Quantum Hough Transform

Karthik Mahesh Varadarajan
Technical University of Vienna
kv@acin.tuwien.ac.at

Abstract

With the rapid growth of quantum computing, several quantum algorithms are being designed for a variety of applications. However, the application of quantum algorithms to image processing and computer vision has been rather restricted. The Quantum Fourier Transform (QFT) is one such algorithm that is used in applications such as discrete logarithms and period finding that posit employment in image processing, but cannot be used for traditional applications such as spectrum estimation and filtering. In this paper, we present a novel quantum version of the popular Hough Transform and show the advantages of the Quantum Hough Transform in terms of the computational and space requirements over the traditional formulation. Specifically, we transform accumulation in Hough space to implicit quantum superposition resulting in gains in space and speed requirements.

***Keywords: Quantum algorithms, Hough Transform, Quantum superposition, Quantum computing**.*

1. INTRODUCTION

Quantum computers offer the possibility of enhanced computing speeds for several types of computational problems that are inherently hard to solve on traditional computers. With recent developments in the field of theoretical quantum computing, efforts have also been made to identify algorithms designed for conventional computing that are particularly well-suited for implementation on quantum computers. Some popular conventional algorithms that have found quantum computing implementations include prime factorization (Shor's algorithm [1]), index search (Grover's algorithm [2]), periodicity detection in modulo arithmetic (Simon's algorithm [3]) and Quantum Fourier Transform [4]. These algorithms have been largely restricted to the domain of traditional artificial intelligence with few applications in the domain of image processing and computer vision. Quantum Fourier Transform (QFT) for example is used in the for Period Finding, Discrete Logarithms, Order Finding, Quantum Counting, Hidden Sub-Group problems, Encryption Decoding, but is not applicable to traditional fields of application of Discrete Fourier Transform (DFT) such as Spectrum Analysis, Filtering, Compression etc. This is because, quantum algorithms do not work directly in the space of the input vectors but on quantum states and the resulting outputs from the application of quantum algorithms are also quantum states with the only observable or measurable quantity being the state of the system and not its amplitude. For example, in the case of QFT, the measurable quantity is the quantum state in the encoded quantum fourier basis, while the amplitude of the frequency component can only be measured as the squared probability of occurrence of this state. Thus, it can be seen that it is rather difficult to build quantum versions of conventional algorithms and even in cases this is possible, the output is in a state that cannot be directly used in the traditional application of interest. However, it has been observed that quantum algorithms (in cases where the output can be directly used in the application of interest) provide orders of speed-up that are not possible with conventional algorithms. The most popular quantum algorithms operate in polynomial time whereas the corresponding conventional algorithms operate in exponential time, thereby resulting in an exponential order of speed-up. Much of the speed-up in the case of quantum algorithms is attributed to prior knowledge about the structure of the problem being used [10]. The accumulation of histories in the case of a classical problem to as much as 50% of the information required to solve a problem serves as a significant factor in the quantum speed-up.

Two specific sub-fields of Quantum Computing are also relevant from the standpoint of computer vision. The first of these is Quantum Signal Processing (QSP) and the second is composed of quantum systems for optimization of object recognition problems. Besides these two sub-fields, there have been a few algorithms that draw inspiration from quantum state modeling towards solving conventional image processing and computer vision problems. Notable among these include the use of quantum interference analogies in mutation for genetic programming for image registration [5], use of qubit like structures called Conjugate Information Variables that have non-isotropic sensitivity regions employable for classification of rising, falling and flat regions of functions (or images) and hence be used for edge-sensitive adaptive filtering [6], quantum representations for 4-color channel images (RGB-A) [7], quantum representations for joint color image encoding and image structure description for recognition [8] and the use of quantized states (though not quantum states) for Total Variation (TV) in a Mumford-Shah like energy formulation for image segmentation and denoising [9].

The field of QSP [11] is aimed at developing new or modifying existing algorithms by applying some of the principles of quantum mechanics. Hence these are not true quantum algorithms that can be implemented on a quantum computer, but rather use principles that are borrowed from quantum literature. Hence, the algorithms developed using QSP are not limited by the constraints of quantum mechanics. Some of the original QSP applications include frame theory, quantization and sampling methods, compressed sensing matched filters, subspace coding, detection, parameter estimation, covariance shaping and multi-user wireless communication systems. These QSP algorithms use a Quantum Input Mapping followed by the application of either Rank-One Measurements (ROM) or Subspace Measurements, again followed by an Output Mapping in order to achieve various required tasks. QSPs have been used in image processing, recently, for a number of tasks such as image half-toning, edge detection, cryptography - by mapping binary images into the $|0>$ and $|1>$ states of the quantum system and estimating the probabilities of these two states as $|p_0|^2$ and $|p_1|^2$ using an exponential function of the variables measured from the image intensity values [12]. Such a formulation has been used for structural element description for noise filtering in [13]. This has also been used in relation to neighborhood pixel intensities for defining features, corners and subsequently edges in [14].

The second sub-field includes quantum algorithms for optimization of object recognition problems. Primary among these algorithms for image feature matching include those formulated as Quadratic Unconstrained Binary Optimization (QUBO) and solved using D-Wave Superconducting Adiabatic Quantum Computer (ADC) with the theory developed in [15,16,17] and hardware implementations demonstrated in [18,19]. Another "true" quantum algorithm that is focused on object recognition and de-noising that works on quantum states at all stages of the processing is [20].

2. ALGORITHM

In this paper, we further the scope of application of quantum algorithms to image processing and computer vision by creating a quantum version of the Hough Transform [22], the conventional version of which has been used for numerous applications such as edge detection/selection and object template matching. It should be noted here that the algorithm proposed in this paper – Quantum Hough Transform or QHT is a "true quantum algorithm" that can be implemented on quantum computers as opposed to QSP which subsumes quantum inspired algorithms. Furthermore, using quantum complexity analysis, we demonstrate that time and space requirements of QHT are several orders of magnitude smaller than that required for traditional Hough transform (HT) implementations, thereby demonstrating the need and benefits of QHT over traditional HT.

2.1 Hough Transform (HT)

A popular algorithm used in image processing for detection of features such as edges, curves, circles, contours, planes, cylinders and even entire shapes is the Hough Transform. The Generalized Hough Transform (GHT) [23] in particular is extremely useful in template matching of arbitrary objects described using models. In the case of GHT, the problem is transformed to one of determination of spatial coordinates of the shape template being matched in the image of interest. An edge based implementation of GHT was first proposed by measuring the distance from points in the shape. While HT can be used for a variety of purposes, with the classical HT being used for analytic functions, the simplest variant of the HT which is used for the detection of edges in images remains the most popular and widely used application of HT. All variants of HT use a voting procedure in parameter space called the accumulator space from which object candidate are localized by extraction of local maxima. Since an accumulator has to be maintained across the entire parameter space of the function, HT typically takes a great amount buffer space for implementation. For example, for the task of finding edges in an image, wherein the HT uses the parametrization (r, θ), the space complexity of the HT is given by $O(n_r.n_\theta)$, where n_r is the number of bins along the r parameter space and n_θ is the number of bins along the θ parameter space. Given that the range of r spans the space from 0 to $\sqrt{l^2 + w^2}$, where l and w are the image dimensions and θ spans the entire range of angles from 0 to 360, it can be seen that even for a simple 2 parameter accumulator space, the computational requirements can be quite high depending on the necessary resolution. While variants of Hough Transform exist that attempt at limiting the computational complexity by reducing the size of the accumulator search space or by using additional information from the image space such as image gradients to coarsely determine the HT parameters, we focus our attention in this paper on the simplest form of HT operating in the full accumulator space since most such optimization modifications are also applicable to the HT estimation method presented in this paper.

2.2 Quantum Representation of Hough Transform Space

The first transformation required to build the quantum algorithm is the transformation of the input pixel co-ordinate space $p(x, y)$ into the one dimensional complex vector space $\mathbb{C}^1$, also called the z-plane. Hence,

$$p(x, y) = (p_x, p_y) \Rightarrow p_x + p_y i$$

In order to enable the QHT to maintain the same ranges as that of HT, we use a discretized and bounded version of the complex space restricted in range to that of the original image space. Hence,

$$p_x \in \{1 \dots l\}, \qquad p_y \in \{1 \dots w\}$$

Using the parametrization (r, θ), where r represents the shortest distance from the origin to the line and θ represents the angle that the vector from the origin to the line (at its closest point) makes with the first coordinate axis, for the Hough lines to be determined, the original formulation for the Hough lines

$$y = \left(-\frac{cos\theta}{sin\theta}\right)x + \left(\frac{r}{sin\theta}\right)$$

can be rewritten as

$$r(\theta) = xcos\theta + \ ysin\theta$$

For an arbitrary point (x_0, y_0) on the line, the value of $r(\theta)$ is given by

$$r(\theta) = x_0cos\theta + \ y_0sin\theta$$

Hence, the value of $r(\theta)$ from arbitrary pixels in the image containing the line can be expected to be roughly constant for a range of θ values - θ_k, typically from 0 degrees to 180 degrees, where $k \in \{\, k_{min} = 0, \dots, k_{max}\}$. Hence, the required values of r and θ corresponding to the parameterization of a single Hough line can be determined as

$$\max_k r(\theta)$$

The analogous equations in the complex plane representation are given by

$$\begin{aligned} r(\theta) &= x_0cos\theta + \ y_0sin\theta \\ &= x_0cos\theta - i^2y_0sin\theta \\ &= \ x_0.cos\theta - (i.y_0).(i.sin\theta) \\ &= p_{x0}.cos\theta - p_{y0}.(i.sin\theta) \\ &= p_{x0}.cos(-\theta) + p_{y0}.(i.sin(-\theta)) \\ &= \ p_0.e^{-i\theta} \end{aligned}$$

Representing the above equation using a dot product and introducing the matrix notation, along with the range subscript k, we have

$$r(\theta_k) = [p_{x0}\ p_{y0}]\begin{bmatrix} cos(-\theta_k) \\ sin(-\theta_k) \end{bmatrix}$$

where $\begin{bmatrix} cos(-\theta_k) \\ sin(-\theta_k) \end{bmatrix}$ corresponds to $e^{-i\theta_k}$.

Since θ_k typically ranges from 0 degrees to 180 degrees (or 0 to π) in incremental steps based on k_{max}, we have

$$\theta_k = \frac{2\pi k}{2k_{max}}$$

Hence,

$$r(\theta_k) = p_0.e^{-i\frac{2\pi k}{2k_{max}}}$$
$$= p_0.\omega^{-\frac{k}{2}}$$

where,

$$\omega = e^{\frac{i2\pi}{k_{max}}}$$

is the k_{max} th root of unity.

Given a binary edge image of size $l.w$ pixels with values $I_{x0,y0}$ at the pixel $p_0(x_0, y_0)$, the pixel indices corresponding to an image intensity value of 1 can be linearized into the following complex vector

$$P = \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ \dots \\ p_N \end{bmatrix}$$

where $I_{p_m} = 1, m \in \{1, \dots, N\}$.

A conventional HT requires an accumulator space spanning the entire range of θ and r in order to compute the Hough Transform. However, with the QHT, we simulate the accumulation process for finding the peak in the (r, θ) space through quantum superposition. In order to describe the problem in a format suitable for quantum superposition, we break down the accumulator into two sub-matrix representations.

The first sub-matrix representation (QHT1) involves the estimation of accumulated r for various values of θ.

$$\begin{bmatrix} p_0 \\ p_1 \\ \dots \\ \dots \\ p_N \end{bmatrix} . \begin{bmatrix} 1 & 1 & \dots & \dots & 1 \\ \omega^{-\frac{1}{2}} & \omega^{-\frac{1}{2}} & \dots & \dots & \omega^{-\frac{1}{2}} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \omega^{-\frac{k_{max}}{2}} & \omega^{-\frac{k_{max}}{2}} & \dots & \dots & \omega^{-\frac{k_{max}}{2}} \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \\ \dots \\ \dots \\ r_{k_{max}} \end{bmatrix}$$

Simultaneously, the following outer-product (QHT2) or projection matrix is also computed using the partial product terms from the above matrix computation.

$$\begin{bmatrix} p_0 \\ p_1 \\ \dots \\ \dots \\ p_N \end{bmatrix} . \begin{bmatrix} 1 \\ \omega^{-\frac{1}{2}} \\ \dots \\ \dots \\ \omega^{-\frac{k_{max}}{2}} \end{bmatrix}^T = \begin{bmatrix} r_{00} & r_{10} & \dots & \dots & r_{N0} \\ r_{01} & r_{11} & \dots & \dots & r_{N1} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ r_{0k_{max}} & r_{1k_{max}} & \dots & \dots & r_{Nk_{max}} \end{bmatrix}$$

Now, representing the input complex pixel space vector p_m in the equivalent quantum superposition state, we have

$$\sum_{i=0}^{N} p_i |i\rangle$$

where $|i\rangle = |0\rangle \dots |N\rangle$ are orthonormal basis vectors. Using the above representation, the first sub-matrix representation yields

$$|i\rangle \to \sum_{k=0}^{k_{max}} r_k |k\rangle$$

which is a quantum superposition of the basis vectors in $|k\rangle$. It should however be noted that the output of QHT1 cannot be measured directly in the space of r_k. In other words, the magnitude of r_k cannot be calculated due to the nature of quantum states. The only output that can be measured is the state $|k\rangle$ itself (index) and the probability of measurement of this state $|k\rangle$ is given by $|r_k|^2$. For the case of the HT, this is not a disadvantage, but rather a useful phenomenon. While, the use of QFT is restricted in the case of measurement of spectra due to the inability to measure the magnitude of frequency components as a result of this phenomenon, this constraint is not a drawback in the case of QHT, since the goal of QHT is the estimation of $|k\rangle$ or k and r that maximize r_k. Since the values of k that maximize r_k are observed during quantum measurement with a very high probability, this enables the determination of the Hough peak. Furthermore, by performing several measurements, multiple (possible) candidate Hough peaks can be estimated. This also enables the estimation of multiple line segments in an image through the estimation of several Hough peaks. Thus, quantum superposition inherently acts as an accumulator, preserving the quantum states, without the need for an explicit one that demands high storage and computation requirements.

The second transformation QHT2 transforms the input quantum state as follows

$$|i\rangle \to \begin{bmatrix} \sum_{j=0}^{N} r_{j0} |j0\rangle \\ \sum_{j=0}^{N} r_{j1} |j1\rangle \\ \dots \\ \dots \\ \sum_{j=0}^{N} r_{jk_{max}} |jk_{max}\rangle \end{bmatrix}$$

Once the candidate values of k, corresponding to the Hough peaks are determined from QHT1, the appropriate quantum state vectors can be selected using the measured $|k\rangle$ states. The selected superposition state(s) vector,

$$\sum_{j=0}^{N} r_{jk_{sel}} |jk_{sel}\rangle$$

is then used as input to the Quantum Counting algorithm [21]. Parameters for discretizing the r space can also be directly incorporated into the counting algorithm. This algorithm counts the instances of each selected $r_{jk_{sel}}$ up to a threshold and also returns the indices of j, thus not just enabling the detection of r corresponding to the Hough peaks, but also the pixels in the image that correspond to the Hough line. Thus, detection of r, θ as well pixel indices corresponding to the Hough Transform are computed using the Quantum formulation.

2.3 Complexity Analysis

It should be noted that the Quantum Counting algorithm uses only $\Theta\left(\frac{1}{\epsilon}\sqrt{\frac{N}{t}}\right)$, where t is the number of selected r_j in the list.

Furthermore, it should also be noted that in the computation of QHT1 and QHT2, each basis state index can be represented in the binary form (here, N is assumed to be a power of 2 - n for the sake of simplicity, though an approximation can be used for any value of N)

$$|i\rangle = |i_1, i_2, \dots, i_n\rangle$$
$$= |i_1\rangle \otimes |i_2\rangle \otimes \dots \otimes |i_n\rangle$$

where,

$$i = i_1 2^{n-1} + i_2 2^{n-2} + \dots + i_n 2^0$$

Since the product terms in QHT1 are powers of ω, the computation of intermediate terms can be chained similar to that with QFT. Operations on $n-$qubits can be factored into tensor products of n single qubit operations, which can be implemented using Hadamard gates (H) and Controlled Phase Gates (R_θ). Figure 1 demonstrates the QHT circuit, along with the QHT1 and

QHT2 sub-matrix components and their tap points in blue and red respectively. Similar to QFT, it can be seen that the number of gates required for the implementation is given by

$$\frac{n(n+1)}{2} = O(n^2)$$

(since each factor requires one H gate and one R_θ gate more than the previous factor), which is polynomial in the number of qubits. On the other hand, HT typically requires at least $O(N^2)$ computations, which is an exponential order more than that required for QHT. Furthermore, as discussed earlier, QHT performs accumulation implicitly through Quantum Superposition as opposed to HT that requires a space complexity of $O(n_r.n_\theta)$, depending on the granularity of the parametrization. The only space requirement in the Quantum Hough domain computation comes from the space used by the Quantum Counting method to estimate most probable values of r. However, since θ has already been chosen at this stage of computation, the space requirements for the Quantum Counting algorithm is linear in k_{max} $\big(O(k_{max})\big)$.

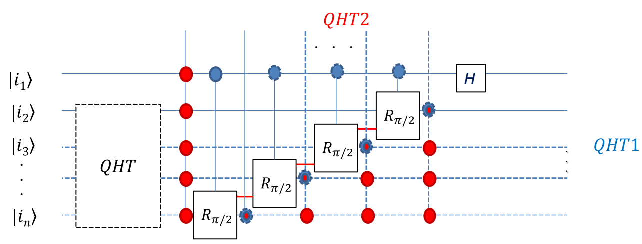

Figure 1. QHT Circuit Implementation

3. CONCLUSION

In this paper, we have presented a novel quantum version of the popular Hough Transform and shown the advantages of the Quantum Hough Transform in terms of computational and space requirements over the traditional formulation. A circuit has also been designed for the implementation of the algorithm. The transformation of the accumulation in Hough space into an implicit quantum superposition results in gains in space and speed requirements. It should however be noted that the determination of Hough peaks using QHT has been done through a two-step process that determines peaks in the dependent θ space, before re-estimating peaks in the r space. Future work will involve reformulation of the problem to estimate peaks in the joint r, θ space and comparison of performance and trade-offs between the two approaches. In addition, further work is planned towards the goal of implementing and evaluating the algorithm on a quantum computer.

4. REFERENCES

[1] Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM journal on computing 26.5 (1997): 1484-1509.

[2] Grover L.K.: A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212.

[3] Simon, D.R. (1994), "On the power of quantum computation", Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on: 116–123

[4] L. Hales , S. Hallgren, An improved quantum Fourier transform algorithm and applications, Proceedings of the 41st Annual Symposium on Foundations of Computer Science, p.515, November 12–14, 2000.

[5] Talbi, H., Draa, A., & Batouche, M. C. (2004, April). A genetic quantum algorithm for image registration. In Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on (pp. 395-396). IEEE.

[6] Nölle, M., & Suda, M. (2011). Conjugate Variables as a Resource in Signal and Image Processing. arXiv preprint arXiv:1108.5720.

[7] Sun, B., Le, P. Q., Iliyasu, A. M., Yan, F., Garcia, J. A., Dong, F., & Hirota, K. (2011, September). A multi-channel representation for images on quantum computers using the RGBα color space. In Intelligent Signal Processing (WISP), 2011 IEEE 7th International Symposium on (pp. 1-6). IEEE.

[8] Venegas-Andraca, S. E. (2005). Discrete quantum walks and quantum image processing (Doctoral dissertation, Univ Oxford).

[9] Shen, J., & Kang, S. H. (2007). Quantum TV and applications in image processing. Inverse Problems and Imaging, 1(3), 557.

[10] Castagnoli, G. (2009). Discussing the explanation of the quantum speed up. arXiv preprint arXiv:0910.2313.

[11] Eldar, Y. C., & Oppenheim, A. V. (2002). Quantum signal processing. Signal Processing Magazine, IEEE, 19(6), 12-32.

[12] Tseng, C. C., & Hwang, T. M. (2003). Quantum digital image processing algorithms. In 16th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP 2003).

[13] Zhou, C., Hu, Z., Wang, F., Fan, H., & Shang, L. (2010). Quantum Collapsing Median Filter. In Advanced Intelligent Computing Theories and Applications (pp. 454-461). Springer Berlin Heidelberg.

[14] Liang Chen, Xu-ming Ye, Li Yu, Bo-xia Xu, Xin-min Shi (2010). A novel feature detector based on quantum-inspired method. In International Conference on Machine Vision (ICMV).

[15] Neven, H., Rose, G., & Macready, W. G. (2008). Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization. arXiv preprint arXiv:0804.4457.

[16] Neven, H., Denchev, V. S., Rose, G., & Macready, W. G. (2008). Training a binary classifier with the quantum adiabatic algorithm. arXiv preprint arXiv:0811.0416.

[17] Neven, H., Denchev, V. S., Rose, G., & Macready, W. G. (2009). Training a large scale classifier with the quantum adiabatic algorithm. arXiv preprint arXiv:0912.0779.

[18] Neven, H., Denchev, V. S., Drew-Brook, M., Zhang, J., Macready, W. G., & Rose, G. (2009). NIPS 2009 demonstration: Binary classification using hardware implementation of quantum annealing. Quantum, 1-17.

[19] Denchev, V. S., Ding, N., Vishwanathan, S. V. N., & Neven, H. (2012). Robust Classification with Adiabatic Quantum Optimization. arXiv preprint arXiv:1205.1148.

[20] Schaller, G., & Schützhold, R. (2006). Quantum algorithm for optical-template recognition with noise filtering. Physical Review A, 74(1), 012303.

[21] Brassard, Gilles, Peter Høyer, and Alain Tapp. "Quantum counting." Automata, Languages and Programming. Springer Berlin Heidelberg, 1998. 820-831.

[22] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15 (January, 1972)

[23] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, p.111-122, 1981