IN ENTANGLED QUANTUM SYSTEMS.

Entangled Quantum System which measure and manipulate the systems as a whole, rather than independent basis(Feynman,1982). Entangled quantum plays a vital role in quantum Computing and Ø/p for building Algorithms.

Entangle states

$$|\Psi\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

$$|GHZ\rangle = \frac{|000\rangle - |111\rangle}{\sqrt{2}}$$

One of the most useful ways of solving a problem in mathematics (or) computer science is to transform it into some other problem for which a solution is known. There are a few transformations of this type that appear so often and in so many different contexts that the transformations are studied for their own sake. A great discovery of quantum computation has been that some such transformations can be computed much faster on a quantum computer than on a classical computer, a discovery which has enabled the construction of fast algorithms for quantum computers (Feynman,1982).One such transformation is the discrete Fourier transform. In the usual mathematical notation, the discrete Fourier transform takes a input vector of complex numbers $x_0,\ldots.x_{N-1}$ where the length N of the vector is a fixed parameter. It outputs the transformed data, a vector of complex numbers $y_0\ldots\ldots y_{N-1}$ defined by

$$Y_K = 1/\sqrt{N} \sum_{j=0}^{N-1} x_j\, e^{2\Pi j\, ik/N}$$

The quantum Fourier transform is exactly the same transformation, although the conventional notation for the quantum Fourier transform is somewhat different. The quantum Fourier transform on orthonormal basis |0> ….|N-1> is defined to be a linear operator with following action on the basis states (shor, 1994),

$$|\,j\,\rangle = 1/\sqrt{N} \sum_{jk=0}^{N-1} x_j\, e^{2\Pi j\, ik/N}\ \ |K\rangle$$

Equivalently, the action on an arbitrary state may be written

$$\sum_{j=0}^{N-1} x_j\,|J\rangle \longrightarrow \sum_{k=0}^{N-1} y_k\,|K\rangle$$

Where the amplitudes y_k are the discrete Fourier transform of the amplitudes x_j, it not obvious from the definition, but this transformation is a unitary transformation ,and thus can be implemented as the dynamics for a quantum computer. We shall demonstrate the unitarily of the Fourier Transform by a constructing a manifestly unitary quantum circuit computing the Fourier transform. It is also easy to prove directly that the Fourier transform in unitary. In the following we taken $=2^n$, where n is some integer, and the basis |0>….|2^n-1> is the computational basis for an n qubit quantum computer(shor,1994). It is helpful to write the state |j> using the binary representation j = j_1, j_2, ….j_n. More formally, j = $j_1 2^{n-1} + j_2 2^{n-2} + \ldots j_n 2^0$. It is also convenient to adopt the notation O.j_1, j_{1+1} …j_m to represent the binary fraction $j_{1/2} + j_{s+1/4} + \ldots\ldots j_m / 2^{m-l+1}$. With a little algebra the quantum Fourier transform can be given the following useful product representation (simon,1997).

$$|j_1 \ldots j_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi io.j_n}|1\rangle)\ (|0\rangle + e^{2\pi io.j_{n-1}j_n}|1\rangle) \ldots\ldots (|0\rangle + e^{2\pi io.j_1 j_2 \ldots j_n}|1\rangle)}{2^{n/2}}$$

This product representation is so useful that you may even wish to consider this to be the definition of the QFT. As we explain shortly this representation allows us to construct an efficient quantum circuit computing the Fourier transform, a proof that the quantum Fourier transform is unitary, and provides insight into algorithms based upon the quantum Fourier Transform (shor,1994). As an incidental bonus we obtain the classical fast Fourier Transform, The equivalence of the product representation (1) and the definition 2 follows from some elementary algebra

$$|j\rangle = 1/2^{n/2} \sum_{k=0}^{2n-1} e^{2\pi jik\, |2n}\, |k\rangle$$

and equal to

$$\frac{(|0\rangle + e^{2\pi io.j_n}|1\rangle)\ (|0\rangle + e^{2\pi io.j_{n-1}j_n}|1\rangle) \ldots\ldots (|0\rangle + e^{2\pi io.j_1 j_2 \ldots j_n}|1\rangle)}{2^{n/2}}$$

The product representation (1) makes it easy to derive an efficient circuit for the quantum Fourier transforms. Such a circuit is 2 the gate R_k denotes the unitary transformation

$$R_k \equiv \begin{vmatrix} 1 & 0 \\ 0 & e^{2\pi i/2\, k} \end{vmatrix}$$

To see that the pictured Circuit computer the quantum Fourier transform, consider what happens when the state |j_1….j_n> is input. Applying the Hadamard gate to the first bit produces the state

$$1/2^{1/2}\ (|o\rangle + e^{2\pi i0.j1}\ |1\rangle) / \ j_2 \ldots . j_n\rangle ,$$

Since $e^{2\pi i0.j1}$ = -1 when $j_1 = 1$ and is +1 otherwise, applying the controlled –R_2 Rate produces the state.

$$1/ 2^{1/2}\ (|0\rangle + e^{2\pi io.j_1 j_2 \ldots j_n}|1\rangle)\ \ |j_2 \ldots\ldots j_n\rangle$$

We continue applying the controlled –R_3 , R_4 thought R_n gates, each of which adds an extra bit to the phase of the co-efficient of the first |1>,1/ $2^{1/2}$ (|0> + $e^{2\pi io.j_1 j_2 \ldots j_n}$|1>) |$j_2$......$j_n$>

4. SHOR'S ALGORITHM

Shor's algorithm for factoring a given integer n can be broken into some simple steps.

1. Determine if the number n is a prime, a even number, or an integer power of a prime number. If it is we will not use Shor's algorithm. There are efficient classical methods for determining if a integer n belongs to one of the above groups, and providing factors for it if it does. This step would be performed on a classical computer.
2. Pick a integer q that is the power of 2 such that $n^2 \le q < 2n^2$. This step would be done on a classical computer (Feynman,1982).
3. Pick a random integer x that is co prime to n. When two numbers are co prime it means that their greatest common divisor is 1. There are efficient classical methods for picking such an x. This step would be done on a classical computer.
4. Create a quantum register and partition it into two sets, register one and register two. Thus the state of our quantum computer can be given by: $left|\ \rangle$ reg1, reg2 . Register one must have enough qubits to represent integers as large as $q-1$. Register two must have enough qubits to represent integers as large as $n-1$.
5. Load register one with an equally weighted superposition of all integers from 0 to $q-1$. Load register two with the 0 state. Our quantum computer would perform this operation. The total state of the quantum memory register at this point is:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, 0\rangle$$

6. Apply the transformation $x^a \bmod n$ to for each number stored in register one and store the result in register two. Due to quantum parallelism this will take only one step, as the quantum computer will only calculate $x^{|a\rangle} \bmod n$, where $|a\rangle$ is the superposition of states created in step 5.

This step is performed on the quantum computer. The state of the quantum memory register at this point is:

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a, x^a \bmod n\rangle$$

7. Measure the second register, and observe some value k. This has the side effect of collapsing register one into a equal superposition of each value a between 0 and $q-1$ such that

$$x^a \bmod n = k$$

8. The quantum computer performs this operation. The state of the quantum memory register after this step is: (shor,1994),

$$\frac{1}{\sqrt{||A||}}\sum_{a'=a'\in A}|a', k\rangle$$

Where A is the set of a's such that $x^a \bmod n = k$, and $||A||$ is the number of elements in that set. Compute the discrete Fourier transform on register one. The discrete Fourier transform when applied to a state $|a\rangle$ changes it in the following manner:

$$a\rangle = \frac{1}{\sqrt{q}}\sum_{c=0}^{q-1}|c\rangle * e^{2\pi iac/q}$$

This step is performed by the quantum computer in one step through quantum parallelism. After the discrete Fourier transform our register is in the state:

$$\frac{1}{\sqrt{||A||}}\sum_{a'\in A}\frac{1}{\sqrt{q}}\sum_{c=0}^{q-1}|c, k\rangle * e^{2\pi ia'c/q}$$

9. Measure the state of register one, call this value m, this integer m has a very high probability of being a multiple of q/r, where r is the desired period. The quantum computer performs this step.

10. Take the value m, and on a classical computer do some post processing which calculates r based on knowledge of m and q. There are many ways to do this post processing, they are complex are are omitted for clarity in presentation of the quantum core of Shor's Algorithm. This post processing is done on a classical computer.

11. Once you have attained r, a factor of n can be determined by taking $\gcd(x^{r/2}+1, n)$ and $\gcd(x^{r/2}-1, n)$. If you have found a factor of n, then stop, if not go to step 4. This final step is done on a classical computer (shor,1994)

Step 11 contains a provision for Shor's algorithm failing to produce the factors of n. Shor's algorithm can fail for multiple reasons, for example the discrete Fourier transform could be measured to be 0 in step 9, making the post processing in step 10 impossible. At other times the algorithm will sometimes find factors of 1 and n, which is correct but not useful.

5. CONCLUSION

The above equations can be programmed using Matlab to convert the bit to qubit and QFT functions for images are been applied. Still the study is been continued for the application of the qft transforms for the remote sensing sensors for various feature identification.

REFERENCE:

[1] A. Ekert and R. Jozsa, Quantum computation and Shor’s factoring algorithm, Reviews of Modern Physics 68 (1996) 733-753.

[2] Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, Proc. R. Soc. London A439 (1992) 553-558.

[3] D. Simon, On the power of quantum computation, Proc. 35th Annual Symposium on Foundations of Computer Science (1994) 116 and SIAM Journal on Computing 26 (1997) 1474-14

[4] A. Nielsen and I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge (2000).

[5] R.P. Feynman, Simulating Physics with computers, Int. J. Theory. Phys. 21 (1982)467-488.