Quantum Radon Transform and Its Application

Guangsheng Ma, Hongbo Li, and Jiman Zhao,

Abstract—This paper extends the Radon transform, a classical image processing tool for fast tomography and denoising, to the quantum computing platform. A new kind of periodic discrete Radon transform (PDRT), called quantum Radon transform (QRT), is proposed. The QRT has a quantum implementation that is exponentially faster than the classical Radon transform. Based on the QRT, we design an efficient quantum image denoising algorithm. The simulation results show that QRT preserves the good denoising capability as in the classical PDRT. Also, a quantum algorithm for interpolation-based discrete Radon transform (IDRT) is proposed, which can be used for fast line detection. Both the quantum extension of IDRT and the line detection algorithm can provide polynomial speedups over the classical counterparts.

Index Terms—Radon transform, quantum computation.

I. INTRODUCTION

RADON transform, proposed by Johann Radon in 1917 [1], is an important image processing tool with widespread applications in computed tomography, geophysics, and remote sensing, etc. [2]. It changes a function \( f \) defined on the plane to a function \( R_\theta f(\rho) \) defined on the space of lines in the plane, whose value on the line with interception \( \rho \) and slope \( \theta \) equals the integral of function \( f \) along the line:

\[
R_\theta f(\rho) := \int_{\mathbb{R}^2} f(x, y) \delta(\rho - x \sin \theta + y \cos \theta) dx dy,
\]

where \( \theta \in (0, \pi], \rho \in \mathbb{R}, \) and \( \delta \) is the Dirac function. By definition, Radon transform possesses the capability of detecting singularities along straight lines, and performs better at denoising images with linear singularities than other image processing tools [2], [3].

To implement Radon transform, discretization is necessary. However, the different discretization methods will result in different discrete Radon transforms (DRTs) that have different applications. In discretization, there are two methods to approximate the line integral in (1): the interpolation method, and the periodic discrete grid method.

The interpolation method (IDRT) evaluates the integral along a straight line by making interpolation among the adjacent points on the discrete image grid of the line. The earlier DRTs are based on this method [1]. In 1987, Beylkin discovered an exact inversion formula, and proved that if the discrete version is based on Radon’s original formula, then the reconstruction can only be approximate [4]. The IDRT can be used in line detection, X-ray computed tomography, etc. Performing the IDRT on an \( N \times N \) image often requires at least \( \Omega(N^3) \) arithmetic operations [6].

The periodic discrete grid method (PDRT) calculates the integrals along a set of warped lines, and does not have direct connection with the continuous Radon transform [6], [7], [8]. Still the method possesses some very nice properties, such as the exact reconstruction property, Fourier slice property, etc. Matus and Flusser [9] first investigated PDRT on \( \mathbb{Z}^2_p \), where \( p \) is prime. Then Hsung et al. [10] extended PDRT to \( \mathbb{Z}^2_p \).

The PDRT has been used in image denoising [3], tomographic reconstruction [11], image watermarking and encryption [12], etc. To compute the PDRT of an \( N \times N \) image, \( \Omega(N^2 \log N) \) arithmetic operations are required [13].

With the emergence of quantum computing, for many important computational problems, it is found that quantum algorithms can provide dramatic speedup [14], for example, exponentially fast quantum algorithms such as quantum Fourier algorithm, Shor’s factoring algorithm [15]; polynomially fast quantum algorithms such as Grover’s search algorithm [16], and so on [17].

To speed up image processing in the forthcoming quantum computing age, quantum image processing (QIMP), whose topics range from quantum image representations to image processing, has drawn a lot of attention in the last decade [18], [19], [20]. Early work on QIMP concentrated on the quantum representation of images e.g., [21], [22], [23]. Roughly speaking, there are two typical representation methods: (1) amplitude representation method, such as the Real Ket representation [24] which utilizes the qubits’ amplitude and a computational basis state to encode the grayscale and the location of a pixel respectively; (2) basis representation method, such as the novel enhanced quantum representation (NEQR) [25] which utilizes a register’s computational basis state to encode both the grayscale and the location of pixels.

Another focus in QIMP is to develop image processing tools in the quantum computation framework [25], [27], for example, Yao [28] proposed an efficient quantum image edge detection algorithm for Real Ket images. Image segmentation, watermarking, scrambling, and some other image processing problems have also been investigated in QIMP [29], [30], [31].

However, there is still no extension of Radon transform to the field of quantum image processing. In this paper, we make such extension.

We first extend PDRT to quantum computation framework. The main difficulty in making the extension comes from the fact that there is no preference of using unitary transforms in designing classical algorithms; conversely, quantum algorithms mostly use unitary operators. To realize the classical ‘non-unitary’ PDRT with unitary transformations, we design a quantum reversible multiplication, and then use Fourier slice.
property of PDRT with replacing the traditional multiplication by the quantum reversible multiplication. Finally, we obtain a transform that is different from any existing PDRTs. We call this new transform quantum Radon transform (QRT).

The QRT preserves many good properties as in the classical PDRT, such as Fourier slice property. By replicating the denoising experiments designed specially for testing PDRT [3], it is shown that the QRT is of the good denoising capability as in the classical PDRT (cf. Fig. 1). The most important advantage of QRT is that for an \( N \times N \) image, the QRT can be implemented in time \( O(\log^3 N) \), which runs exponentially faster than the classical PDRT, which has runtime \( \Omega(N^3 \log N) \). As the application, a QRT-based quantum denoising method is proposed, which runs exponentially faster than the classical PDRT denoising method.

We then extend IDT to the quantum case. The quantum extension of IDT can provide polynomial speedup, with the input being a Real ket quantum image and the output being encoded with the NEQR representation. As the application, a quantum algorithm for line detection using IDT is given, which enables the line detection process to be speeded up polynomially in the average case.

This paper is arranged as follows. In Section II, we introduce some background on Radon transform and classical/quantum image processing. In Section III, we present a reversible quantum multiplication. In Section IV, we introduce QRT and explore some basic properties of it. In Section V, we extend interpolation-based DRT to the quantum case. In Section VI, we present two applications of our proposed quantum transforms.

II. PRELIMINARIES

A. Periodic Discrete Radon Transform

Throughout this paper, we use \( p \) to denote a prime number, use \( n \) to denote a positive integer, and use \( N \) to denote a power of 2. \( [n] \) is the subset of integers \( \{0, 1, ..., n-1\} \). We use \( i \) to denote the imaginary unit. The \( L^2 \)-norm of vector \( \vec{a} = (a_0, ..., a_{n-1}) \) is \( ||\vec{a}||_2 = \sqrt{\sum_{i\in [n]} |a_i|^2} \).

We begin with a specific kind of discrete 'line', an example of which is given in Fig. 1:

**Discrete line** \( L_{l,k}^n \). The discrete line on lattice \( \mathbb{Z}_n^2 \) with interception \( l \) and slope \( k \) is

\[
L_{l,k}^n = \begin{cases} 
\{(x,y) \mid x + ky = l \mod n, x, y \in [n]\}, & \text{if } k \in [n]; \\
\{(x,l) \mid x \in [n]\}, & \text{if } k = n. 
\end{cases}
\]

The periodic discrete Radon transform (PDRT) – sometimes called finite Radon transform – is defined as summations of function values at points over these discrete lines.

**Definition 1 (Periodic discrete Radon transform [9]):** The PDRT of a function \( f \) defined on lattice \( \mathbb{Z}_n^2 \) is

\[
r_k(l) = \frac{1}{\sqrt{n}} \sum_{(i,j) \in L_{l,k}^n} f(i,j), \quad l \in [n], \ k \in [n+1].
\]

The following proposition gives an important property of PDRT:

**Proposition 1 (Fourier slice property of PDRT [3]):** Let \( \mathcal{F}_1 \) and \( \mathcal{F}_2 \) be the 1-D and 2-D discrete Fourier transform, respectively. For a function \( f \) defined on \( \mathbb{Z}_n^2 \), let \( r_k(l) \) be the PDRT of \( f \), so that it is defined on \( \mathbb{Z}_n \). Then for any \( \omega, k \in [n] \),

\[
\mathcal{F}_1\{r_k\}(\omega) = \mathcal{F}_2\{f\}(\omega, k\omega \mod n); \quad (4)
\]

for \( k = n \),

\[
\mathcal{F}_1\{r_n\}(\omega) = \mathcal{F}_2\{f\}(0, \omega). \quad (5)
\]

**Proof:** For any \( k \in [n] \), the discrete Fourier transform of function \( r_k(l) \) in variable \( l \) is

\[
\mathcal{F}_1\{r_k\}(\omega) = \frac{1}{n} \sum_{l,y \in [n]} f(l-ky,y)e^{-2\pi\sqrt{-1}\frac{l\omega}{n}}
\]

\[
= \frac{1}{n} \sum_{l,y \in [n]} f(l-ky,y)e^{-2\pi\sqrt{-1}\frac{(l-k)y+ky\omega}{n}}
\]

\[
= \mathcal{F}_2\{f\}(\omega, k\omega \mod n). \quad (6)
\]

For \( k = n \),

\[
\mathcal{F}_1\{r_n\}(\omega) = \frac{1}{n} \sum_{x,l \in [n]} f(x,l)e^{-2\pi\sqrt{-1}\frac{l\omega}{n}} = \mathcal{F}_2\{f\}(0, \omega). \quad (7)
\]

This Fourier slice property provides a fast implementation of PDRT. For an \( n \times n \) image \( f \), each value of its PDRT can be computed in time \( O(n) \), so the whole PDRT can be obtained in time \( O(n^3) \) if directly computed by definition. On the other hand, the Fourier slice property of PDRT allows one to compute the PDRT in time \( O(n^2 \log n) \); since the 1-D and 2-D (inverse) Fourier transforms can be implemented in time \( O(n \log n) \) and \( O(n^2 \log n) \) [32], and the PDRT of \( f \) can be obtained by performing 1-D inverse Fourier transform on \( \mathcal{F}_2\{f\} \) according to (4), the computational complexity of the PDRT is thus reduced to \( O(n^2 \log n) \).

Besides designing fast algorithms, another important practical issue is to recover the original image from its Radon transform. In [9], Matúš and Flusser proposed the following reconstruction formula for PDRT on \( \mathbb{Z}_p^2 \):
\textbf{Proposition 2 (Reconstruction formula for PDRT on } \mathbb{Z}_p^2): \text{ Let } r_k(l) \text{ be the PDRT of a function } f \text{ defined on } \mathbb{Z}_p^2. \text{ Then for any } i, j \in [p],
\begin{align*}
f(i, j) = \frac{1}{\sqrt{p}} \sum_{\{(l,k):(i,j)\in L_{l,k}^p\}, \ell \in [p], k \in [p+1]} r_k(l) - \frac{1}{p} \sum_{x,y \in [p]} f(x,y). \quad (8)
\end{align*}

\textbf{Proof:} The following geometric properties of discrete lines are easy to verify:
1. Every discrete line \( L_{l,k}^p \) contains \( p \) lattice points, and two parallel discrete lines have no point of intersection;
2. For any fixed slope \( k \), the \( p \) parallel lines \( L_{l,k}^p \) (where \( l \in [p] \)) provide a complete cover of the lattice \( \mathbb{Z}_p^2 \);
3. Two discrete lines \( L_{l,k}^p \) and \( L_{l',k'}^p \) of different slopes will interact in exactly one point.

For all the \( p+1 \) lines through a fixed point \((i, j)\), by term 1, every two of them have only one point in common, which is just \((i, j)\). Since there are \( p(p+1) \) points on the lines, and there are \( p+1 \) copies of point \((i, j)\) on such lines, there are all together \( p^2 \) different points on these lines, which are exactly the total number of points in lattice \( \mathbb{Z}_p^2 \):
\begin{align*}
\sum_{x,y \in [p]} f(x,y) + pf(i, j) = \sqrt{p} \sum_{\{(l,k):(i,j)\in L_{l,k}^p\}, \ell \in [p], k \in [p+1]} r_k(l). \quad (9)
\end{align*}

In general, the size of an image is not the square of a prime number. There is a series of work to extend PDRT to images of more general sizes \([10, 33, 9]\). For one example, Kingston \([34]\) extends PDRT to images of size \( p^n \times p^n \). The reconstruction formula for such images is much more complicated, e.g., (11) in \([34]\).

\textbf{B. Classical Image Denoising}

Image denoising is to remove noise from a noisy image, so as to restore the true image \([35, 36]\). Suppose we are given a real-valued noisy signal
\begin{align*}
h_i = f_i + e_i, \quad i \in [n],
\end{align*}
where \( \vec{f} = (f_0, f_1, ..., f_{n-1}) \) is original signal, and \( e_i \) is the noise sampled independently from the normal distribution \( V(0, \sigma^2) \), where \( 0 \) is the mean and \( \sigma^2 \) is the variance. After performing some denoising method on \( \vec{h} = (h_0, h_1, ..., h_{n-1}) \) to prepare the proceed signal \( \vec{h}' \), if the following noise level decreases (i.e., Risk(\( \vec{h}', \vec{f} \)) \leq Risk(\( \vec{h}, \vec{f} \))):
\begin{align*}
\text{Risk}(\vec{h}', \vec{f}) = \frac{1}{n} E(||\vec{h}' - \vec{f}||_2^2),
\end{align*}
where \( E(\cdot) \) is the expectation, then we say such denoising method is effective.

\begin{align*}
\text{SNR}(\vec{h}, \vec{f}) = 10 \log_{10} \left(\frac{||\vec{h}||_2^2}{||\vec{h} - \vec{f}||_2^2}\right),
\end{align*}
where \( f \) and \( h \) are some specific test signals. In comparison with the probability value in \([11]\), SNR is more easily accessible by numerical experiments. A denoising method is said to be effective if \( \text{SNR}(\vec{h}', \vec{f}) > \text{SNR}(\vec{h}, \vec{f}) \).

Now, we consider image denoising, i.e., 2-dimensional signal denoising. The simplest 2-D DWT denoising method is to apply 1-D denoising method to each row in the signal matrix, then use 1-D denoising method to each column. However, image possesses various geometric features. By the work of Do and Vetterli \([33]\), the following PDRT denoising method are more effective than the general 2-D DWT in denoising images with obvious singularities along straight lines, as shown in Fig. \[8\]

1. Apply the PDRT to the noisy image to obtain its PDRT \( r_k(l) \),
2) For each slope $k$, perform once 1-D DWT denoising method on the PDRT $r_k(l)$ along the direction of interception $l$.
3) Perform inverse PDRT by [9].

In [3], the authors explain why PDRT is better: by PDRT, the typical linear singularities of pure image are represented by a few wavelet coefficients (in step 2) while randomly located noisy singularities are unlikely to produce significant coefficients. This is unlike using the 2-D DWT where both noisy pixels and image singularities can produce significant wavelet coefficients. Therefore one can remove the noise with less damage to the original image by properly thresholding wavelet coefficients.

A routine computation shows that the time complexity of denoising an $n \times n$ image using 2-D DWT is $O(n^2)$. The time complexity of PDRT denoising method is $O(n^3)$, because performing inverse PDRT has a time complexity $O(n^3)$ by [9].

C. Quantum Image Representation and Preparation

**Definition 2 (Real ket representation [25]):** Let $f$ be an $N \times N$ image, where $N = 2^n$, and $f(i,j) \geq 0$ is the image intensity at point $(i,j)$, then its Real Ket representation is

$$\sqrt{\sum_{i,j\in[N]} |f(i,j)|^2} \sum_{i,j\in[N]} f(i,j)|i|j, \quad (13)$$

**Definition 3 (Novel enhanced quantum representation [25]):** The NEQR representation of image $f$ is

$$\frac{1}{N} \sum_{i,j\in[N]} |f(i,j)| |i||j|. \quad (14)$$

In this paper, we use the Real Ket representation of images in the quantum computation framework. The famous quantum Fourier transform [17] and HHL algorithm [37] are both based on this amplitude encoding method. Unless otherwise specified, the term 'quantum image' refers to an image encoded with the Real Ket method.

To show the connection between the above two representation methods, we need an efficient operation called **conditional rotation** [37], [38]:

**Proposition 3 (Conditional rotation):** Let $a$ be the $k$-bit finite precision representation of a positive number that is smaller than 1. Then the following mapping can be performed in time $O(k^{2.5})$:

$$|a|0 \rightarrow |a|(a|0 + \sqrt{1-a^2}|1)), \quad (15)$$

**Proof:** Let $a := \sum_{i\in[N]} a_i 2^{-i-1}$, where $a_i \in \{0,1\}$. Then $|a_i|$ is the $(i+1)$-st qubit of $|a| = |a_{k-1} \cdots a_1 a_0\rangle$ counted from right. For $s > 0$, define 1-qubit quantum gate

$$R_s = \begin{bmatrix} \cos s & -\sin s \\ \sin s & \cos s \end{bmatrix}. \quad (16)$$

Then it holds that

$$\prod_{i\in[k]} R_{a_{i+1}}(a_i) = \prod_{i\in[k]} R_{2^{-i}a_i} = R_a. \quad (17)$$

By performing the following $k$ successive 1-bit conditional rotations: for each $i \in [k]$ the corresponding rotation is $R_{2^{-i}a_i}$, where $a_i \in \{0,1\}$ is the control bit, on the last qubit of $|a\rangle$), one gets

$$|a\rangle|0 \rightarrow |a\rangle(\cos(a)|0 + \sin(a)|1)). \quad (18)$$

By Lemma 48 in [39], given $|a\rangle$ where $a \in (0,1)$, using Taylor series approximation allows to prepare the state $|\arccos(a)\rangle$ in time $O(k^{2.5})$, where $\arccos(a) \in [0, \frac{\pi}{2})$. So with the help of ancilla qubits and [18], the following sequence of mappings can be implemented in time $O(k^{2.5})$:

$$|a\rangle|0 \rightarrow |a\rangle|\arccos(a)\rangle|0 \rightarrow |a\rangle\arccos(a)(a|0 + \sqrt{1-a^2}|1)) \rightarrow |a\rangle|0|(|a|0 + \sqrt{1-a^2}|1)). \quad (19)$$

By the following proposition, a Real Ket image can be prepared from its NEQR version, and this preparation procedure is efficient if the condition number $\kappa := \max_{i\in[N]} |f(i,j)|/\min_{i\in[N]} |f(i,j)|$ is large enough.

**Proposition 4:** Let $\vec{a}$ be a real vector realized by a unitary operator $U : |i\rangle|0 \rightarrow |i\rangle|a_i\rangle$, $i \in [N]$ where $a_i$ is the $m$-bit finite precision representation of the vector entries, and $U$ can be performed in time $O(T_U)$. Given $a_{\max} = \max_i |a_i|$, the state $|\vec{a}\rangle$ can be prepared in time $O(T_U + \text{poly}(m, \log N)/\kappa^2)$ by the following mapping:

$$|0\rangle \rightarrow \cos \theta|\vec{a}\rangle|0 \rightarrow \cos \theta|\vec{a}\rangle|1), \quad (19)$$

where

1) $|\vec{a}\rangle := \sum_{i\in[N]} a_i |i||a_i\rangle$;
2) $a_i^\perp$ denotes the vector of entries $a_i^\perp := \sqrt{1 - |a_i/a_{\max}|^2}$ for $i \in [N]$;
3) the coefficient $\cos \theta = \sqrt{\sum_{i\in[N]} |a_i|^2 / N a_{\max}^2}$ and $\min_{i\in[N]} |a_i| / \max_{i\in[N]} |a_i| = \kappa$, and $\kappa$ is the condition number of vector $|\vec{a}\rangle$.

**Proof:** With the number $a_{\max}$ and unitary operator $U$ at hand, we can implement the following transform [37]:

$$|i\rangle|a_i\rangle \rightarrow |i\rangle|a_i/a_{\max}\rangle. \quad (20)$$

We first perform the this transform on the input state $\sum_{i\in[N]} 1/\sqrt{N} |i\rangle|a_i\rangle$ and then use the rotation conditioned on $|\vec{a}\rangle/\sqrt{N a_{\max}^2}$ (cf. Proposition 3):

$$|i\rangle|a_i/a_{\max}\rangle|0\rangle \rightarrow |i\rangle|a_i/a_{\max}\rangle(\cos \theta |0\rangle + \sin \theta |a_{\max}^0\rangle |1\rangle), \quad (21)$$

and finally undo $U$ to clear the second register. After the above operations, we measure the last qubit.

The running time of the above procedure is $O(T_U + \text{poly}(m, \log N))$. The expected result $|0\rangle$ by measurement has probability $\sum_{i\in[N]} |a_i|^2/N a_{\max}^2 > a_{\min}^2 a_{\max}^2 = \kappa^2$, which indicates that we have successfully prepared the state $|\vec{a}\rangle := \sum_{i\in[N]} a_i/\sqrt{N} |i\rangle$.
Remark: It is also possible to convert a Real Ket image to its NEQR version by applying phase estimations \[17\]. The close connections between different image representations give researchers more confidence in developing quantum image processing tools based on a particular representation, as they are likely to become universal once quantum techniques are sufficiently developed.

Quantum image preparation, also known as quantum (initial) state preparation, has been extensively studied over years, e.g., [40], [41], [38]. Current techniques allow to efficiently prepare the Real Ket (or NEQR) state of an \(N \times N\) image in time \(O(\text{polylog}N)\) if the prepared image has some special structure\[4\]. The time required to prepare an arbitrary quantum image is at most no more than the classical preparation time, up to a logarithmic factor. Specifically, a quantum image of form (13) or (14) can be prepared by applying at most \(N^2\) conditional operations, each of which has a time complexity \(\text{polylog}(N)\).

The following proposition will be used in Section[17]. It states that the inner products of quantum states can be estimated in parallel:

**Proposition 5 (Parallel swap test \[43\]):** Given \(2N\) quantum states \(|\bar{u}_0\rangle, |\bar{v}_0\rangle, \ldots, |\bar{u}_{N-1}\rangle, |\bar{v}_{N-1}\rangle\), and two state preparation unitaries: \(|k\rangle|0\rangle \rightarrow |k\rangle|\bar{u}_k\rangle\) and \(|k\rangle|0\rangle \rightarrow |k\rangle|\bar{v}_k\rangle\) (where \(k \in [N]\)) that can be implemented in time \(O(T_{in})\). There is a quantum algorithm with runtime \(O(\frac{\text{polylog}N}{\epsilon})\) to achieve \(|i\rangle|0\rangle \rightarrow |i\rangle|s_i\rangle\) for all \(i \in [N]\), where \(|s_i - \langle \bar{u}_i | \bar{v}_i \rangle| \leq \epsilon\).

A detailed proof can be found in Theorem 1 in [43].

### III. Quantum Reversible Multiplication

This section presents a reversible modular multiplication in the quantum computation framework. Fix \(N = 2^n\). For fixed \(0 \leq i < 2^{n-1}\) and \(C = 2i\), the multiplication in \(Z_N\) by \(C\) is irreversible, because the mapping: \(a \rightarrow ac \mod N\) maps both \(a = 0\) and \(a = 2^{n-1} = 0\). On the other hand, for \(D = 2i + 1\), the multiplication in \(Z_N\) by \(D\) is reversible, because if \(ad = bd \mod N\) for some \(a, b \in [N]\) and \(a \neq b\), then \((a-b)D \mid 2^n\). Since \(D\) is odd, it must be that \(a-b\) is a multiple of \(2^n\); in particular, \(|a-b| \geq 2^n\), which is impossible for \(a, b \in [N]\).

Below, we design a unitary realization of the multiplication in \(Z_N\) by any odd number \(D \in [N]\). For \(1 \leq k \leq n\),

- Let \(M_k\) be a to-be-realized unitary operator performing \(|a_k\rangle|b_k\rangle \rightarrow |a_k\rangle|a_k b_k \mod 2^k\rangle\), \(\forall a_k, b_k \in [2^k]\) and odd \(a_k\).

- Let \(A_k\) be the following controlled addition in \(Z_{2^k}\):

\[
|a_k\rangle|b_k\rangle|c\rangle \rightarrow \begin{cases} |a_k\rangle|a_k + b_k \mod 2^k\rangle|c\rangle, & \text{if } c = 1; \\ |a_k\rangle|b_k\rangle|c\rangle, & \text{if } c = 0; \end{cases}
\]

\(\forall a_k, b_k \in [2^k], c \in [2].\)

The time complexity for performing each \(k\)-qubit addition is \(O(k^2)\) \[44\], so is the time complexity of performing unitary operator \(A_k\) (see Section 4.2 in [17]).

\[\text{For any } 1 \leq k \leq n, \text{ for any } a_k \in [2^k], \text{ let the binary representation of integer } a_k \text{ be } B_{i_{k-1} \ldots i_0} := \sum_{i \in [k]} 2^i i, \text{ where } i \in [2].\]

When \(k = 1\), multiplication operator \(M_1\) is the identity:

\(|1\rangle|b_1\rangle \rightarrow |1\rangle|b_1\rangle, \forall b_1 \in [2].\)

In the following, we realize multiplication operator \(M_{k+1}\) by \(A_l\) for \(1 \leq l \leq k\) recursively.

For any \(a_{k+1}, b_{k+1} \in [2^{k+1}]\) where \(a_{k+1}\) is odd, let their binary representations be \(B_{i_{k-1} \ldots i_0}\) and \(B_{j_{k-1} \ldots j_0}\) respectively. Then their modular multiplication is

\[
(2^{k+1} i + B_{i_{k-1} \ldots i_1})(2B_{j_{k-1} \ldots j_1} + j_0) \mod 2^{k+1}
\
= (2B_{i_{k-1} \ldots i_1}B_{j_{k-1} \ldots j_1} + B_{i_{k-1} \ldots i_1} j_0 + 1 + (-1)^{j_0} 2^{k+1}) \mod 2^{k+1}
\
= (2B_{i_{k-1} \ldots i_1}B_{j_{k-1} \ldots j_1} \mod 2^k + 2^j_{j_0} B_{i_{k-1} \ldots i_1} j_0) \mod 2^{k+1}
\]

where \(\delta_{j_0}^{j_1}\) is the Kronecker symbol.

So the modular multiplication \(M_{k+1}\) of two integers \(B_{i_{k-1} \ldots i_1}, B_{j_{k-1} \ldots j_0} \in [2^{k+1}]\), each containing \(k + 1\) binary digits, can be decomposed into two operators: the modular multiplication \(M_k\) of integers \(B_{i_{k-1} \ldots i_1}, B_{j_{k-1} \ldots j_1} \in [2^k]\), which occurs in the first \(k\) binary digits, followed by the controlled modular addition \(A_k\) of the above modular multiplication result and \(B_{i_{k-1} \ldots i_1}\), which also occurs in the first \(k\) binary digits, while the control digit \(j_0 \in [2]\) remains in the last binary digit. The quantum circuit realizing this decomposition is shown in Fig. 3.

In (23) the modular multiplication \(M_k\) can be further decomposed into \(M_{k-1}\) and \(A_{k-1}\), and by doing so recursively, \(M_{k+1}\) is finally decomposed into a series of controlled modular additions: \(A_1, A_2, \ldots, A_k\). Since performing \(A_j\) has time complexity \(C_j^2\) for fixed constant \(C > 0\) and varying \(1 \leq j \leq k\), by \(1^2 + 2^2 + \ldots + k^2 = O(k^3)\), we get that the time complexity of realizing \(M_{k+1}\) (hence \(M_k\)) by (14), is \(O(k^3)\).
IV. QUANTUM RADON TRANSFORM

The classical PDRT in (4) can be viewed as a mapping \( \mathbb{R}^n \rightarrow \mathbb{R}^{n(n+1)} \) as follows:

\[
\left( f(0,0), f(1,0), \ldots, f(n-1, n-1) \right) \xrightarrow{\text{PDRT}} \left( r_0(0), r_0(1), \ldots, r_n(n-1) \right),
\]
(24)

whose transformation matrix is not unitary\(^2\). It is hard to directly design a quantum algorithm for PDRT by (4). So, we consider utilizing the related Fourier slice property in (4).

By replacing the traditional multiplication in (4) with the quantum reversible multiplication, we successfully design an efficient quantum algorithm that can realize a novel transform similar to PDRT. We name it the quantum Radon transform:

**Definition 4 (Quantum Radon transform):** Let \( f(x, y) \) be a function defined on \( \mathbb{Z}^2_n \) and \([.]\) be the floor function, and let \( \tilde{f} \) defined on \( \mathbb{Z}^2_2 \) be related to \( f \) as

\[
\tilde{f}(x', y') := \frac{1}{2} (-1)^{x' \mod n + y' \mod n} f(x', y') \mod 2, x', y' \in [2^n].
\]
(25)

Recall that \( L^2_{2n} \) represent discrete lines on the lattice \( \mathbb{Z}^2_n \):

\[
L^2_{2n} := \{(x', y') | x' + ky' = l \ (\text{mod} \ 2n)\}, \ x', y' \in [2^n].
\]

Then, the **quantum Radon transform** of \( f \) is a function defined on \( \mathbb{Z}^2_2 \) as

\[
QR_f(l, k) = \frac{1}{\sqrt{2^n}} \sum_{(x', y') \in L^2_{2n}} \tilde{f}(x', y'), \ l, k \in [2^n] \quad (26)
\]

The definition of QRT is derived from the following Algorithm\(^1\) rather than a deliberate construction, and it seems a little complicated. Now, let us give a closer observation of \( \tilde{f} \) in (25). For any \( x, y \in [n] \),

\[
f(x, y) = \tilde{f}(x, y) = -\tilde{f}(x + n, y) = -\tilde{f}(x, y + n) = \tilde{f}(x, y + n).
\]
(27)

Graphically speaking, the QRT of an image \( f \) can be viewed as the classical PDRT that performs on the symmetrized double-sized original image. Fig. 4 shows the particular summation method adopted by the QRT. Intuitively, the alternating sums seem to tend to suppress the changes in the Radon domain along the direction of interception, resulting in a lower sensitivity of QRT to linear singularities. Even worse, only a half of slopes (lines) could be detected by QRT, since when slope \( k \) is even, by (27) it holds that

\[
QR_f(l, k) = 0.
\]

However, by experiments (see Fig. 8), it has been found that QRT preserves the good denoising capability as in the classical PDRT; it will be further discussed in Section VI-A later. We hold the view that the QRT has a practical value comparable to that of the classical PDRT. On the other hand, as an advantage, QRT is invertible, which is derived from the following invertible QRT algorithm. The most important advantage lies in that the QRT can be performed exponentially faster than the classical PDRT, by the following quantum Radon transform algorithm:

\[\text{Algorithm 1 Quantum Radon transform}\]

**Input:** An \( N \times N \) quantum image \( \sum_{x,y \in [N]} f(x,y) |x⟩|y⟩ \).

**Output:** \( \sum_{l,j'' \in [2N]} QR_f(l,j'')(|l⟩|j'’⟩) \).

1. Prepare the state \( \sum_{x,y \in [N]} f(x,y) |x⟩|y⟩ |1⟩ |1⟩ \).

2. Apply phase shifts conditioned to \( |x⟩ |y⟩ \) to prepare

\[
\sum_{x,y \in [N]} g(x,y) |x⟩ |y⟩ |1⟩ |1⟩,
\]
(29)

where \( g(x,y) = f(x,y) e^{-2\pi i \frac{1}{2N} xy} \).

3. Perform quantum Fourier transforms on \( |x⟩ \) and \( |y⟩ \) respectively to get

\[
\sum_{i,j'' \in [N]} \left( \frac{1}{N} \sum_{x,y \in [N]} g(x,y) e^{-2\pi i \frac{1}{2N} x y} \right) |i⟩ |1⟩ |j''⟩ |1⟩.
\]
(30)

4. Perform inverse quantum reversible multiplication on registers \( |i'⟩ = |i,1⟩ \) and \( |j''⟩ = |j,1⟩ \) to prepare

\[
\sum_{i',j'' \in [2N]} \left( \frac{1}{2N} \sum_{x',y'' \in [2N]} \tilde{f}(x',y'') e^{-2\pi i \frac{1}{2N} x'y''} \right) |i'⟩ |j''⟩.
\]
(31)

where \( i' \odot j'' = j' \), \( \odot \) is the quantum reversible multiplication such that \( i' \odot j'' = j'' \) for odd \( i' \in [2N] \).

5. Apply inverse quantum Fourier transform to \( |i'⟩ \) to prepare

\[
\sum_{l,j'' \in [2N]} QR_f(l,j'')(|l⟩ |j''⟩).
\]
(32)

where \( i' \odot j'' = j' \).

\[\text{Fig. 4. A toy example to show the different summation methods adopted to compute the PDRT (left) of} f \text{ and QRT (right) of} f, \text{ where} f \text{ is a function defined on} \mathbb{Z}^2_1. \text{ Fig. 4a shows the points used to compute} QR_f(1, -3) \text{ mod} 2n. \text{ Fig. 4b shows the points used to compute} QR_f(1, 1) \text{ mod} 2n.\]

---

\(^2\)The lines \( L^2_{2n} \) have intersection, so the rows of transformation matrix corresponding to \( r_0(0) \), \( r_1(0) \) are not orthogonal.
A. Analysis of Algorithm \([\text{I}]\)

The Step 1 of Algorithm \([\text{I}]\) is adding two extra qubits initially in the state \(|1\rangle\rangle\). In Step 2, the conditional phase shifts can be implemented in time logarithmic in \(N\), by a method similar to the implementation of conditional rotation (cf. Proposition \([3]\)). After applying Step 3, the resulting state can be viewed as the quantum Fourier transform of some \((2N + 2)\)-qubit state

\[
\sum_{x, y, i, j \in [N]} g'(x', y') |x'| |y'|, \quad \text{i.e.,}
\]

\[
\begin{align*}
F_2\{g'(i, j)\} & (2i + 1, 2j + 1) = F_2\{g(i, j), \quad i, j \in [N], \quad (34) \\
F_2\{g'(i, j)\} &= 0, \quad i, j \in [2N],
\end{align*}
\]

The explicit expression of function \(g'\) can be deduced by inverse Fourier transform as follows:

\[
g'(l', t') = \frac{1}{2N} \sum_{x, y, i, j \in [N]} g(x, y) e^{-2\pi i \frac{lx + ly}{N}},
\]

\[
\times e^{2\pi i \frac{2l + 1}{2N} x + 2t + 1},
\]

\[
= \frac{1}{2N} \sum_{l, t \mod N} \left( \sum_{x, y \in [2N]} \tilde{f}(x', y') e^{-2\pi i \frac{x' + y' + l l' + x t'}{N}} \right) |i'\rangle |j'\rangle,
\]

\[
(35)
\]

After applying inverse quantum reversible multiplication to \(|i'\rangle |j'\rangle\) of \(\text{Step 4}\), by \(\text{22}\), the resulting state is as in \(\text{31}\). Now, from the fact that for any \(j' \in [2N]\),

\[
\begin{align*}
\{i' \odot j'' = i'' \}, & \quad \text{if } i' \text{ is odd,} \\
F_2\{\tilde{f}\}(i', j'') = 0, & \quad \text{if } i' \text{ is even,}
\end{align*}
\]

one gets

\[
\begin{align*}
F_2\{\tilde{f}\}(i', j''') &= F_2\{\tilde{f}\}(i', j''), \\
& \quad i, j \in [2N].
\end{align*}
\]

Then the resulting state in \(\text{Step 4}\) can be rewritten as

\[
\sum_{i', j''} F_2\{\tilde{f}\}(i', j''') |i'\rangle |j''\rangle.
\]

The final result in \(\text{Step 5}\) is by the following proposition:

**Proposition 6:** [Fourier slice property of QRT] In the notations of Algorithm \([\text{I}]\), for any given slope \(j''' \in [2N]\), the 1-D Fourier transform of \(QR_T(l, j'')\) with respect to \(l\) is

\[
F_1\{QR_T(l, j'')\}(i') = F_2\{\tilde{f}\}(i', j'''), \quad i' \in [2N].
\]

**Proof:** Let \(F_1^{-1}\) be the inverse transform of \(F_1\). Then,

\[
F_1^{-1}\{\tilde{f}\}(i', j''') \quad \text{is the inverse transform of } F_1{\tilde{f}}.
\]

\[
\begin{align*}
F_1^{-1}\{\tilde{f}\}(i', j''') &= \frac{1}{\sqrt{2N}} \sum_{i'' \in [2N]} \left( \sum_{x, y \in [2N]} \tilde{f}(x, y) e^{-2\pi i \frac{ix + iy}{N}} \right) e^{2\pi i \frac{ix + iy}{N}}
\]

\[
= \frac{1}{\sqrt{2N}} \sum_{i', t \in [2N]} \left( \sum_{x, y \in [2N]} \tilde{f}(x, y) e^{-2\pi i \frac{x + y + l l' + x t'}{N}} e^{2\pi i \frac{x + y}{N}} \right)
\]

\[
= \frac{1}{\sqrt{2N}} \sum_{i', t \in [2N]} \left( \sum_{x, y \in [2N]} \tilde{f}(x, y) e^{-2\pi i \frac{x + y + l l' + x t'}{N}} e^{2\pi i \frac{x + y}{N}} \right)
\]

\[
= \frac{1}{\sqrt{2N}} \sum_{(x, y) \in L_{i', t'}^N} \tilde{f}(x, y) = QR_T(l, j'''),
\]

\[
(37)
\]

The overall runtime of Algorithm \([\text{I}]\) is the sum of \(O(\log N)\) time required to perform conditional phase shifts \([45]\), \(O(\log^2 N)\) time required to perform quantum Fourier transform and its inverse, and \(O(\log^4 N)\) time required to perform the inverse multiplication. Observe that all quantum gates used in Algorithm \([\text{I}]\) are unitary and invertible. So, an efficient inverse QRT algorithm follows immediately by reversing all circuits in Algorithm \([\text{I}]\). In conclusion, we have the following theorem:

**Theorem 1:** Given an \(\times N\) quantum image state

\[
\sum_{x, y \in [N]} f(x, y) |x\rangle |y\rangle,
\]

then its (inverse) QRT can be performed in time \(O(\log^3 N)\).

V. QUANTUM-MECHANICAL IMPLEMENTATION OF DRT WITH THE INTERPOLATION METHOD

The discretization of Radon transform needs to approximate the line integral. A simple solution is to sample and sum along the integral line, where the interpolation method can be employed to estimate the undefined values at non-integer lattice sample points. Several kinds of sample and interpolation methods are available for implementing IDRT \([46]\), \([47]\), \([48]\). Below, we define a simplest kind of IDRT (SIDRT), and give its quantum algorithm, with the aim of showing a general quantum approach to achieving this interpolation-based kind of DRT.

**Definition 5:** Let the set of slopes of the basically horizontal lines and basically vertical lines be

\[
S^\parallel := \{\tan \theta | \theta = \frac{\pi}{N}, -\frac{N}{4} \leq j < \frac{N}{4} \text{ and } j \in \mathbb{Z}\},
\]

\[
S^\perp := \{\cot \theta | \theta = \frac{\pi}{N}, -\frac{N}{4} < j \leq \frac{N}{4} \text{ and } j \in \mathbb{Z}\},
\]

respectively, where agree that \(\frac{\pi}{8} = \infty\) and \(\frac{\pi}{8} = 0\). Let \(\Delta_{ki}\) be the fractional part of \(ki\). Then the SIDRT of a function \(f\) on \(\mathbb{Z}_N^2\) is

\[
P_k(l) = \frac{1}{\sqrt{N}} \sum_{i \in [N]} \left[ \sqrt{1 - |\Delta_{ki}|^2} f(i, l + [ki] \mod N) \right]
\]

\[
+ \Delta_{ki} f(i, l + [ki] + 1 \mod N), \quad k \in S^\parallel, l \in [N],
\]

\[
P_k(l) = \frac{1}{\sqrt{N}} \sum_{i \in [N]} \left[ \sqrt{1 - |\Delta_{ki}|^2} f(i + l \mod N, i) \right]
\]

\[
+ \Delta_{ki} f(l + [\frac{l}{k}] + 1 \mod N, i), \quad k \in S^\perp, l \in [N].
\]

By definition, \(P_k(l)\) can be viewed as an approximate discrete line integral along the line with interception \(l\) and slope \(k\).

The SIDRT is proposed for easily achieved in the quantum case. Although it adopts a relatively simple interpolation method, the SIDRT is so useful that enables to detect lines in complicated image, as shown in Figure \([\text{10}]\), more details can be found in Section \([\text{VI-B}]\).

Now, we turn to the quantum implementation of the SIDRT. We first consider approximating the integrals along basically horizontal lines, i.e., \(P_k(l)\) as defined in \(\text{39}\) where \(l \in [N]\) and \(k \in S^\parallel\). Let \(k_0 := \tan(\frac{\pi}{8} - \frac{\pi}{8})\). For any \(i \in [N], \theta \in [\frac{\pi}{8}]\), we can prepare a quantum state that contains the location information.
of lattice points used to compute \( P_{k_0}(l) \), by the following sequence of mappings:

\[
|\theta\rangle|0\rangle \rightarrow |\theta\rangle|ik_0\rangle |0\rangle \\
\rightarrow |\theta\rangle|ik_0\rangle |\Delta_{ik_0}\rangle \left( \sqrt{1 - |\Delta_{ik_0}|^2} |0\rangle + |\Delta_{ik_0}| |1\rangle \right) \\
\rightarrow |\theta\rangle|ik_0\rangle |0\rangle \left( \sqrt{1 - |\Delta_{ik_0}|^2} |0\rangle + |\Delta_{ik_0}| |1\rangle \right).
\]

In (41), the first step is by trigonometric function \( |\theta\rangle \) and arithmetic \( i \) in computation basis. The second step is to use the decimal part of \( ik_0 \) to perform control rotations (cf. Proposition 5) on the last qubit. The third step is uncomputing \( |\Delta_{ik_0}\rangle \).

Combining (41) with the addition in the computational basis gives the following lemma:

**Lemma 2:** The following ‘location state’ can be prepared in time \( O(\text{polylog} N) \):

\[
|\theta\rangle|0\rangle \rightarrow |\theta\rangle \sum_{i \in [N]} \frac{1}{\sqrt{N}} \left( \sqrt{1 - |\Delta_{ik_0}|^2} |i\rangle |ik_0\rangle \right) |0\rangle \\
+ |\Delta_{ik_0}\rangle|0\rangle \left( \sqrt{1 - |\Delta_{ik_0}|^2} |0\rangle + |\Delta_{ik_0}| |1\rangle \right), \quad \theta \in \left[ \frac{N}{2}, 2 \right].
\]

**Remark:** The state on the right hand side of (42) is called ‘location state’, since its first two qubits record the locations of points used to compute \( P_{k_0}(0) \) for \( k \in S^\perp \), according to \( S^\perp \).

Now, with the unitary capable of preparing a quantum image \( f \) as follows:

\[
|0\rangle \rightarrow \sum_{i,j \in [N]} f(i,j) |i\rangle |j\rangle,
\]

one can approximate the SIDRT of \( f \) by the following theorem:

**Theorem 3:** Given the unitary that can prepare a quantum image \( \sum_{i,j \in [N]} f(i,j) |i\rangle |j\rangle \) in time \( O(T_{in}) \), one can approximate \( P_{k_0}(l) \), the SIDRT of \( f \), to precision \( O(\epsilon) \) in time \( O(T_{in} \text{polylog} N) \). Namely, there is a quantum algorithm runs in time \( O(T_{in} \text{polylog} N) \) to achieve the mapping

\[
|\theta\rangle|l\rangle |0\rangle \rightarrow |\theta\rangle|l\rangle |\tilde{P}_{k_0}(l)\rangle,
\]

\( \forall l \in [N], \theta \in [N] \),

where \( |\tilde{P}_{k_0}(l) - P_{k_0}(l)| \leq \epsilon \), and \( k_0 := \tan(\frac{\pi}{N} - \frac{\pi}{2}) \) such that \( k_0 \in S^\perp \cup S^\perp \) as defined in (39).

**Proof:** We first consider the realization of \( \tilde{P}_{k_0}(l) \) for \( \theta \in \left[ \frac{N}{2}, \ldots, \frac{N}{2} \right] \), i.e., the case where \( k_0 \in S^\perp \). Observe that the following ‘image state’ can be prepared in time \( O(T_{in} \text{polylog} N) \) by the mapping

\[
|\theta\rangle|l\rangle |0\rangle \rightarrow |\theta\rangle|l\rangle \sum_{i,j \in [N]} f(i,j) |i\rangle |j\rangle \left( (\sqrt{2} |0\rangle + \sqrt{2} |1\rangle) \right),
\]

where \( \theta \in \left[ \frac{N}{2}, \ldots, \frac{N}{2} \right] \) and \( l \in [N] \). Lemma 2 allows us to prepare the ‘location state’ by the mapping

\[
|\theta\rangle|l\rangle |0\rangle \rightarrow |\theta\rangle|l\rangle \sum_{i \in [N]} \frac{1}{\sqrt{N}} \left( \sqrt{1 - |\Delta_{ik_0}|^2} |l\rangle |i\rangle |ik_0\rangle \right) \\
+ |\Delta_{ik_0}\rangle|0\rangle \left( \sqrt{1 - |\Delta_{ik_0}|^2} |l\rangle |1\rangle \right).
\]

Now, by parallel swap test (Proposition 5), one can approximate the \( \frac{N}{2} \) inner products of the ‘image state’ in (45) and the ‘location state’ in (46) in parallel \( \theta, l \). Since the result of each inner product is just \( \frac{1}{\sqrt{2}} \tilde{P}_{k_0}(l) \), one can perform

\[
|\theta\rangle|l\rangle |0\rangle \rightarrow |\theta\rangle|l\rangle |p_{k_0}(l)\rangle, \quad \forall \theta \in \left[ \frac{N}{2}, \ldots, \frac{N}{2} \right], \quad l \in [N],
\]

in time \( O(\sqrt{\frac{T_{in}}{2}} \text{polylog} N) \), where \( |p_{k_0}(l) - \frac{1}{\sqrt{2}} \tilde{P}_{k_0}(l)| \leq \frac{\pi}{\sqrt{2}} \).

The theorem holds by setting \( \tilde{P}_{k_0}(l) = \sqrt{2}p_{k_0}(l) \). An efficient implementation of multiplication by \( \sqrt{2} \), a known constant, in the computation basis can be found in [48]. The realization of (44) for \( \theta \in \left[ \frac{N}{2}, \ldots, \frac{N}{2} \right] \) is similar.

Given a Real Ket quantum image \( f \) and its preparation unitary as in (44), by Theorem 3 after performing the transform of (44) on input state \( \sum_{\theta \in [N]} \frac{1}{\sqrt{N}} |\theta\rangle|l\rangle |0\rangle \), one can prepare the SIDRT of \( f \) in NEQR encoded form. This NEQR encoded output is already is can be used for practical application, such as line detection shown in Section VI-B later. Moreover, if necessary, one can continue to transform the NEQR encoded output into its Real Ket version by Proposition 4 thus keeping the input and output encoded in the same way.
A. Efficiency Analysis of Theorem 4

Below, we discuss what is a reasonable choice of the precision \(\epsilon\) in Theorem 4.

**Random image.** The term ‘random image’ refers to an image whose each pixel value is sampled from the uniform distribution \(U[0, 1]\) independently.

**Proposition 7:** Let \(f\) be an \(N \times N\) random image, i.e., \(f(i, j) \sim U[0, 1]\) for \(i, j \in [N]\). Then the expectation of the minimal value of the SIDRT of normalized \(f\) is no less than \(\Omega(N^3)\).

**Remark:** The reason why we consider the normalized rather than the original image here is that the quantum image is a normalized state.

**Proof:** We begin with a probability inequality. Let \(a, b\) be two discrete random variables whose density function are \(P(a = a_i) = p_i, P(b = b_j) = q_j\), where \(i \in [N]\), respectively. Then by Cauchy-Schwarz inequality,

\[
E\left(\frac{a}{b}\right) = \sum_{i,j \in [N]} p_i q_j \frac{a_i}{b_j} = \frac{\left(\sum_{i,j} p_i q_j (a_i b_j)\right)\left(\sum_{i,j} p_i q_j \frac{a_i}{b_j}\right)}{E(b)} \geq \frac{\left(\sum_{i,j} p_i q_j (a_i b_j)^2\right)\left(\sum_{i,j} p_i q_j \frac{a_i}{b_j}\right)}{E(b)} = \frac{E(\sqrt{a})}{E(b)} E\left(\frac{a}{b}\right). \tag{48}
\]

Let \(U[0, 1]\) be the uniform distribution on \([0, 1]\), and let random variables \(x_k, y_k \sim U[0, 1]\) for \(k \in [N^2]\). Notice that \(f(i, j) \sim U[0, 1]\). Let \(P_{k_0}(l_0)\) be the SIDRT of \(f\) as in (39), then for any \(k_0, l_0 \in [N]\), the expectation of the SIDRT of normalized \(f\) at point \((k_0, l_0)\) is

\[
E\left(\frac{P_{k_0}(l_0)}{\sqrt{\sum_{i,j \in [N]} x_l^2 + \sum_{k \in [N^2-N]} y_k^2}}\right) \geq E\left(\sqrt{\sum_{k \in [N]} x_k} \right) \approx \sqrt{\frac{3}{2N^{0.5}}} (N \to \infty), \tag{49}
\]

where the last inequality follows from that \(E(\nu) \leq E(\nu^2)\) for arbitrary random variable \(\nu\), and the last approximate equality is by the central limit theorem [49] which states that for any central and independently distributed (IID) random variables \(\nu_0, \nu_1, \ldots, \nu_{n-1}\) with mean \(\mu\) and variance \(\sigma^2\), \(E\left(\sqrt{\sum_{j=0}^{n-1} \nu_j}\right) \approx \sqrt{n \mu} - \frac{\sigma^2}{\mu} (n \to \infty)\), and so

\[
E\left(\sqrt{\sum_{k \in [N]} x_k}\right) \approx \sqrt{\frac{N}{2} - \frac{1}{6}}.
\]

The above Proposition implies that a reasonable choice of precision in Theorem 4 is \(\epsilon = \Theta(\frac{1}{N^3})\), e.g., \(\epsilon = \frac{1}{100N^3}\), for producing a good approximation to SIDRT. In this case, when compared with the classical SIDRT whose running time is \(\Omega(N^3)\), quantum SIDRT achieves a polynomial speedup by Theorem 4 because the quantum image preparation time \(T_{in} \leq O(N^2)\) in the worst case, thus \(\frac{T_{in}}{T_{Cl}} \leq O(N^{2.5}) < N^3\).

VI. QUANTUM APPLICATION

We present two potential applications—quantum image denoising using QRT and quantum line detection using SIDRT.

A. Denoising using QRT

We replicate the denoising experiments in [4], which are specially designed for testing PDRT. The experiment results shown in Fig. 4 suggest that our QRT is of the good denoising capability as in the classical PDRT. Below, we give an efficient quantum image denoising algorithm using QRT.

**Step 1.** Apply the QRT algorithm on the input \(N \times N\) quantum image \(f\) to perform

\[
\sum_{x, y \in [N]} f(x, y)|x\rangle\langle y| \rightarrow \sum_{l, k \in [2N]} QR_f(l, k)|l\rangle\langle k|.
\]

**Step 2.** Perform Hadamard gate (namely, \(2 \times 2\) Haar transform as shown in Fig. 2) on the least significant qubit (LSB) of \(|l\rangle\) in (50) to produce a state

\[
\frac{1}{\sqrt{2}} \sum_{l' \in [N^2]} (QR_f((2l' + 1, k) + QR_f((2l' + 1, k))|l'0\rangle|k\rangle
+ (QR_f((2l' + 1, k) - QR_f((2l' + 1, k))|l'1\rangle|k\rangle. \tag{51}
\]

**Step 3.** Measure the LSB of the first register. If the outcome is \(|0\rangle\), then apply the inverse Hadamard gate to the measured qubit, and then execute the inverse QRT algorithm. The resulting state is (up to a normalization factor):

\[
QR^{-1}\left( \sum_{l' \in [N], x \in [2]} \frac{1}{\sqrt{2}} (QR_f((2l' + 1, k) + QR_f((2l' + 1, k))|l'x\rangle|k\rangle
\right)
\]

The above is a quantum analogue of PDRT denoising method introduced in Section II-B. Notice that the Hadamard gate in Step 2 plays the role of Haar wavelet, and measuring LSB to obtain \(|0\rangle\) in Step 3 has the effect of making all wavelet coefficients zero. So, this quantum QRT denoising method indeed simulates denoising using QRT+Haar wavelet and threshold \(\infty\).

To evaluate the efficiency of this denoising method, we consider the success probability (namely, the probability of measuring \(|0\rangle\) in Step 3). Since \(QR_f(l', k) = 0\) holds for any even \(k \in [2N]\), the success probability is thus reduced to:

\[
\frac{1}{2} \sum_{l', k \in [2N]} |QR_f(2l', 2k + 1) + QR_f(2l' + 1, 2k + 1)|^2. \tag{52}
\]

By the following theoretical analysis (cf. Proposition 8) and numerical simulation (cf. Fig. 9), this probability of success is quite high, particularly, it has no tendency to be tiny as the image size \(N\) increases.

**Proposition 8:** Let \(f\) be an \(N \times N\) random image, i.e., \(f(i, j) \sim U[0, 1]\) for \(i, j \in [N]\). Given the noisy image
\[
h(i, j) = f(i, j) + e_{ij}, \quad i, j \in [N],\]
where \(e_{ij}\) is the noise sampled from the normal distribution \(V(0, \sigma^2)\) independently, and \(\epsilon \in \mathbb{R}\) is noise level, then the QRT denoising method can be implemented with an average probability of success \(p > 1/2\).

**Proof:** By Definition 4, for any fixed \(l, k \in [2N]\),
\[
QR_h(l, k) - QR_f(l, k) = \frac{1}{2\sqrt{2N}} \sum_{(i, j) \in L^N_{lk}} \epsilon e_{ij}. \quad (53)
\]
Notice that for two independent random variables \(x \sim V(u_1, \sigma_x^2), \ y \sim V(u_2, \sigma_y^2)\), it holds that \(x \pm y \sim V(u_1 \pm u_2, \sigma_x^2 + \sigma_y^2)\). So, the distribution of the right-hand side of (53) is \(\epsilon V(0, \sigma^2/4)\). Let \(E_{f,e}(QR_h)\) denote the expectation of \(QR_h\) with respect to the random variables \(f, e\). Then, for any \(l, k \in [2N]\),
\[
E_{f,e}([QR_h(l, k) - QR_h(l + 1, k)]^2) = E_f([QR_f(l, k) - QR_f(l + 1, k)]^2) + E_e([\epsilon^2 / 2\sqrt{2N}] \sum_{s \in [2N]} e_s - \sum_{s \in [2N]} e_s')^2) \quad (54)
\]
\[
= E_f([QR_f(l, k) - QR_f(l + 1, k)]^2) + \epsilon^2 \sigma^2 / 2, \quad (55)
\]
where \(e_s, e_s' \sim V(0, \sigma^2)\) are both the short hands of IID noises \(e_{ij}\) in (53). (54) is by combining (53) and the fact that the mean of random variables \(e_s, e_s'\) is 0. Similarly,
\[
E_{f,e}([QR_h(l, k) + QR_h(l + 1, k)]^2)
\]
From (52), our aim is to lower bound the following kind of

\[ (\text{SNR}) \text{ indicates a better performance of denoising. So, } QR_T, \text{ with a comparable performance to } PDR_T, \text{ is better than DWT at handling the image which is piecewise smooth with singularities along a straight line (here is the diagonal line).} \]

\[ E_f(|QR_f(l, k) + QR_f(l + 1, k)|^2) + \epsilon^2 \sigma^2 / 2. \quad (56) \]

Now, we denote

\[ \Delta_+ = \sum_{l,k \in [N]} |QR_h(2l, 2k + 1) + QR_h(2l + 1, 2k + 1)|^2 \]
\[ \Delta_- = \sum_{l,k \in [N]} |QR_h(2l, 2k + 1) - QR_h(2l + 1, 2k + 1)|^2 \]
\[ \Delta'_+ = \sum_{l,k \in [N]} |QR_f(2l, 2k + 1) + QR_f(2l + 1, 2k + 1)|^2 \]
\[ \Delta'_- = \sum_{l,k \in [N]} |QR_f(2l, 2k + 1) - QR_f(2l + 1, 2k + 1)|^2. \]

From (62), our aim is to lower bound the following kind of average probability of success:

\[ P_{\text{success}} = \frac{E_{f_x}(\Delta_+ - \Delta_-)}{E_{f_x}(\Delta_+ + \Delta_- + 2N^2 \sigma^2 / 2)} \quad \text{by} \quad (55), \ (50) \]

where the denominator in (57) is by the following relations:

\[ \sum_{l,k \in [N]} \Delta'_+ + \Delta'_- = \sum_{l,k \in [2N]} 2|QR_f(l, k)|^2 = 2 \sum_{i,j \in [N]} |f(i,j)|^2 \]

\[ \text{by (50), } QR_f(l, k) \text{ is a sum of values } \hat{f}(i,j) \text{ over line } L_{l,k}^{2N} \text{ on } \mathbb{Z}_{2N}^2, \text{ where each } \hat{f}(i,j) \text{ takes values from uniform distribution on } [0,1] \text{ for } (i,j) \in P_1, \text{ and each } \hat{f}(i,j) \text{ takes} \]

\[ \text{Notice that if } k \text{ is even, then } QR_f(l, k) \equiv 0. \text{ We first consider estimating } E_f(|QR_f(l, k) + QR_f(l + 1, k)|^2) \text{ and} \]
\[ E_f(|QR_f(l, k) - QR_f(l + 1, k)|^2) \text{ for any odd } k \in [2N]. \]

Denote intervals

\[ P_1 := [0, N) \times [0, N) \cup [N, 2N) \times [N, 2N), \]
\[ P_2 := [0, N) \times [N, 2N) \cup [N, 2N) \times [0, N), \quad (59) \]

and denote the number of points in \( L_{l,k}^{2N} \cap P_1, L_{l,k}^{2N} \cap P_2 \) by

\[ C_1(L_{l,k}^{2N}) = \text{card} \{ (x, y) \in L_{l,k}^{2N} \mid (x, y) \in P_1 \}, \]
\[ C_2(L_{l,k}^{2N}) = \text{card} \{ (x, y) \in L_{l,k}^{2N} \mid (x, y) \in P_2 \}. \quad (60) \]

respectively. By geometry, it can be verified that

\[ C_1(L_{l,k}^{2N}) + C_2(L_{l,k}^{2N}) = 2N, \quad \forall l,k \in [2N], \quad (61) \]
\[ |C_1(L_{l,k}^{2N}) - C_1(L_{l+1,k}^{2N})| = 2, \quad \forall \text{ odd } k \in [2N]. \quad (62) \]
values from uniform distribution on $[-1,0]$ for $(i,j) \in P_2$. Combining (61) and (62) gives that for any odd $k$,

\[
E_f(|QR_f(l,k) - QR_f(l+1,k)|^2) = \frac{1}{8N} E\left(\sum_{C_1(l)+C_2(l+1)} \nu - \sum_{C_1(l)+C_2(l+1)} \nu^2\right) = \frac{1}{8N} E\left(\sum_{2N+2} \nu - \sum_{2N-2} \nu^2\right),
\]

where $C_1(l)$ is the short hand of $C_1(L_{b,k}^N)$, and $\sum_k \nu$ denotes the summation of $k$ numbers of independent random variables $\nu \sim \mathbb{U}[0,1]$. Notice that for IID random variables $\nu_1, \nu_2 \sim \mathbb{U}[0,1]$, we have $E(|\nu_1|^2) = |E(\nu_1)|^2 + D(\nu_1)$, where $D$ is the variance, $D(\nu_1 + \nu_2) = 2D(\nu_1)$, and $E(\nu_1 + \nu_2) = E(\nu_1) \pm E(\nu_1)$. So,

\[
E\left(\sum_{2N+2} \nu - \sum_{2N-2} \nu^2\right) = D\left(\sum_{2N+2} \nu - \sum_{2N-2} \nu\right) + E\left(\sum_{2N+2} \nu - \sum_{2N-2} \nu\right) = \frac{N}{3} + 2.
\]

Combining (64) and (63) gives

\[
E_f(\sum_{\theta, l \in [N]} |QR_f(2l, 2k + 1) - QR_f(2l+1, 2k+1)|^2) = N^2 \times \frac{1}{8N} \left(\frac{N}{3} + 2\right) = \frac{N^2}{24} + \frac{N}{4}.
\]

Then by (57), (58), (64), and $E\left(\sum_{i,j \in [N]} |f(i,j)|^2\right) = \frac{N^2}{3}$, the lower bound is achieved as follows:

\[
P_{\text{success}} = \frac{E(2|f|^2) - (\frac{N^2}{24} + \frac{N}{4}) + N^2\sigma^2}{E(2|f|^2) + N^2\sigma^2} = 1 - \frac{1}{\frac{N^2}{24} + \frac{N}{4} + \sigma^2} > 1/2.
\]

Remark: For a real-world image, due to the smoothness of images, the ratio of wavelet coefficients to filter coefficients is likely to be very small, so that the probability of success of quantum denoising method is likely to be high, such as 95% shown in Figure 9.

By treating the probability of success as a constant, we conclude that the QRT-based quantum image denoising method has a time complexity $O(N^3)$, which is exponentially faster than the classical PDRT denoising method.

Also, there are more quantum techniques that allow to simulate denoising using other Daubechies wavelets [26] and thresholds [83] in the quantum computation framework.

B. Line detection using SIDRT

The SIDRT enables one to detect possible line in images: when the SIDRT $P_\theta(l)$ of image $f$ reaches maximum for some pair $l$ and $k$, there is likely to be a straight line with gradient $k$ and interception $l$ in the image. For example, suppose that $f$ depicts a line composed of points with grayscale 1, while the grayscale of all other points in the picture are 0. Then the SIDRT reaches maximum at the corresponding line, with the coordinates of the maximum in the Radon domain giving the slope and interception of the line respectively. Fig. 10 shows a powerful line detection capability possessed by SIDRT.

In classical case, performing SIDRT has a running time $\Omega(N^3)$, and finding the maximum of SIDRT has a time complexity $O(N^2)$. So, the total running time of classical SIDRT-based line detection algorithm is $\Omega(N^3)$; in contrast, there is a quantum algorithm that can run in $O(N^{1.5})$ time to execute line detections, by the following proposition:

**Proposition 9:** Let $f$ be an $N \times N$ quantum image that can be prepared in time $O(T_{in})$ by the unitary $U : |0\rangle \rightarrow |\tilde{f}\rangle$. There exist a quantum algorithm that can detect line in $f$ by using SIDRT in time $O(\frac{N^{1.5}}{\epsilon}T_{in}\log N)$, where $\epsilon$ is the desired precision. This algorithm outputs a pair $(\theta_0, l_0)$ such that $P_{\theta_0}(l_0) \geq (1 - \frac{2}{\sqrt{3}}) \max_{\theta,l} P_{\theta}(l)$.

**Proof:** Applying Theorem 3 to the input state $\sum_{\theta, l \in [N]} \frac{1}{N} |\theta\rangle |l\rangle |0\rangle$ with setting precision $\frac{\sqrt{\epsilon}}{T_{in}\log N}$, one can prepare the following state in time $O(\frac{N^{1.5}}{\epsilon}T_{in}\log N)$:

\[
\sum_{\theta, l \in [N]} \frac{1}{N} |\theta\rangle |l\rangle |\tilde{P}_{\theta}(l)\rangle,
\]

where $|\tilde{P}_{\theta}(l) - P_{\theta}(l)| \leq \frac{\sqrt{\epsilon}}{T_{in}\log N}$. Then by the quantum algorithm for finding the maximum among $N$ terms with query complexity $O(\sqrt{N})$ [50], one can find $(\theta_0, l_0)$ such that $P_{\theta_0}(l_0) = \max_{\theta,l} \tilde{P}_{\theta}(l)$ in time $O(\frac{N^{1.5}}{\epsilon}T_{in}\log N)$.

By Proposition 7 in the average case, it holds that

\[
\max_{\theta,l} P_{\theta}(l) \geq \frac{\sqrt{3}}{2N^{0.5}}.\quad \text{Now that}
\]

\[
|\tilde{P}_{\theta}(l) - P_{\theta}(l)| \leq \frac{\epsilon}{N^{0.5}} \leq \frac{2\epsilon}{\sqrt{3}} \max_{\theta,l} P_{\theta}(l),
\]

it holds that

\[
\tilde{P}_{\theta_0}(l_0) = \max_{\theta,l} \tilde{P}_{\theta}(l) \geq (1 - \frac{2\epsilon}{\sqrt{3}}) \max_{\theta,l} P_{\theta}(l).
\]

VII. CONCLUSION

This paper presents a novel discrete Radon transform for efficient quantum implementation. By theoretical analysis and
maximum at point \((13, 246)\), which implies there may be a straight line of the gradient \(-1.251\) and a distinct straight line connecting \((228, 53)\) and \((97, 217)\) is of the gradient of China (Grant Nos. 11471040 and 11761131002).

Fig. 10. Line detection using SIDRT. The upper is the detected image, where a classical application value that it can be used as a ‘reversible PDRT’.

This work is supported by Chinese Postdoctoral Science Foundation Grant No. 2020M680716, China National Key Research and Development Projects 2020YFA0712300, 2018YFA0704705, and National Natural Science Foundation of China (Grant Nos. 11471040 and 11761131002).

Fig. 10. Line detection using SIDRT. The upper is the detected image, where a classical application value that it can be used as a ‘reversible PDRT’.

numerical experiments, it is shown that our new proposed quantum Radon transform has similar functionality to the classical PDRT, and is more suitable for quantum implementation than PDRT. In addition to its quantum advantage, the QRT has a classical application value that it can be used as a ‘reversible PDRT’.

Also, a polynomially fast quantum implementation of another interpolation-based kind of DRT is given. There are two problems that deserve further investigation: 1. is it possible to provide exponential speedup for performing interpolated-based DRTs or line detection? 2. Considering that the computational basis state of the maximum amplitude is most likely to be observed by quantum measurement, can this property be used to design a better line detection algorithm?
