

The Quantum Realization of Image Linear Gray Enhancement

Kai Liu

Guangxi Normal University

Hai-Sheng Li (✉ lhs_ecjtu@126.com)

Guangxi Normal University

Research Article

Keywords: Gray transformation, Image enhancement, Quantum computation, Quantum circuit, Quantum image processing

Posted Date: July 12th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1817143/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

The Quantum Realization of Image Linear Gray Enhancement

Kai Liu¹ · Hai-Sheng Li¹

Received: date / Accepted: date

Abstract Linear gray enhancement is a spatial domain image enhancement technique commonly used in classical computers, mainly including image negative, image contrast stretching, and piecewise linear gray transformation. In order to realize these three linear gray enhancement techniques in the quantum computers, this paper proposes three types of linear gray transformation schemes for quantum images based on the generalized model of novel enhanced quantum image representation(GNEQR), and the quantum circuits that realize these three transformation methods are constructed according to the schemes. The proposed circuits take advantage of efficient quantum arithmetic operations and parallel Controlled-NOT modules to factor classical transformations into basic unitary operators such as the Controlled-NOT gates and the Toffoli gates. The results show that the linear gray enhancement algorithm for quantum images is better than the classical algorithm in both spatial complexity and time complexity.

Keywords Gray transformation · Image enhancement · Quantum computation · Quantum circuit · Quantum image processing

1 Introduction

The quantum computing model has become a hot topic in recent years, and it was first proposed by the American physicist Feynman in 1982 [1]. The famous

Kai Liu
E-mail: 2210491656@qq.com

✉ Hai-Sheng Li
E-mail: lhs.ecjtu@126.com

¹ College of electronic Engineering, Guangxi Normal University, 541004, Guilin, Guangxi, P.R. China

Moore's Law states that computer performance will double every 2-3 years [2]. However, Moore's Law cannot hold forever with the electronic components cannot shrink indefinitely. The emergence of quantum computation provides the possibility to solve this problem, and its properties such as superposition and entanglement that can make computation faster and more efficient [3]. Researchers have also proposed quantum algorithms that can be used in practice, such as the famous Shor's integer factoring algorithm [4] and Grover's database searching algorithm [5], which further provide strong support for proving that quantum computation has stronger computing power than classical computation. With the rapid development of quantum computation and quantum information, the research on quantum computation has gradually deepened into the image processing [6]. Quantum image processing (QIP) has become an important branch of quantum computation research. QIP focuses on performing classical image processing tasks in quantum computers to improve processing efficiency, especially in large-scale image processing tasks [7]. At present, the two major directions of QIP are quantum image representation and quantum image processing algorithms.

In recent years, many classical models of quantum image representation have been proposed. In 2003, the first quantum image representation model, called the qubit lattice representation, was proposed by Venegas-Andrace [8]. More recently, Le. et al. proposed a flexible representation for quantum images (FRQI) [9] using quantum superposition state to store the colors and the corresponding positions of an image. In 2013, Zhang. et al. proposed a novel enhanced quantum representation model (NEQR) [10] based on FRQI, and years later the generalized model of NEQR (GNEQR) was proposed by Li. et al [11]. Comparing GNEQR with NEQR, the former not only has high storage efficiency in color image representation but also can represent the rectangular image. At the same time, many quantum image processing algorithms based on these representation models have been proposed, such as quantum image geometric transformation [12-14], quantum image scaling [15], quantum image compression [11], quantum image watermarking [16, 17], quantum image scrambling [18-20], quantum image encryption [21], quantum image edge detection [22], quantum image enhancement [23].

Image enhancement is basically improving the interpretability or perception of information in images for human viewers and providing 'better' input for other automated image processing techniques. The principal objective of image enhancement is to modify attributes of an image to make it more suitable for a given task and a specific observer. There exist many techniques that can enhance a digital image without spoiling it. The enhancement methods can broadly be divided into the following two categories: spatial domain methods and frequency domain methods [24].

In spatial domain techniques, we directly deal with the image pixels. The pixel values are manipulated to achieve desired enhancement. On the other hand, frequency based domain image enhancement is a term used to describe the analysis of mathematical functions or signals with respect to frequency and operate directly on the transform coefficients of the image, such as fourier

transform, discrete wavelet transform (DWT), and discrete cosine transform (DCT). Image enhancement techniques like spatial domain methods can again be classified into two broad categories: point processing operation and neighborhood enhancement operation. However, compared with classical image processing, quantum image processing technology is still in infancy, there is not much research about the spatial domain methods in the quantum field. Several detection schemes of quantum image edges based on the Sobel operator [25], Prewitt operator [26], Kirsch operator [27] and Laplacian operator [28] has been proposed in previous years, but these algorithms belong to the category of quantum image neighborhood enhancement operation. Point processing operation is the simplest spatial domain operation as operations are performed on single pixel only. Pixel value of the processed image depend on pixel value of original image. The Point processing approaches can be classified into two broad categories: linear gray transformation and nonlinear gray transformation [29].

Linear gray transformation for quantum image is the focus of our research. This paper proposed three types of linear gray transformation schemes for quantum images based on point processing operation in quantum computers. We provide quantum circuits for realizing image negative, image contrast stretching, and piecewise linear gray transformation by using high-efficiency quantum arithmetic operations. As we know, this work has not been studied yet.

The rest of the paper is organized as follows. A brief background on the GNEQR representation, three types of linear gray transformation, quantum gates and quantum arithmetic operations is presented in Sect. 2. The quantum circuit architecture of these linear gray transformation is discussed in Sect. 3. This is followed in Sect.4 by the theoretical analysis of network complexity. Finally, a conclusion is given in Sect. 5.

2 Preliminaries

2.1 The generalized model of NEQR (GNEQR)

GNEQR is the generalized model of NEQR [11]. For a gray image with the size $2^n \times 2^m$ and gray range 2^q , GNEQR encodes the position information into $(n+m)$ qubits and encodes the gray intensity into q qubits. According to the GNEQR, a quantum image can be written as the form shown below.

$$|I\rangle = \frac{1}{\sqrt{2^{n+m}}} \sum_{X=0}^{2^n-1} \sum_{Y=0}^{2^m-1} |C_{XY}\rangle |X\rangle |Y\rangle, \quad (1)$$

$$|C_{XY}\rangle = |C_{XY}^{q-1} C_{XY}^{q-2} \cdots C_{XY}^0\rangle, C_{XY}^i \in \{0, 1\}, C_{XY} \in \{0, 2^q - 1\}, \quad (2)$$

$$|X\rangle |Y\rangle = |x_{n-1} x_{n-2} \cdots x_0\rangle |y_{m-1} y_{m-2} \cdots y_0\rangle, x_j, y_j \in \{0, 1\}, \quad (3)$$

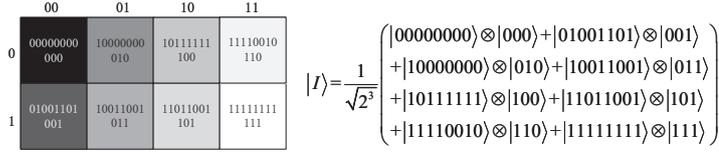


Fig. 1: An example of a $2^1 \times 2^2$ gray image and its GNEQR representation

where the binary sequence $|C_{XY}\rangle$ is the gray value of the corresponding pixel coordinate $|X, Y\rangle$. Fig. 1 shows an example of a $2^n \times 2^m$ gray image and its GNEQR representation.

2.2 Linear gray transformation

Linear gray transformation is to adjust the gray of the target image by establishing gray mapping, and to linearly expand or compress the gray of the image [29]. The gray mapping relationship can be described as follows.

$$O(x, y) = T[I(x, y)], \quad (4)$$

where T is a transformation that maps the original image into the transformed image, (x, y) is the pixel coordinates, $I(x, y)$ is the original image, and $O(x, y)$ is the transformed image. Fig. 2 shows the schematic diagram of mapping the original image into the transformed image. It can be seen from Fig. 2 that the pixel value has changed after the transformation. In classical point processing operation, there are three types of linear gray level transformation: image negative, image contrast stretching and piecewise linear gray transformation. Each gray transformation method has a different adaptation scenario. For different images, it is necessary to use the appropriate transformation method to achieve a better enhancement effect. Fig. 3 shows the principle diagrams of three types of gray transformation, where C_{XY} represents the pixel value of the original image, C'_{XY} represents the pixel value of the transformed image, and the color depth of the image is q bits.

The most basic and simple operation in digital image processing is to compute the negative of an image. The pixel gray values are inverted to compute

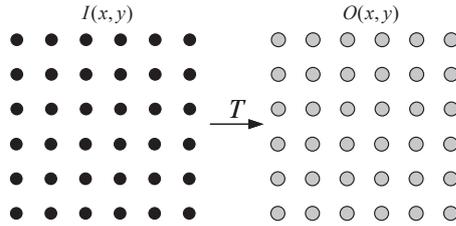


Fig. 2: The image gray level transformation process

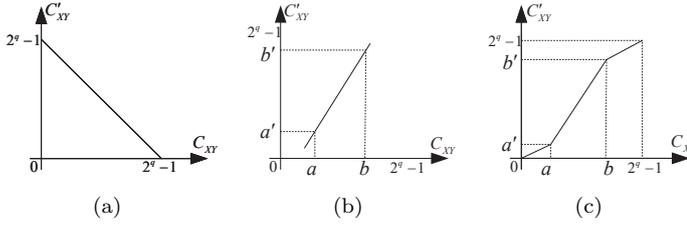


Fig. 3: The principles of three types of linear gray transformations. **a** Image negative. **b** Image contrast stretching. **c** Piecewise linear gray transformation

the negative of an image. For example, if C_{XY} is the gray value of the original image, the gray value C'_{XY} of the negative image can be computed as

$$C'_{XY} = 2^q - 1 - C_{XY}. \quad (5)$$

Fig. 3(a) shows the principle diagram of image negative.

Image contrast stretching is to expand the gray value of all pixels in the same proportion to increase the pixel difference between adjacent pixels and achieve the purpose of contrast enhancement [30]. Suppose the pixel gray value range of the input image $I(x, y)$ is $[a, b]$, the gray value range of the transformed image $O(x, y)$ is $[a', b']$. The image contrast stretching can be described as

$$C'_{YX} = a' + k(C_{YX} - a), \quad (6)$$

$$k = \frac{b' - a'}{b - a}, \quad (7)$$

where k is the gray value stretching coefficient, represents the slope of the oblique line in Fig. 3(b).

If the gray range of the original image is divided into two or more segments for linear transformation, we call it piecewise linear gray transformation. Piecewise linear gray transformation can stretch the gray details of specific regions of the target image and relatively suppress uninteresting regions according to the actual situation [30]. Fig. 3(c) shows the schematic diagram of piecewise linear gray transformation. Suppose the segmentation points are $[a, a']$ and $[b, b']$, linear gray transformation can be described as

$$C'_{XY} = \begin{cases} k_1 C_{XY} & 0 \leq C_{XY} < a \\ k_2 (C_{XY} - a) + a' & a \leq C_{XY} \leq b \\ k_3 (C_{XY} - b') + b' & b < C_{XY} \leq 2^q - 1 \end{cases}, \quad (8)$$

$$k_1 = \frac{a'}{a}, k_2 = \frac{b' - a'}{b - a}, k_3 = \frac{2^{q-1} - b'}{2^{q-1} - b}, \quad (9)$$

where k_1 , k_2 , and k_3 are the slope of the three broken lines in Fig. 3(c).

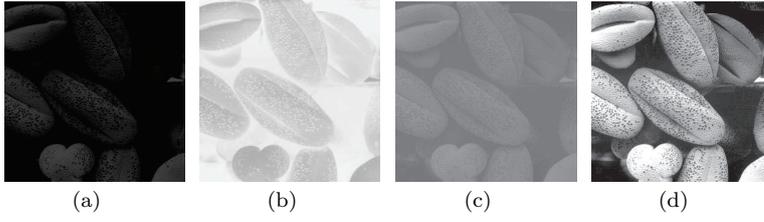


Fig. 4: The transformed images for pollen images. **a** Original image, **b** Negative of the image, **c** Image after contrast stretching, **d** Image after piecewise linear gray transformation.

In order to show the different transformation effect of the three types of linear gray transformations, we selected the pollen image as an example and obtain different transitioned image by using three types of transformations respectively, as shown in Fig. 4.

2.3 Quantum gates

The computational basis states $|0\rangle$, $|1\rangle$, and their dual states $\langle 0|$, $\langle 1|$ can be expressed in the row and column vectors as:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \langle 0| = [1 \ 0], \langle 1| = [0 \ 1].$$

Suppose U is the matrix of an n -qubit gate, n controlled- U gates are presented in Fig. 5, with matrices that can be expressed as:

$$\begin{cases} C_n^j(U_{2^m}) = (|j\rangle\langle j| \otimes U_{2^m}) + \sum_{i=0, i \neq j}^{2^n-1} (|i\rangle\langle i| \otimes I_{2^m}) \\ V_n^j(U_{2^m}) = (U_{2^m} \otimes |j\rangle\langle j|) + \sum_{i=0, i \neq j}^{2^n-1} (I_{2^m} \otimes |i\rangle\langle i|), \end{cases} \quad (10)$$

where $|j\rangle = |j_{n-1}j_{n-2} \cdots j_0\rangle$ and $|i\rangle = |i_{n-1}i_{n-2} \cdots i_0\rangle$. U_{2^m} is the $2^m \times 2^m$ unitary matrix and I_{2^m} is the $2^m \times 2^m$ identity matrix. The circuits of $C_n^j(U_{2^m})$

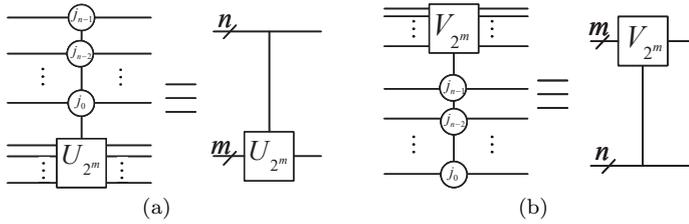


Fig. 5: $C_n^j(U_{2^m})$ gate and $V_n^j(U_{2^m})$ gate

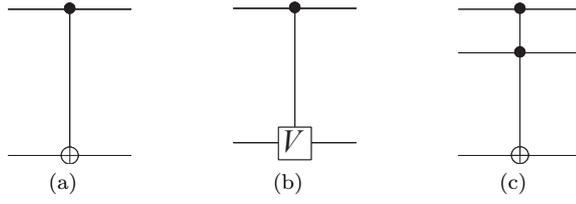


Fig. 6: Representations of some quantum gates, including **a** Controlled-NOT gate (i.e., XOR gate), **b** Controlled-V gate, **c** Toffoli gate

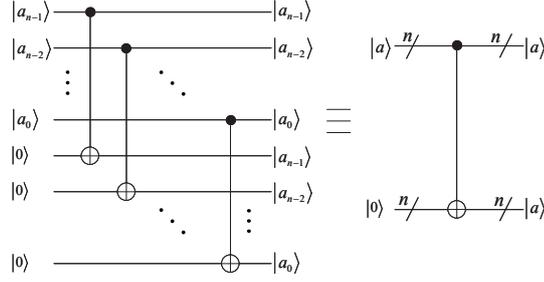


Fig. 7: Quantum circuit for the parallel controlled-NOT module

gate and $V_n^j(U_{2m})$ gate are shown in Fig. 5. For instance, unitary matrices of 1-qubit X and V are

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, V = \frac{1+i}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}, V^\dagger = \frac{1-i}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix},$$

where i is an imaginary unit. Let $n = 1$, $j = 1$, $U_{2m} = X$, $C_1^1(X)$ is the CNOT gate, let $n = 1$, $j = 1$, $U_{2m} = V$, $C_1^1(V)$ is the controlled-V gate, and let $n = 2$, $j = 3$, $U_{2m} = X$, $C_2^3(X)$ is the Toffoli gate. The corresponding controlled gates are presented in Fig. 6.

The parallel controlled-NOT module consists of n CNOT gates, as illustrated in Fig. 7. This module is used to make a copy of n qubit sequence information into the auxiliary qubits $|0\rangle^{\otimes n}$.

And the quantum gates covered in this paper also include

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}, T^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{i\pi}{4}} \end{bmatrix}.$$

In 2009, Thapliyal et al. proposed TR gate, which is constructed by a Toffoli gate, a CNOT gate and two X gates [32]. Amy et al. gave the optimal implementations of Peres gate (PG) in 2013 [31]. Fig. 8 shows the circuit and logical relationship of PG gate and TR gate. In order to further promote the development of quantum image, Li et al. designed fault-tolerant implementations of the TR gate, Peres gate, and variants with better performance than that from the Toffoli gate in 2020. Therefore, based on the fault-tolerant

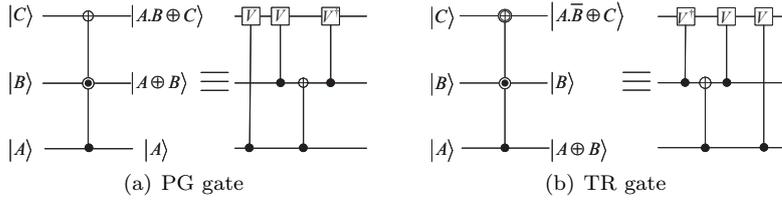


Fig. 8: PG gate and TR gate

implementations of the TR and Peres gates, they implemented fault-tolerant quantum arithmetic operation circuits for quantum image processing by using the TR gate, PG gate and their variants. For a more comprehensive survey of TR1, TR2, PG1, PG2 and other variants of TR and PG gates, readers can be referred to [33].

2.4 Quantum arithmetic operations

From Eqs. (5), (6) and (8), we can see that linear gray transformation mainly use addition operation, subtraction operation, multiplication operation, and comparison operation. Hence, the relevant quantum arithmetic operations are introduced in this section.

2.4.1 Quantum adder and quantum modular subtracter

Li et al. used the PG1 gate and the TR2 gate to design an efficient quantum adder with an auxiliary bit, and the quantum cost of this quantum adder is only $(13n-10)$ [33]. Fig. 9 shows the implementation circuit for the quantum adder. Suppose that $|b\rangle = |b_{n-1}b_{n-2} \cdots b_0\rangle$, $|a\rangle = |a_{n-1}a_{n-2} \cdots a_0\rangle$, and $|s\rangle = |s_n s_{n-1} \cdots s_0\rangle$, the quantum adder implemented the operation as follow:

$$|s\rangle = |b\rangle + |a\rangle. \quad (11)$$

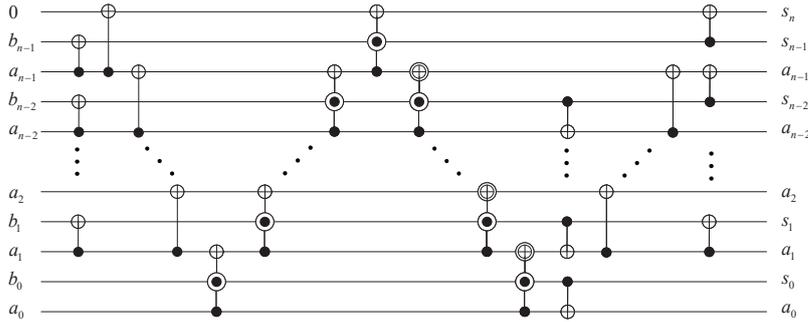


Fig. 9: The implementation circuit of the quantum adder

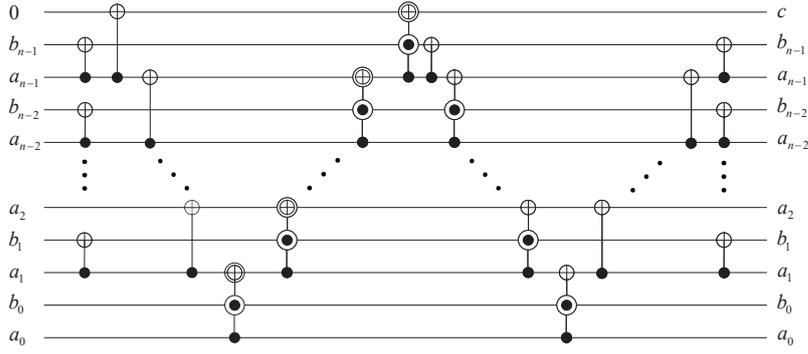


Fig. 10: The implementation circuit of the quantum computer

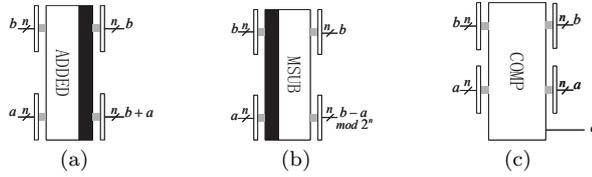


Fig. 11: Quantum arithmetic operation symbol. **a** Quantum adder, **b** Quantum modular subtractor, **c** Quantum comparator

Since the quantum subtractor is not yet mature, the quantum modular subtractor is used instead of the quantum subtractor in the design of related circuits in this paper, this is because subtraction operation is equivalent to modulo subtraction operation without borrow. Ref. [33] notes that quantum modular adder is realized by modifying the circuit of the quantum adder, and by substituting TR1 and PG2 for PG1 and TR2 in the circuit of the quantum modular adder, we can obtain the circuit of the quantum modular subtractor. The quantum modular subtractor implemented the operation as follow:

$$|d\rangle = |b - a\rangle \text{ mod } 2^n, \tag{12}$$

where $|d\rangle = |d_{n-1}d_{n-2} \dots d_0\rangle$. The simplified circuit diagram of the quantum adder and quantum modular subtractor are shown in Fig. 11(b).

2.4.2 Quantum comparator

Comparator is used to compare two positive integers, which occupies an important position in the quantum image processing. Ref. [33] notes that the quantum comparator circuit is implemented using the TR1 and PG1 gates as shown in Fig. 10 and its symbol in Fig. 11(c). The output qubit c is used to

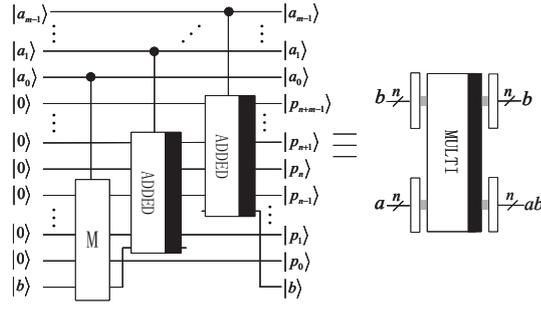


Fig. 12: The implementation circuit of the quantum multiplier

denote the comparison result, i.e.

$$\begin{cases} |a\rangle \geq |b\rangle & |c\rangle = |0\rangle \\ |a\rangle < |b\rangle & |c\rangle = |1\rangle \end{cases} \quad (13)$$

In this paper, it is applied to compare gray values of C_{XY} .

2.4.3 Quantum multiplier

Li et al. designed a controllable adder based on a quantum adder, and then a quantum multiplier can be realized by stacking multiple controlled quantum adders. The circuit of the quantum multiplier is shown in Fig. 12. The multiplier implemented the following:

$$|p\rangle = |a\rangle \times |b\rangle \quad (14)$$

where $|a\rangle = |a_{m-1}a_{n-2} \cdots a_0\rangle$, $|b\rangle = |b_{n-1}b_{n-2} \cdots b_0\rangle$ and $|p\rangle = |p_{m+n-1} \cdots p_1p_0\rangle$.

3 Quantum circuit architecture of image gray linear transformation

3.1 Image linear gray transformation's GNEQR representation

In this paper, we use GNEQR to represent quantum images. Because gray transformation is the operation which focuses on manipulating the gray value of every pixel in the images, we only need to change the gray value C_{XY} in Eq. (1). We define the gray negative operation, the contrast stretching operation and the piecewise linear transformation operation as T_1 , T_2 , and T_3 , respectively. Assume I is the original image with the size $2^n \times 2^m$ and gray range 2^q , O is the transformed image. C_{XY} represents the pixel of the input

image and C'_{XY} represents the pixel of the transformed image. The operation T which on GNEQR quantum images can be defined as

$$O = T(I) = \frac{1}{\sqrt{2^{n+m}}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^m-1} T(|C_{XY}\rangle)|X\rangle|Y\rangle.$$

According to Eq. (5), we define $|C'_{XY}\rangle$ as

$$|C'_{XY}\rangle = T_1(|C_{XY}\rangle),$$

where

$$|C'_{XY}\rangle_{T_1} = |2^q - 1 - C_{XY}\rangle. \quad (15)$$

Likewise, image contrast stretching T_2 and piecewise linear gray transformation T_3 can be described as

$$|C'_{XY}\rangle = T_2(|C_{XY}\rangle),$$

where

$$|C'_{XY}\rangle_{T_2} = |a'\rangle + |k\rangle(|C_{XY} - a\rangle), \quad (16)$$

$$|k\rangle = \left| \frac{b' - a'}{b - a} \right\rangle, \quad (17)$$

and

$$|C'_{XY}\rangle = T_3(|C_{XY}\rangle),$$

where

$$|C'_{XY}\rangle_{T_3} = \begin{cases} |k_1\rangle|C_{XY}\rangle & 0 \leq C_{XY} < a \\ |k_2\rangle(|C_{XY} - a\rangle) + |a'\rangle & a \leq C_{XY} \leq b \\ |k_3\rangle(|C_{XY} - b'\rangle) + |b'\rangle & b < C_{XY} \leq 2^q - 1 \end{cases}, \quad (18)$$

$$|k_1\rangle = \left| \frac{a'}{a} \right\rangle, |k_2\rangle = \left| \frac{b' - a'}{b - a} \right\rangle, |k_3\rangle = \left| \frac{2^{q-1} - b'}{2^{q-1} - b} \right\rangle. \quad (19)$$

Eqs (15), (16), and (18) give three types of linear gray transformation's quantum representation, respectively. In the following, the circuits we will give are based on them.

3.2 Quantum circuit architecture of linear gray transformation

Because the operation T_1 , T_2 , and T_3 are independent of each other as shown in Eqs. (15-18), we will give several circuits to realize them, respectively.

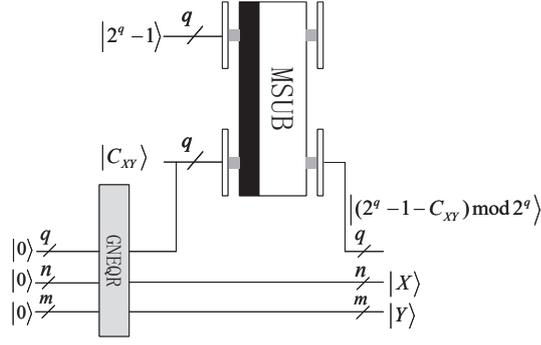


Fig. 13: Quantum image negative network

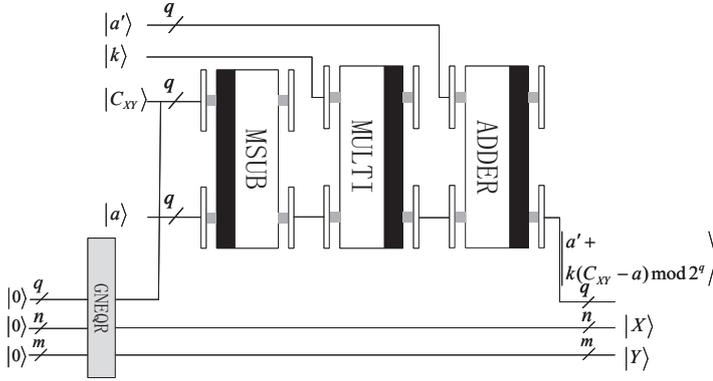


Fig. 14: Image contrast stretching network

3.2.1 Image negative network

The image negative network that realizes T_1 . We use the quantum modular subtractor to replace the subtractor to construct the network structure. By contrasting Eq. (15) with Eq. (12), we can replace b , a in (12) with $2^q - 1$, C_{XY} , respectively.

$$|2^q - 1, C_{XY}\rangle \rightarrow |2^q - 1, (2^q - 1 - C_{XY}) \bmod 2^q\rangle. \quad (20)$$

The quantum image gray negative network is shown in Fig. 13. The final output of the network is the gray information $|C'_{XY}\rangle_{T_1}$ of negative image.

3.2.2 Quantum image contrast stretching network

According to Eq. (16), we will use quantum modular subtractor, quantum multiplier, and quantum adder to design the image contrast stretching network

according to the following three steps:

$$\begin{aligned} |C_{XY}, a\rangle &\rightarrow |C_{XY}, (C_{XY} - a) \bmod 2^q\rangle \\ |k, (C_{XY} - a) \bmod 2^q\rangle &\rightarrow |k, k(C_{XY} - a) \bmod 2^q\rangle \\ |a', k(C_{XY} - a) \bmod 2^q\rangle &\rightarrow |a', a' + k(C_{XY} - a) \bmod 2^q\rangle. \end{aligned} \quad (21)$$

The first step corresponds to the quantum subtracter modulo 2^q , the second one corresponds to quantum comparator and the third one corresponds to the quantum adder. We can cascade the three types of arithmetic operations to realize T_2 as shown in Fig. 14. The output of the whole network, $(a' + k(C_{XY} - a) \bmod 2^q)$, is the gray information $|C'_{XY}\rangle_{T_2}$ of the transformed image.

3.2.3 Piecewise linear gray transformation network

Unlike image negative network and image image contrast stretching network, piecewise linear gray transformation network is more complex and requires more arithmetic operations. In order to show a clear details of this network, the whole workflow of our scheme, as well as the quantum circuit realization, will be discussed in this subsection.

1. Workflow of piecewise linear gray transformation

The flow chart is shown in Fig. 15, which is divided into 5 stages more specifically. Next we will give the details of our proposed network.

2. Quantum circuit realization of piecewise linear gray transformation

According to the flow chart, Fig. 16 gives the piecewise linear gray transformation network. The following five steps will be described the network's details. The five steps are the preparation of quantum state, copying the pixel information, dividing pixel values into three intervals, performing operation and obtaining the transformed image.

Step 1. Preparation

Input the gray image into a quantum computer and we use GNEQR to store the image $|I\rangle$. Hence, the quantum state is as follows:

$$|I\rangle = \frac{1}{\sqrt{2^{n+m}}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^m-1} |C_{XY}\rangle |X\rangle |Y\rangle. \quad (22)$$

Step 2. Auxiliary qubits store the pixel information

In this step we first prepare three $|0\rangle^{\otimes q}$ auxiliary qubits. Two of them are used to store the pixel value information $|C_{XY}\rangle$ of the original image $|I\rangle$ by using the parallel controlled-NOT modules, and the remaining one is used to store the counterpart of the output image $|O\rangle$. This step can be described as

$$|I\rangle |0\rangle^{\otimes q} |0\rangle^{\otimes q} |0\rangle^{\otimes q} \rightarrow \frac{1}{\sqrt{2^{n+m}}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^m-1} |C_{XY}\rangle |C_{XY}\rangle |C_{XY}\rangle |0\rangle |X\rangle |Y\rangle. \quad (23)$$

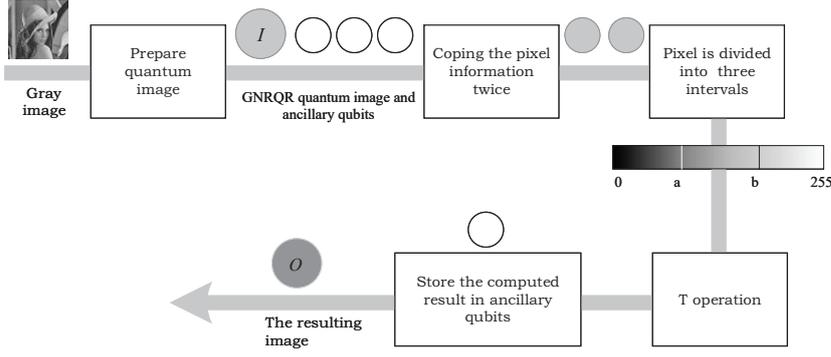


Fig. 15: The Workflow of piecewise linear gray transformation for quantum image

Step 3. Divide pixel values into three intervals

According to the principle of piecewise linear gray transformation in Fig. 3(c), this step we need to use two quantum comparators to divide the pixel value into three intervals. For the convenience of the following description, we mark the comparator at the top of Step3 as COMP1 and the bottom one as COMP2. Use $|C_{XY}\rangle$ and $|a\rangle$ as the input for COMP1 and $|c_1\rangle$ is its output; According to the Eq. (13), we can draw the following conclusions. For COMP1,

$$\begin{cases} |C_{XY}\rangle \geq |a\rangle & |c_1\rangle = |0\rangle, \\ |C_{XY}\rangle < |a\rangle & |c_1\rangle = |1\rangle. \end{cases} \quad (24)$$

For COMP2,

$$\begin{cases} |C_{XY}\rangle \geq |b\rangle & |c_2\rangle = |0\rangle, \\ |C_{XY}\rangle < |b\rangle & |c_2\rangle = |1\rangle. \end{cases} \quad (25)$$

As we know $|0\rangle \leq |C_{XY}\rangle \leq |2^q - 1\rangle$. From Eqs. (26) and (27), $|C_{XY}\rangle$ can be described as

$$\begin{cases} |0\rangle \leq |C_{XY}\rangle < |a\rangle & |c_1\rangle = |1\rangle, \\ |a\rangle \leq |C_{XY}\rangle < |b\rangle & |c_1\rangle|c_2\rangle = |0\rangle|1\rangle, \\ |b\rangle \leq |C_{XY}\rangle \leq |2^q - 1\rangle & |c_2\rangle = |0\rangle. \end{cases} \quad (26)$$

Finally, we divide the pixel value into three intervals by judging the output of the two comparators.

Step 4. Perform operation T_3

If $|c_1\rangle = |1\rangle$, then the current pixel value interval is $|0\rangle \leq |C_{XY}\rangle \leq |a\rangle$. Use quantum multiplier to expand the gray value of this interval by k times, and $k_1 = \frac{a'}{a}$,

$$|k_1, C_{XY}\rangle \rightarrow |k_1, k_1 C_{XY}\rangle. \quad (27)$$

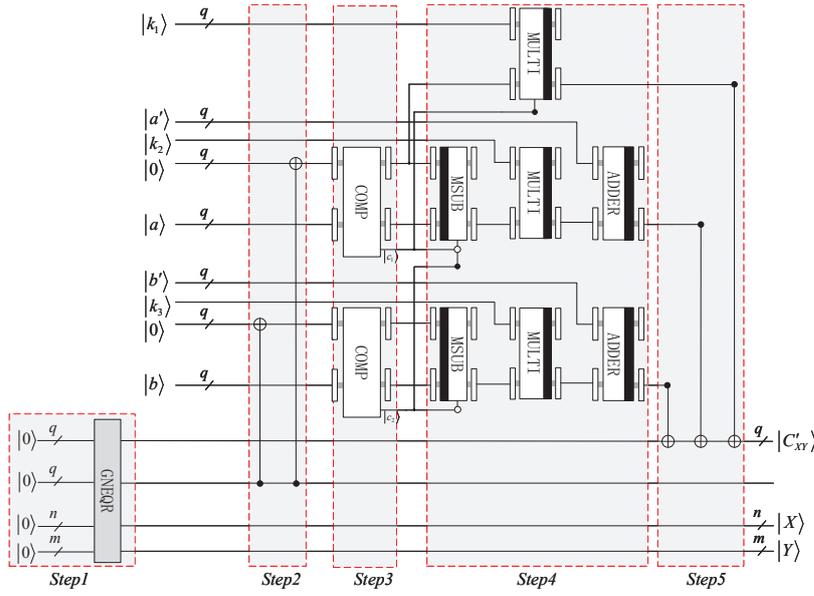


Fig. 16: Piecewise linear gray transformation's quantum network

If $|c_1\rangle|c_2\rangle = |0\rangle|1\rangle$, then the current pixel value interval is $|a\rangle < |C_{XY}\rangle \leq |b\rangle$. Use quantum modulo subtractor, adder and multiplier to realize the following operations on the gray value of this interval, where $k_2 = \frac{b'-a'}{b-a}$.

$$\begin{aligned}
 |C_{XY}, a\rangle &\rightarrow |C_{XY}, (C_{XY} - a) \bmod 2^q\rangle \\
 |k_2, (C_{XY} - a) \bmod 2^q\rangle &\rightarrow |k_2, k_2(C_{XY} - a) \bmod 2^q\rangle \\
 |a', k_2(C_{XY} - a) \bmod 2^q\rangle &\rightarrow |a', a' + k_2(C_{XY} - a) \bmod 2^q\rangle.
 \end{aligned} \tag{28}$$

The operation process is the same as the process of Eq. (21).

If $|c_2\rangle = |0\rangle$, then the current pixel interval is $|b\rangle \leq |C_{XY}\rangle \leq |2^q - 1\rangle$, $k_3 = \frac{2^q - 1 - b'}{2^q - 1 - b}$. The operation is as follows:

$$\begin{aligned}
 |C_{XY}, b\rangle &\rightarrow |C_{XY}, (C_{XY} - b) \bmod 2^q\rangle \\
 |k_3, (C_{XY} - b) \bmod 2^q\rangle &\rightarrow |k_3, k_3(C_{XY} - b) \bmod 2^q\rangle \\
 |b', k_3(C_{XY} - b) \bmod 2^q\rangle &\rightarrow |b', b' + k_3(C_{XY} - b) \bmod 2^q\rangle,
 \end{aligned} \tag{29}$$

the operation process is the same as that of Eq. (21).

Step 5. Get transformed image The results of linear gray transformation in step 4 are stored in the prepared state again by using the parallel controlled-NOT module. Ignoring the garbage output, we will get the resulting quantum image.

$$|O\rangle = \frac{1}{\sqrt{2^{n+m}}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^m-1} |C'_{XY}\rangle_{T_3} |X\rangle |Y\rangle. \tag{30}$$

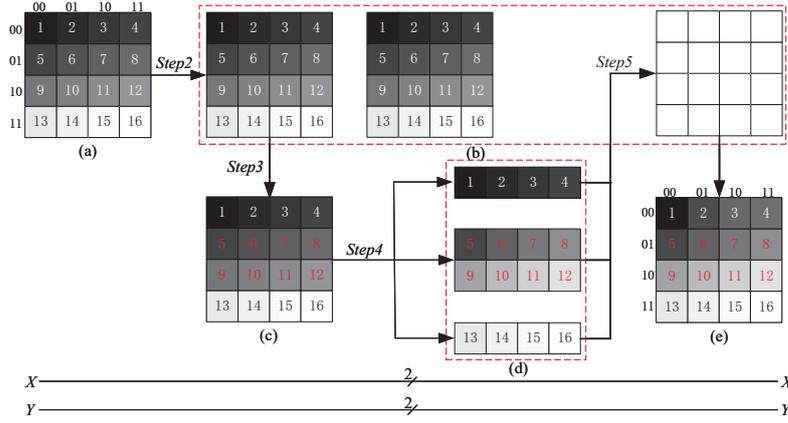


Fig. 17: Piecewise linear gray transformation's process of the sample image.

Table 1: Gray information variation table of the sample image

The ordinal image $I(x, y)$			The resulting image $O(x, y)$		
Symbol	$C_7 \sim C_0$	Pixel	Symbol	$C'_7 \sim C'_0$	Pixel
1	00000000	0	1	00000000	0
2	00001010	10	2	00001010	10
3	00010100	20	3	00010100	20
4	00011110	30	4	00011110	30
5	00100000	32	5	00100000	34
6	00101110	46	6	00101110	62
7	00111100	60	7	00111100	90
8	01001010	74	8	01011000	118
9	01011000	88	9	01011000	146
10	01100110	102	10	01100110	174
11	01110100	116	11	01110100	202
12	01111111	127	12	01111111	224
13	11100110	230	13	11100110	230
14	11110000	240	14	11110000	240
15	11111010	250	15	11111010	250
16	11111111	255	16	11111111	255

3.3 A simple example

Let us consider the simple $2^2 \times 2^2$ GNEQR image with the gray range 2^q shown in Fig. 17(a) as an example. We first assign an initial value to the gray of each pixel and mark each pixel with the number 1-16. Here we regard the gray reversal circuit and the image contrast stretching circuit as part of the

piecewise gray linear transformation circuit. Therefore, we only apply piecewise linear gray transformation to sample image in this section.

According to the network, the change of gray information of the sample image is shown in Table 1. After the quantum image state is prepared, we prepare two $|0\rangle^{\otimes q}$ to store the pixel information of the original image and one $|0\rangle^{\otimes q}$ to store the counterpart of the resulting image in step 2, which is shown in Fig. 16(a). In step 3, we divide the gray value into three intervals, and then each interval is expanded or compressed by a different proportion. Due to the current limitation of quantum floating-point arithmetic operation, gray expansion or compression coefficient k need to be set to integer. We set the two points to (30, 30) and (127, 224), that is to say $a = 30$, $a' = 30$, $b = 127$, $b' = 224$, the 16 pixels are divided into three parts: (1-4), (5-12) and (13-16) by this way. According to Eq. (21), the slopes of the three broken lines in Fig. 3(c) are as follows.

$$k_1 = \frac{a'}{a}=1, k_2 = \frac{b' - a'}{b - a}=2, k_3 = \frac{2^{q-1} - b'}{2^{q-1} - b}=1. \quad (31)$$

In step 4, we perform linear gray transformation on the pixels of the three intervals according to Eqs. (27-29). According to the gray information after transformation in Table 1, the results of piecewise linear gray transformation of the image are shown in Fig. 17(d). Finally, we store the final result in auxiliary qubit, and Figure 17(e) shows the final resulting image. Compared with the original image, we can find that the gray contrast of pixels (5-12) of the resulting image is stretched, and the gray contrast of pixels (1-4) and (13-16) remain unchanged.

4 Network complexity

The quantum image linear gray enhancement scheme relies on the parallelism of quantum computing, which not only makes it better than the classical method in terms of storage space, but also is more efficient in computing efficiency. In this section, we analyze the complexity of the three types linear gray enhancement schemes designed in this paper from two aspects of space and time.

4.1 Spatial complexity

The space complexity means the memory cells an image needs when it is stored in a computer. In a classical computer, a $2^n \times 2^m$ image with 2^q gray range needs $2^n \times 2^m \times q$ bits because it has $2^n \times 2^m$ pixels and each pixel needs q bits to represent. In a quantum computer, according to GNEQR, the same image only needs $n + m + q$ qubits. Table 2 gives three common images as examples. After comparison, we find that the space complexity of storing images in a quantum computer is significantly lower than that in a classical computer.

Table 3: The quantum cost of module

No.	Module	Quantum cost
(1)	Parallel CNOT module	q
(2)	Quantum adder	$13q - 10$
(3)	Quantum comparator	$12q - 8$
(4)	Quantum modular subtractor	$13q - 22$
(5)	Quantum multiplier	$17q^2 - 12q$

Table 2: Space complexity of an image

n	m	q	Bits needed in classical computers $2^n \times 2^m \times q$	Qubits needed in quantum computers $n + m + q$
6	5	8	16384	61
8	8	8	524288	66
10	9	8	4194304	69

4.2 Time complexity

Time complexity means the time an algorithm consumes when it is executing. In a classical computer, the time complexity depends on the number of operation steps to execute the algorithm, while in a quantum computer, the complexity of the quantum circuit network is determined by the number of elementary quantum gates, which is called the quantum cost. Assuming that the original image is a $2^n \times 2^m$ image with 2^q gray range, then we discuss its time complexity in a classical computer and a quantum computer, respectively.

In classical computers, linear gray transformations require operating on each pixel to change the gray value. For each pixel in the image, one modular subtraction operation needs to be completed in image negative; modular subtraction, modular multiplication and modular addition need to be completed in image contrast stretching; and three gray information copying operations, one modular subtraction operation, one multiplication operation and one addition operation in piecewise linear gray enhancement. Therefore, in classical computers, the complexity of the three linear gray transformations is 2^{n+m} , $3 \times 2^{n+m}$, and $6 \times 2^{n+m}$.

In a quantum computer, the complexity of the basic quantum gate is 1, including the X gate, the CNOT gate, and any 2×2 unitary operator. Here we take the CNOT gate as the basic unit for evaluating the circuit complexity. The circuit complexity designed in this paper mainly depends on the quantum arithmetic operation. Ref. [28] gives the quantum cost of each operator, we list them in Table 3. According to the circuit of Fig. 13, Fig. 14 and Fig. 16, we can obtain the number of operators in each circuit. The quantum circuit for image gray negative contains a quantum modular subtractor, so its complexity is $(13q - 22)$. The quantum circuit of image contrast stretching contains 1

Table 4: The time complexity comparison between the classical algorithm and the quantum algorithm

n	m	q	Image negative		Image contrast stretching		Piecewise linear gray transformation	
			Classical	Quantum	Classical	Quantum	Classical	Quantum
6	5	8	2048	82	6144	1170	12288	3730
8	8	8	65536	82	196608	1170	393216	3730
10	9	8	524288	82	1572864	1170	3145728	3730

quantum adder, 1 quantum modular subtractor and 1 quantum modular multiplier. so its complexity is $(13q - 10 + 13q - 22 + 17q^2 - 12q) = (17q^2 + 13q - 32)$; The quantum circuit of piecewise linear gray transformation includes 5 parallel CNOT gate modules, 2 modular subtractors, 2 adders, 3 multipliers and 2 comparators, so its complexity is $\{5q + 2(13q - 22) + 2(13q - 22) + 2(12q - 8) + 3(17q^2 - 12q)\} = (54q^2 + 45q - 86)$. According to the calculation complexity method given above, Table 4 gives three examples to further show the superiority of the processing speed of the quantum image gray enhancement algorithm.

5 Conclusion

Based on the GNEQR quantum image, this paper proposes three types of linear gray transformation schemes and gives the corresponding quantum circuit. The complexity analysis shows that the quantum linear gray enhancement algorithm has better performance than the classical algorithm in both space complexity and time complexity, especially in large-scale image processing tasks, the efficiency advantage is more obvious. At the same time, the realization of quantum image linear gray transformation will also become the basis for other quantum image processing, such as quantum image segmentation, quantum image feature extraction, etc.

Due to the limitation of quantum transcendental function operations, the design of quantum image nonlinear transformation has not been studied yet. In the future research work, we will try to study the design of quantum transcendental function operator, which will also lay the foundation for the realization of nonlinear gray enhancement transformation of quantum images.

Acknowledgements This work is supported by the Science and Technology Project of Guangxi under grant no. 2020GXNSFDA238023.

References

1. Feynman, R.P.: Simulating physics with computers. Int. J. Theor. Phys. 21(6), 467-488 (1982)

2. Nagy, M., Akl, S.G.: Quantum computation and quantum information. *Int. J. Parallel, Emergent Distrib. Syst.* 21(1), 1-59 (2006)
3. Kanamori, Y., Yoo, S.M., Pan, W.D., Sheldon, F.T.: A short survey on quantum computers. *Int. J. Comput. Appl.* 28(3), 227-233 (2006)
4. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124-134 (1994)
5. Grover, L.: A fast quantum mechanical algorithm for database search. In: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 212-219 (1996)
6. Beach, G., Lomont, C., Cohen, C.: Quantum image processing (QuIP). In: *Proceedings of the 32nd IEEE Conference on Applied Imagery and Pattern Recognition*, pp. 39-44, Bellingham (2003)
7. Ilyasu, A.M.: Towards realising secure and efficient image and video processing applications on quantum computers. *Entropy*, 15(8), 2874-2974 (2013)
8. Venegas-Andraca, S., Bose, S.: Storing, processing, and retrieving an image using quantum mechanics. In: *Proceedings of SPIE Conference of Quantum Information and Computation*, vol. 5105, pp. 134-147 (2003)
9. Le, P.Q., Dong, F., Hirota, K.: A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* 10(1), 63-84 (2011)
10. Zhang, Y., Lu, K., Gao, Y., Wang, M.: NEQR: a novel enhanced quantum representation of digital images. *Quantum Inf. Process.* 12(8), 2833-2860 (2013)
11. Li, H.S., Fan, P., Xia, H. Y., Peng, H., and Song, S.: Quantum implementation circuits of quantum signal representation and type conversion. *IEEE Trans. Circuits Syst. I Regul. Pap.* 66(1), 341-354 (2019)
12. Le, P.Q., Ilyasu, A.M., Dong, F., Hirota, K.: Fast geometric transformations on quantum images. *Int. J. Appl. Math.* 40(3), 113-123 (2011)
13. Fan, P., Zhou, R.G., Jing, N., Li, H.S.: Geometric transformations of multidimensional color images based on NASS. *Inf. Sci.* 340, 191-208 (2016)
14. Wang, J., Jiang, N., Wang, L.: Quantum image translation. *Quantum Inf. Process.* 14(5), 1589-1604 (2015)
15. Jiang, N., Wang, L.: Quantum image scaling using nearest neighbor interpolation. *Quantum Inf. Process.* 14(5), 1559-1571 (2015)
16. Mogos, G.: Hiding data in a qimage file. *Lect. Notes Eng. Comput. Sci.* 2174(1), 448-452 (2009)
17. Ilyasu, A.M., Le, P.Q., Dong, F., Hirota, K.: Watermarking and authentication of quantum images based on restricted geometric transformations. *Inf. Sci.* 186(1), 126-149 (2012)
18. Li, H.S., Chen, X., Song, S., Liao, Z., Fang, J.: A block-based quantum image scrambling for GNEQR. *IEEE Access.* 7, 138233-138243 (2019)
19. Jiang, N., Wu, W.Y., Wang, L.: The quantum realization of Arnold and Fibonacci image scrambling. *Quantum Inf. Process.* 13(5), 1223-1236 (2014)
20. Zhou, R.G., Sun, Y.J., Fan, P.: Quantum image Gray-code and bit-plane scrambling. *Quantum Inf. Process.* 14(5), 1717-1734 (2015)
21. Zhou, R.G., Wu, Q., Zhang, M.Q., Shen, C.Y.: Quantum image encryption and decryption algorithms based on quantum image geometric transformations. *Int. J. Theor. Phys.* 52(6), 1802-1817 (2013)
22. Fu, X., Ding, M., Sun, Y., Chen, S.: A new quantum edge detection algorithm for medical images. *Proc. SPIE Int. Soc. Opt. Eng.* 7497, 749724 (2009)
23. Yuan, S., Mao, X., Zhou, J., Wang, X.: Quantum image filtering in the spatial domain. *Int. J. Theor. Phys.* 56(8), 2495-2511 (2017)
24. Maini, R., Aggarwal, H.: A comprehensive review of image enhancement techniques. *arXiv preprint arXiv:1003.4053.* (2010)
25. Fan, P., Zhou, R.G., Hu, W., Jing, N.: Quantum image edge extraction based on classical Sobel operator for NEQR. *Quantum Inf. Process.* 18(1), 1-23 (2019)
26. Zhou, R.G., Yu, H., Cheng, Y., Li, F.X.: Quantum image edge extraction based on improved Prewitt operator. *Quantum Inf. Process.* 18(9), 1-24 (2019)
27. Xu, P., He, Z., Qiu, T., Ma, H.: Quantum image processing algorithm using edge extraction based on Kirsch operator. *Opt. Express*, 28(9), 12508-12517 (2020)

28. Fan, P., Zhou, R.G., Hu, W.W., Jing, N.: Quantum image edge extraction based on Laplacian operator and zero-cross method. *Quantum Inf. Process*, 18(1), 1-23 (2019).
29. Gonzalez, R.C., Woods, R.E., Masters, B.R.: *Digital image processing*, third edition. Prentice-Hall, Inc. 128-130 2007
30. Singh, G., Mittal, A.: Various image enhancement techniques-a critical review. *IJISR*. 10(2), 267-274 (2014)
31. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE T. COMPUT. AID. D.*, 32(6), 818-830 (2013)
32. Thapliyal, H., Ranganathan, N.: Design of efficient reversible binary subtractors based on a new reversible gate. In 2009 IEEE computer society annual symposium on VLSI (pp. 229-234). IEEE. (2009).
33. Li, H.S., Fan, P., Xia, H., Peng, H., Long, G.L.: Efficient quantum arithmetic operation circuits for quantum image processing. *Sci. China-Phys. Mech. Astron.* 63(8), 280311 (2020)