Quantum Information and Computation, Vol. 23, No. 1&2 (2023) 0045–0066


# QUANTUM IMAGE K-NEAREST NEIGHBOR MEAN FILTERING

JINGKE XI
*Computer Science and Technology, China University of Mining and Technology, No. 1 University Road Xuzhou, Jiangsu Province, China.*

SHUKUN RAN
*Computer Science and Technology, China University of Mining and Technology Xuzhou, Jiangsu Province, China.*

KAI XU
*Computer Science and Technology, China University of Mining and Technology Xuzhou, Jiangsu Province, China.*

Received April 22, 2022
Revised December 8, 2022

Quantum image filtering is an extension of classical image filtering algorithms, which mainly studies image filtering models based on quantum characteristics. The existing quantum image filtering focuses on noise detection and noise suppression, ignoring the effect of filtering on image boundaries. In this paper, a new quantum image filtering algorithm is proposed to realize the K-nearest neighbor mean filtering task, which can achieve the purpose of boundary preservation while suppressing noise. The main work includes: a new quantum compute module for calculating the absolute value of the difference between two non-negative integers is proposed, thus constructing the quantum circuit of the distance calculation module for calculating the grayscale distance between the neighborhood pixels and the center pixel; the existing quantum sorting module is improved to sort the neighborhood pixels with the distance as the sorting condition, and thus the quantum circuit of the K-nearest neighbor extraction module is constructed; the quantum circuit of the K-nearest neighbor mean calculation module is designed to calculate the gray mean of the selected neighbor pixels; finally, a complete quantum circuit of the proposed quantum image filtering algorithm is constructed, and carried out the image de-noising simulation experiment. The relevant experimental indicators show that the quantum image K-nearest neighbor mean filtering algorithm has the same effect on image noise suppression as the classical K-nearest neighbor mean filtering algorithm, but the time complexity of this method is reduced from $O\left(2^{2n}\right)$ of the classical algorithm to $O\left(n^2+q^2\right)$.

*Keywords*: Quantum image filtering, Noise suppression, Boundary preservation, Quantum circuit design

## 1 Introduction

Quantum image processing(QIP) is an emerging interdisciplinary subject that integrates quantum computing and image processing. The main research contents include quantum image representation and quantum image processing algorithms. Quantum image representation focuses on the quantum encoding of classical images using quantum properties, which are mainly divided into the following two categories: (1)Encoding images using the angle parameters of quantum bits, including Qubit Lattice [1], FRQI [2], or QSMC&QSNC [3], etc. This kind of method is a probability model and has certain defects, so these methods cannot retrieve quantum images accurately. (2) Encoding images using the basis state sequence of qubits, including CQIR [4], NEQR [5], NCQI [6], etc. These models are widely used because they can make quantum image retrieval more accurate and they can perform more complex operations on images. Quantum image processing algorithms mainly study the processing of quantum images based on different quantum image representation models, such as quantum image geometric transformation [7], quantum image segmentation [8], quantum image watermarking [9], quantum image feature extraction [10], quantum image scaling [11], quantum image encryption [12] and quantum image filtering [13], etc.

As an effective method to suppress image noise, image filtering is a significant preprocessing step in image processing algorithms such as feature extraction, image segmentation, and edge detection. Quantum image filtering is an extension of the classical image filtering algorithm. It mainly studies the filtering algorithm based on quantum images and realizes the acceleration of the classical image filtering algorithm.

Existing quantum image filtering, such as quantum mean filtering, quantum weighted mean filtering, and quantum median filtering, mainly studies the quantum realization of classical linear filtering and nonlinear filtering algorithms. It focuses on noise detection and noise suppression, ignoring the effect of filtering on image boundaries. Boundary-preserving filters can consider noise suppression and boundary preservation simultaneously, such as K-nearest neighbor mean filtering. But no corresponding quantum filtering has been proposed. In this paper, a quantum image K-nearest neighbor mean filtering algorithm is proposed, which aims to use quantum computing characteristics to process the image K-nearest neighbor mean filtering task in parallel and reduce the time complexity. We focused on the detailed quantum circuit design of the four functional modules of the quantum image K-nearest neighbor mean filtering algorithm (Neighborhood Pixel Preparation, Distance Calculation, K-nearest Neighbor Extraction and K-nearest Neighbor Mean Calculation). In addition, a quantum implementation method for calculating the difference between grayscale values is proposed and applied to the construction of the Distance Calculation module. Furthermore, an improved Sort module is also proposed and applied to the construction of the K-nearest Neighbor Extraction module.

The organization structure of this paper is as follows: Section 2 introduced the quantum image representation model NEQR, K-nearest neighbor mean filtering, and five basic quantum computing modules; In Section 3, the quantum circuits of Distance Calculation, K-nearest Neighbor Extraction and K-nearest Neighbor Mean Calculation is designed, and the complete quantum circuit of the proposed algorithm is constructed in detail; Section 4 analyzed the time complexity of quantum image K-nearest neighbor mean filtering which reflects the acceleration performance of quantum image filtering; In Section 5, the effectiveness of K-nearest neighbor mean filtering of the quantum image is verified by simulation experiments; Section 6 summarized the work of this paper.

## 2 Related works

### 2.1 Quantum filter in QIP

These quantum filters work mainly on various quantum images stored on a quantum computer. In the frequency domain, [13] and [14] proposed two filtering algorithms based on quantum Fourier transform, respectively; the former works on quantum grayscale images, and the latter works on quantum color images. In the spatial domain, [15] and [16] proposed a general quantum image linear filtering model, which can realize the linear filtering of different masks. [17] carried out a quantum implementation of a Gaussian filtering process in weighted mean filtering but did not use quantum superposition states to process the filtering task in parallel. [18], [19] and [20] mainly designed quantum circuits around nonlinear median filtering. The former mainly focused on realizing median filtering, while the latter two improved it to enhance the filtering effect by adding different noise detection modules. Moreover, [21] proposed a quantum image midpoint filter that integrates nonlinear and linear filtering, which has a good suppression effect on Gaussian noise.

### 2.2 Quantum filter in QNN

The quantum filter in the field of quantum machine learning is mainly used to replace the convolution layer in the classical neural network. This quantum-classical hybrid computing method is used to accelerate the classical machine learning algorithm. In [22], A new type of quantum convolution conversion layer is proposed, which uses a fixed quantum filter to replace the classical filter, and has higher test accuracy. In addition, [23] proposed a new type of quantum variational filter. The filter only needs to replace the scalar product with the quantum product, which can be combined with the classical neural network to achieve the purpose of training the neural network on the quantum computer. Compared with the fixed filter of the former, the variational filter can perform training similar to the classical neural network in the hybrid quantum neural network, expanding the trainability of quantum machine learning.

### 2.3 The NEQR image representation model

Zhang proposed a NEQR image representation model in [5]. The NEQR model uses the qubit basis sequence to store the grayscale value and position information. Compared to the use of amplitude angle storage grayscale information, the NEQR model can easily make more quantum image processing algorithms and can be accurately retrieved. The NEQR model can be represented by Eq. (1) for an image having a size of $2^n \times 2^n$ and the grayscale range $[0, 2^{q-1}]$.

$$|I\rangle = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}|C_{YX}\rangle\,|Y\rangle|X\rangle = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\otimes_{i=0}^{q-1}\left|C^i_{(Y,X)}\right\rangle|Y\rangle|X\rangle \tag{1}$$

In Eq. (1), $\left|C_{(Y,X)}\right\rangle$ represent the grayscale value of images, and $|Y\rangle|X\rangle$ represent the pixel position information in vertical and horizontal directions, respectively. Fig. 1 gives an example of the NEQR representation of a $2 \times 2$ grayscale image.

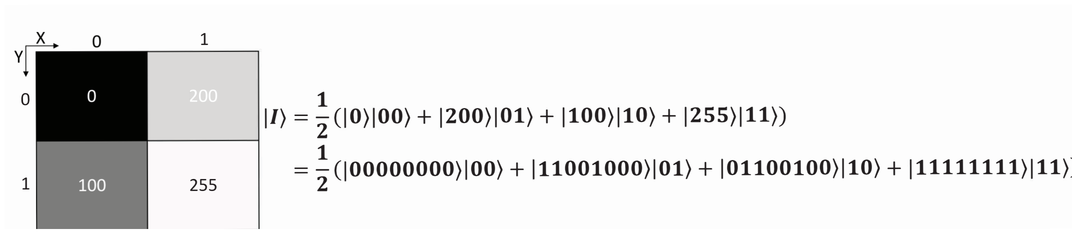


Fig. 1. A 2脳2 grayscale image and its NEQR representation

### *2.4 K-nearest neighbor mean filtering*

K-nearest neighbor mean filtering is a representative boundary preserving filtering. For each filter window, it identifies the K pixels closest to the grayscale value of the processing pixel and replaces the original pixel with the average of K pixels. If the processing pixel is a non-noise point, the grayscale value of the same region can be obtained by calculating the grayscale value mean of the K-nearest pixels, which maintains the pixel definition. If the processing pixel is a noise point, the smoothing process can suppress noise because of its relatively isolated state from the surrounding pixels. In addition, the K-nearest neighbor mean filtering has the advantage of statistical sorting filters and linear filters, which has a good suppression effect on multiple noises, suitable for image denoising in the case of mixing noise.

The key to the boundary preserving filtering is to determine the boundary point and the non-boundary point. As shown in Fig. 2, pixel 1 was the boundary point of the dark area, while pixels in the filtered window contained both light and dark areas. In this filter window, $K = 4$ points that are closest to the grey value of pixel 1 are selected for calculation, which avoids the aliasing averaging of the information of two areas and, in this way, achieves the goal of protecting the boundary.

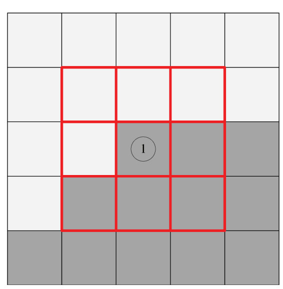


Fig. 2. Schematic diagram of K-nearest neighbor mean filtering

### *2.5 Basic quantum modules*

In this section, we will briefly introduce five quantum computing modules and their quantum circuits, including Swap module, Cyclic shift module, Adder module, Comparator module, and Division-by-two module. These basic quantum computing modules can help us construct more complex quantum function modules.

### *2.5.1 Swap module*

The function of the Swap module is to exchange qubit sequences representing different information, which is composed of quantum Swap gates. In quantum image filtering, $\left|C_{(Y,X)}\right\rangle$ and $\left|C_{(Y',X')}\right\rangle$, which represent two different grayscale values, are processed by the Swap module, and the grayscale information will be exchanged with each other. Fig. 3 shows the quantum circuit of the module and its schematic diagram.

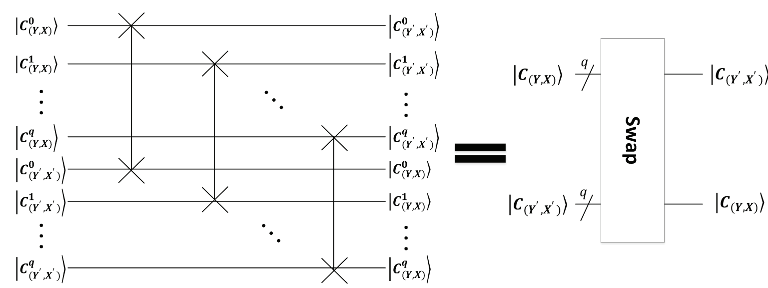


Fig. 3. Quantum circuit of the Swap module

### *2.5.2 Cyclic shift module*

The function of the Cyclic shift module is to move the entire image in different directions as a whole, which is a commonly used in the operation of an image. To obtain the grayscale information of the neighboring pixels, we need to use these four modules: $CS_{X+}$, $CS_{X-}$, $CS_{Y+}$ and $CS_{Y-}$ [7]. The definitions of these four modules are shown in Eq. (2).

$$
\begin{aligned}
CS_{X+}|I\rangle &= \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\left|C_{(Y,X')}\right\rangle|Y\rangle|(X+1) \bmod 2^n\rangle \\
CS_{X-}|I\rangle &= \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\left|C_{(Y,X')}\right\rangle|Y\rangle|(X-1) \bmod 2^n\rangle \\
CS_{Y+}|I\rangle &= \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\left|C_{(Y',X)}\right\rangle|(Y+1) \bmod 2^n\rangle|X\rangle \\
CS_{Y-}|I\rangle &= \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\left|C_{(Y',X)}\right\rangle|(Y-1) \bmod 2^n\rangle|X\rangle
\end{aligned}
\tag{2}
$$

The quantum circuit of the $CS_{X+}$ and $CS_{X-}$ is shown in Fig. 4. Similarly, quantum circuits of $CS_{Y+}$ and $CS_{Y-}$ can be formed by replacing the inputs $|X\rangle$ of $CS_{X+}$ and $CS_{X-}$ with $|Y\rangle$.

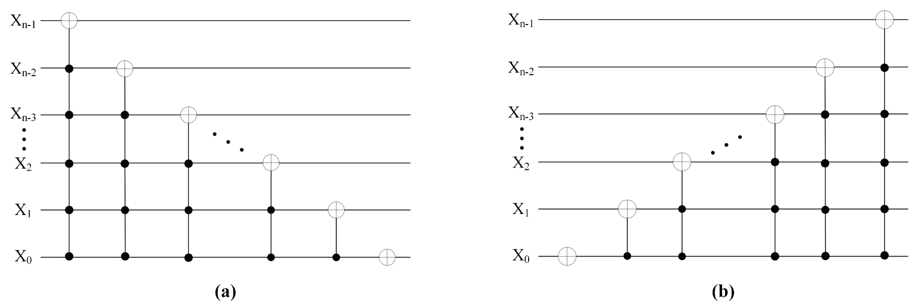


Fig. 4. Quantum circuit of the $CS_{X+}$ module and $CS_{X-}$ module

### 2.5.3 *Adder module*

The function of the Adder module is to add two non-negative integers. This paper uses the Adder module proposed by Steven et al. [24] for mean calculation. This Adder module is composed of two sub-modules, UMA and MAJ. Fig. 5 shows the quantum circuit of the N-bit Adder module.The n-bit Adder module has two inputs $a = a_{n-1}a_{n-2}\dots a_0$ and $b = b_{n-1}b_{n-2}\dots b_0$, and also requires two auxiliary qubits $c_0 = |0\rangle$ and $z = |0\rangle$. It has three outputs $s = s_{n+11}s_n\dots s_0$, $b$ and $s_n$, where $s$ is the sum of the binary sequence $a$ and $b$, and $s_n$ is the highest carry of $s$.

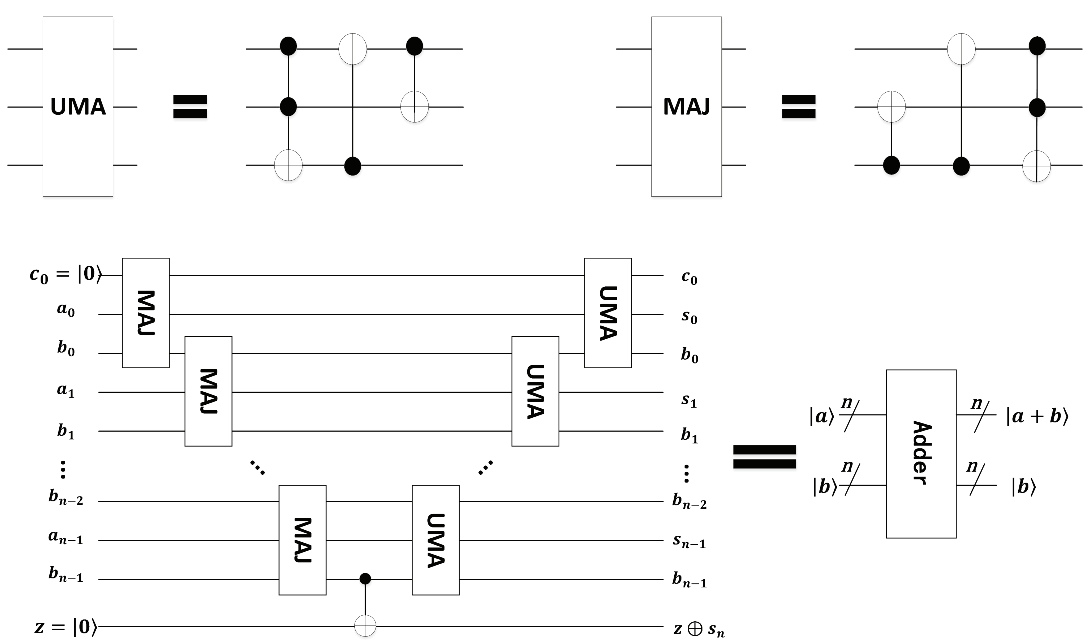


Fig. 5. Quantum circuit of the the n-bit Adder module

### 2.5.4 Comparator module

The function of the Comparator module is conditional judgment. In this paper, the Comparator module designed by Xia et al. [25] is used in the K nearest-neighbor pixel extraction circuit. The Comparator module was designed based on the Adder model. The difference is that the Comparator module replaces MAJ and UMA with CGC and ICGC, where ICGC is a reversible circuit of CGC to ensure that the quantum state of the input will be reserved. Fig. 6 shows the quantum circuit of the N-bit Comparator module. The N-bit Comparator module has two inputs $a = a_{n-1}a_{n-2}\ldots a_0$ and $b = b_{n-1}b_{n-2}\ldots b_0$, and requires two auxiliary qubits $c_0 = |0\rangle$ and $z = |0\rangle$; it has three outputs $a$, $b$ and $c_{out}$, where $c_{out}$ is the comparison result of the binary sequence $a$ and $b$. If $c_{out} = 0$, it means $a \geq b$; otherwise, it means $a < b$ .

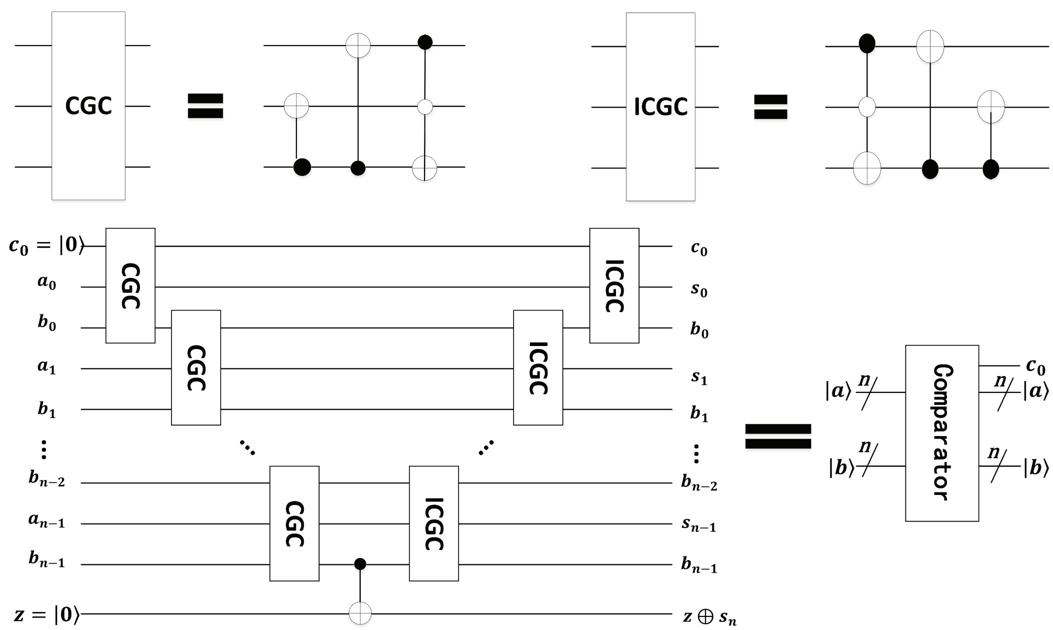


Fig. 6. Quantum circuit of the n-bit Comparator module

### 2.5.5 Division-by-two module

The function of the Division-by-two module is to perform the division-by-two operation on non-negative integers. In this paper, the Division-by-two module in [17] is used in the mean calculation. It adds a $|0\rangle$ auxiliary qubit to move the lowest position of the non-negative integer to $|0\rangle$, and uses a set of Swap gates to move the $|0\rangle$ to the highest position to complete the non-negative integer division by two. Fig. 7 presents the detailed quantum circuit. It should be noted that the Division-by-two module will have an error of 0.5 in the division-by-two operation for odd numbers, but it does not affect the calculation of the grayscale value.

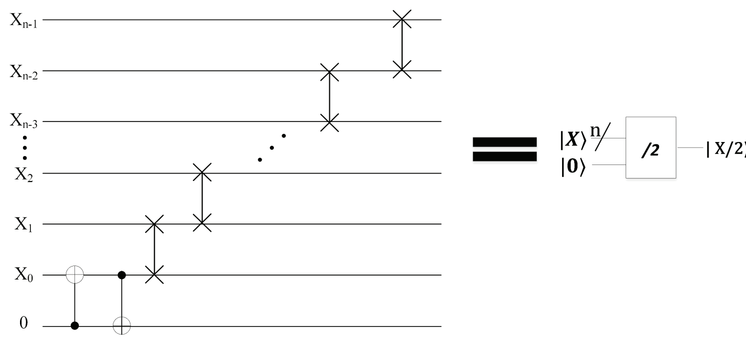


Fig. 7. Quantum circuit of the Division-by-two module

## 3 Proposed quantum image filtering

The quantum image K-nearest neighbor mean filtering includes four functional modules: neighborhood pixels preparation module, distance calculation module, K-nearest neighbor extraction module, and K-nearest neighbor mean calculation module. In this section, we first designed the quantum circuit for these four quantum functional modules in detail and then analyzed the complete filtering process and the corresponding quantum state of the proposed quantum filtering algorithm step by step. Finally, the complete quantum circuit for quantum image K-nearest neighbor mean filtering is constructed. It should be noted that the processed object is a $2^n \times 2^n$ quantum greyscale image based on the NEQR model, and the grayscale range is $[0, 2^{q-1}]$, and the filtering window is $3 \times 3$.

### *3.1 Neighborhood pixels preparation module*

The function of this module is to bind the grayscale information of the neighboring pixels and the central pixel to the same quantum state for further computing operations. The advantage of this image preparation method is that there is no need to prepare multiple identical quantum images, which can reduce the cost of qubits. Similar to the neighborhood preparation method proposed by Abdalla et al. [21], we need to prepare $8q$ qubits to initialize to the $|0\rangle$ state.

In this module, $2n+q$ qubits are used to store the position information and grayscale information of the original image, and prepared $8q$ qubits are used to store the grayscale information of eight neighboring pixels. For preparing the neighborhood pixels, the Cyclic shift module is used to obtain the position information of the neighborhood pixels; the NEQR color setting operation (excluding the H gate) is used to bind the grayscale value of this position to the center pixel. Eq. (3) represents the quantum state after the preparation of the image and neighborhood pixels. It shows that the neighborhood pixels share a pair of qubits representing the position with the center pixel. Fig. 8 shows a specific quantum circuit.

$$|I\rangle = |0\rangle^{\otimes 8q} \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{\text{neighbor}}\rangle^{\otimes 8q} \left|C_{(Y,X)}\right\rangle |Y\rangle |X\rangle \tag{3}$$

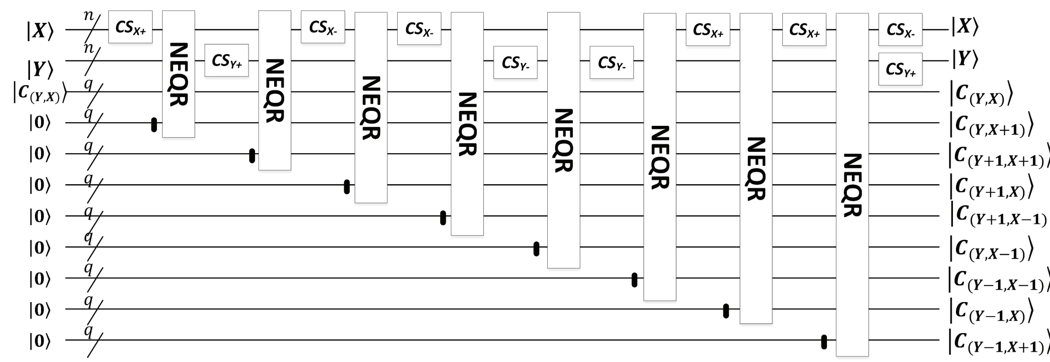


Fig. 8. Quantum circuit of the theneighborhood pixels preparation module

### *3.2 Distance calculation module*

The K-nearest neighbor mean filtering needs to calculate the distance between the neighbor pixel and the center pixel as a condition for finding the K-nearest neighbor, which means that the absolute value of the difference between two non-negative integers needs to be calculated, as shown in Eq. (4).

$$|d\rangle = |C_{\text{neighbor}} - C_{\text{central}}| \tag{4}$$

In general, we can first use the Comparator module to judge the size of two grayscale values, and then subtract the smaller value from the larger grayscale value, so that the output result is positive. This operation requires a Comparator and two Controlled Adder modules, which have high time complexity. We accomplish the same calculation by adding two sign bits and a controlled $CS_{X+}$ module, avoiding the extra Comparator module overhead and reducing the number of quantum gates. Algorithm 1 shows the specific calculation process. Fig. 9 shows the quantum circuit of the absolute value of the difference between two numbers calculation module (ADC). This module uses the quantum inverse circuit of the Adder module to perform the subtraction of integers.

The input of this module is $a = a_{n-1}a_{n-2}\ldots a_0$, $b = b_{n-1}b_{n-2}\ldots b_0$, and a sequence of $|0\rangle$ qubits ready to store $|a-b|$, and two sign bits $s_a$ and $s_b$ added to $a$ and $b$ in advance. In this module, a set of C-NOT gates is applied to $|b\rangle$ to copy its quantum state to $|0\rangle^{\otimes n}$, then it input $a$ and $b$ into the inverse circuit of the Adder module to perform the subtraction. If $a \geq b$, according to the function of the Adder module, the quantum state of $|a\rangle$ remains unchanged, and the result after subtraction is the absolute value of the difference between two non-negative integers, which is stored in $|b\rangle$; Otherwise, $|b\rangle$ stores the complement code of the absolute value. According to the complement code principle, a set of C-NOT gates and a controlled $CS_{X+}$ module is added in the quantum circuit to ensure that $|b\rangle$ stores the absolute value. At this time, if the quantum state of $|s_b\rangle$ is set to $|1\rangle$, then C-NOT gates will flip all the quantum states of $|b\rangle$, and use the $CS_{X+}$ module to add one to the flipped qubits to obtain the absolute value of the difference between the two numbers. At last, a set of Swap gates are used to restore the

**Algorithm 1** Calculate $d = |a - b|$.

**Input:** Give two sequences of integers, a sequence of distance, and two auxiliary qubits:

$a = a_{n-1}a_{n-2}\dots a_0$,

$b = b_{n-1}b_{n-2}\dots b_0$,

$|d\rangle = |0\rangle^{\otimes n}$, $|0\rangle^{\otimes 2}$

**Output:** The absolute value of the difference between two integers $|d\rangle$

Step1: Copy the information of $|b\rangle$ to $|d\rangle$ through the C-NOT gate:

$$\begin{aligned}|I\rangle &= \text{Copy}\left(|0\rangle^{\otimes n}|b\rangle\right)|0\rangle \otimes |a\rangle|0\rangle \\ &= |b\rangle \otimes |b\rangle|0\rangle \otimes |a\rangle|0\rangle\end{aligned}$$

Step2: Calculate the absolute value of the difference between two integers:

**if** $a \geq b$ **then**

$b \Leftarrow \text{QADD}^{-1}(\text{a}, \text{b})$

**else**

$b \Leftarrow \text{QADD}^{-1}(\text{a}, \text{b}) + 1$

**end if**

Step3: Exchange the information of $|b\rangle$ and $|d\rangle$ with each other:

$$\begin{aligned}|J\rangle &= \text{Swap}(|b\rangle|d\rangle)|0\rangle \otimes |a\rangle|0\rangle \\ &= |d\rangle \otimes |b\rangle \otimes |a\rangle\end{aligned}$$

**return** $|d\rangle \bigotimes |b\rangle \bigotimes |a\rangle$;

information of $|b\rangle$ and store the calculated value on $|0\rangle^{\otimes n}$.

With the above module, the grayscale distance between all neighboring pixels and the central pixel can be calculated, and then we can design the quantum circuit of the distance calculation module, as shown in Fig. 10. In this module, $8q$ qubits initialized to $|0\rangle$ need to be prepared to store the distance between the neighborhood pixel and the central pixel, and Eq. (5) shows the quantum state $|I\rangle$ at this time.

$$|I\rangle = |0\rangle^{\otimes 8q} \otimes \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{\text{neighbor}}\rangle^{\otimes 8q} \left|C_{(Y,X)}\right\rangle |Y\rangle|X\rangle \tag{5}$$

In addition, this module needs eight of the above modules to calculate the distance of the neighborhood pixels from the center pixel. For these sub-modules, $|C_{central}\rangle$ is the first input of each module, and $|C_{other}\rangle$ is the second input in turn. Besides, the prepared $8q$ qubits also need to be input into these sub-modules equally. Finally, the quantum state

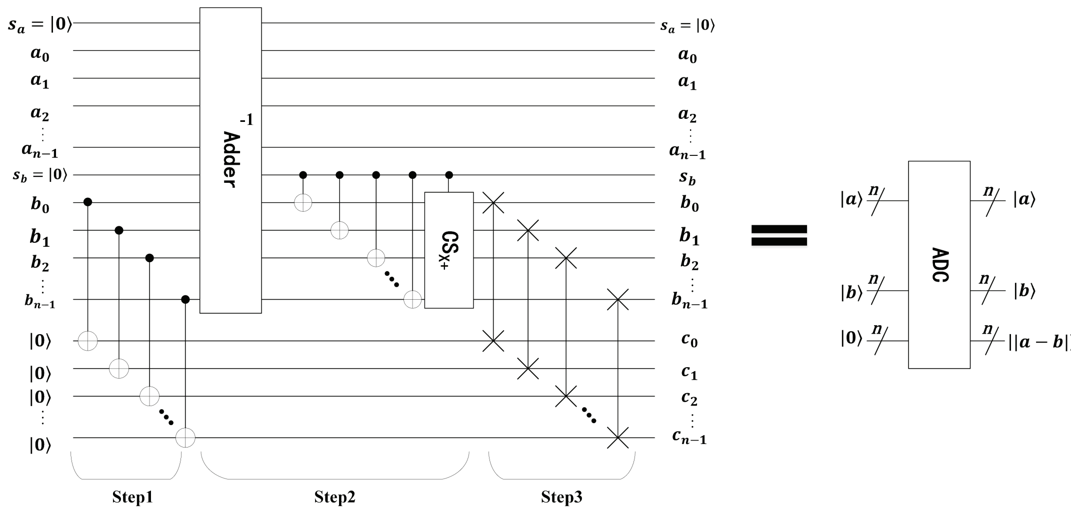


Fig. 9. Quantum circuit of the ADC module

$|J\rangle$ shown in Eq. (6) can be obtained.

$$|I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |d_{\text{neighbor}}\rangle^{\otimes 8q} |C_{\text{neighbor}}\rangle^{\otimes 8q} \left|C_{(Y,X)}\right\rangle |Y\rangle|X\rangle \tag{6}$$

### *3.3 K-nearest neighbor extraction module*

Sorting neighborhood pixels in the order of nearest to farthest distance is a key step in K-nearest neighbor mean filtering for quantum images, which determines the extraction of K-nearest neighbor pixels. The Sort module proposed by Li et al. [18] is to sort two numbers, that is, use a Comparator module to compare the input $|a\rangle$ and $|b\rangle$, and perform an ascending operation on $|a\rangle$ and $|b\rangle$ according to the comparison result. In the process of extracting the K-nearest neighbor, we need to use the distance between the neighbor pixels and the center pixel as the sorting condition. Therefore, this paper improved the Sort module that conforms to this model. As shown in Fig. 11, a Comparator module is used to compare $|x\rangle$ and $|y\rangle$, and based on the comparison result, pair ($|a\rangle$,$|b\rangle$) and ($|x\rangle$,$|y\rangle$) perform the swap operation simultaneously.

Based on the improved Sort module and the principle of bubble sorting, the K-nearest neighbor extraction module is designed in detail. The quantum circuit of this module is shown in Fig. 12. The module consists of 28 Sort modules, whose input includes the grayscale value of neighborhood pixels and the corresponding distance. After bubble sorting, the qubits of the storage distance will be sorted in ascending order, and the grayscale information will also be moved to the corresponding qubit position. So far, it is only necessary to select K pixels from top to bottom in order, that is, the extraction of K-nearest neighbor is completed.

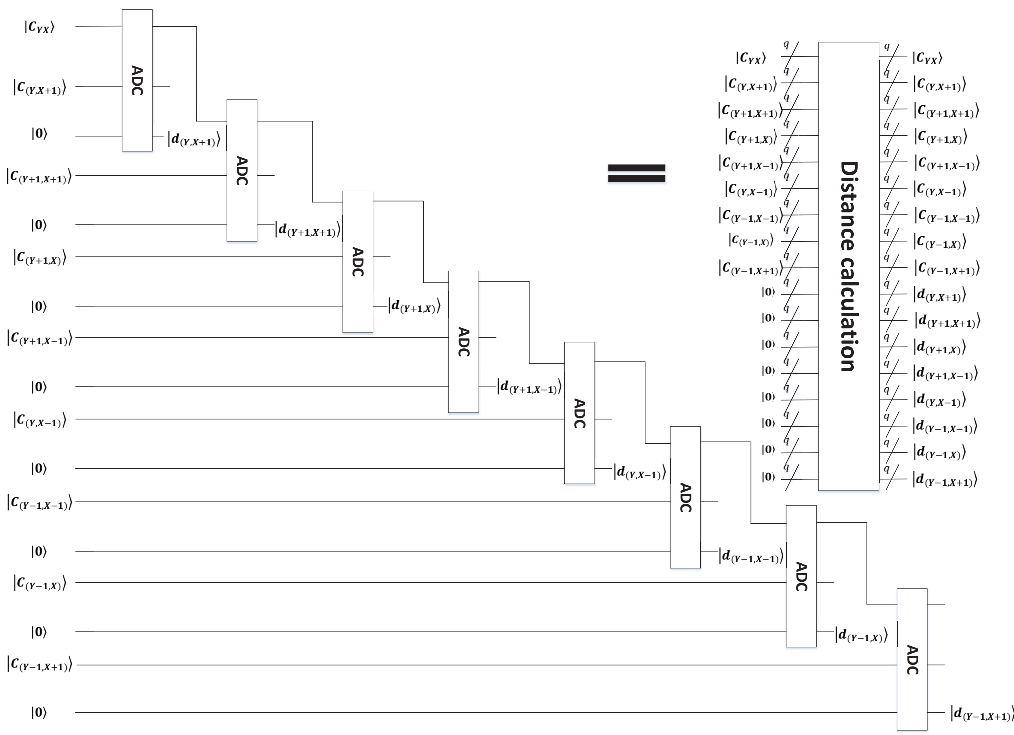


Fig. 10. Quantum circuit of the distance calculation module

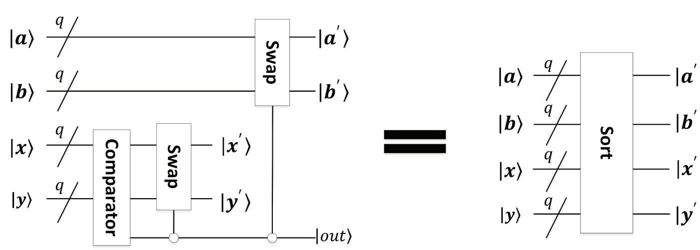


Fig. 11. Quantum circuit of the Sort module

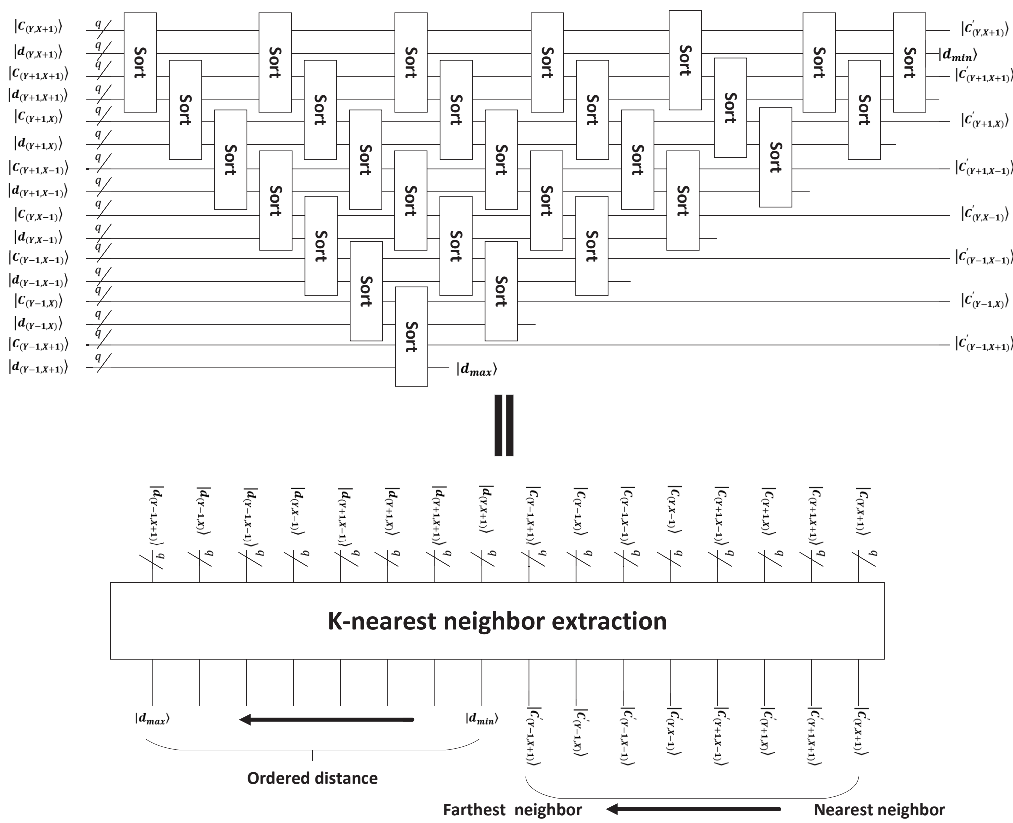


Fig. 12. Quantum circuit of the K-nearest neighbor extraction module

### *3.4 K-nearest neighbor mean calculation module*

The function of the K-nearest neighbor mean calculation module is to calculate mean of the K-nearest neighbors, which determines the filtering effect. The module first expands the grayscale value using $|0\rangle^{\otimes 4k}$ qubits to prevent overflow, then uses $K-1$ Adder modules to sum the grayscale value of K-nearest neighbors and stores the result at the K-neighbor pixel location. Let $U_i$ denote the unitary matrix of the Adder module in the Hilbert space, $|C_k\rangle|C_{k-1}\rangle\ldots|C_2\rangle|C_1\rangle$ denote the extended K-nearest neighbors, then the calculation step is as shown in Eq. (7).

$$\begin{aligned} QADD\left(|C_k\rangle|C_{k-1}\rangle\ldots|C_2\rangle|C_1\rangle\right) &= U_i^{\otimes k-1}|C_k\rangle|C_{k-1}\rangle\ldots|C_2\rangle|C_1\rangle \\ &= |C_{\text{sum}}\rangle\left|C'_{k-1}\right\rangle\ldots|C'_2\rangle|C'_1\rangle \end{aligned} \tag{7}$$

Next, it is necessary to divide $|C_{sum}\rangle$ by K to complete the calculation of the mean value. Due to the high time complexity of the existing quantum divider, this module selects the Division-by-two module to complete $K = 2^i (i = 0, 1, 2)$, such as $K = 4$. Let $U_j$ denote the unitary matrix of the Division-by-two module in the Hilbert space, then the calculation step is as shown in Eq. (8).

$$\begin{aligned} QDIV\left(|C_{\text{sum}}\rangle\left|C'_{k-1}\right\rangle\ldots|C'_2\rangle|C'_1\rangle\right) &= U_j^{\otimes\frac{k}{2}}|C_{\text{sum}}\rangle\left|C'_{k-1}\right\rangle\ldots|C'_2\rangle|C'_1\rangle \\ &= |C_{k-\text{ mean}}\rangle\left|C'_{k-1}\right\rangle\ldots|C'_2\rangle|C'_1\rangle \end{aligned} \tag{8}$$

The specific quantum circuit is shown in Fig. 13.

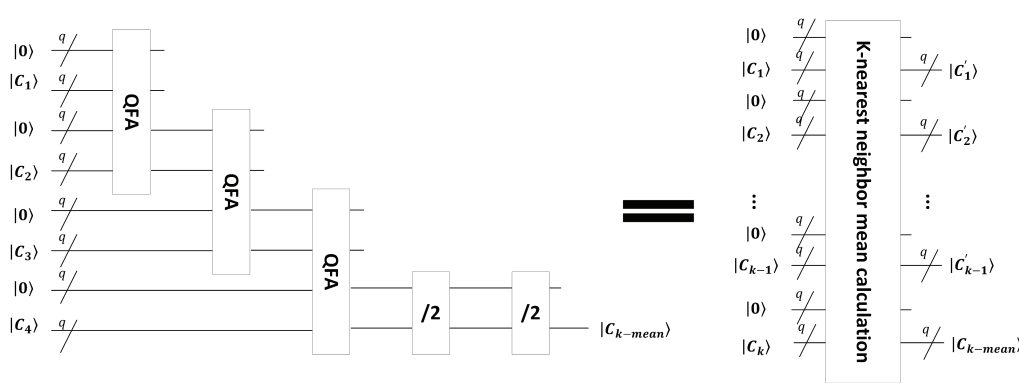


Fig. 13. Quantum circuit of the K-nearest neighbor mean calculation module

### *3.5 Detailed analysis of the proposed filtering*

In this section, the proposed quantum filtering process is analyzed step by step. Then the proposed functional module is used to design the complete quantum filtering circuit. The steps are as follows:

Step1: Prepare an original image, that is, the quantum image preparation module based on the NEQR model prepares the classical image into the corresponding quantum state, and prepares 16 initial qubit sequences $|0\rangle^{\otimes q}$. Eq. (9) gives the quantum state after $|I\rangle$ preparation.

$$|I\rangle = |0\rangle^{\otimes 8q} \otimes |0\rangle^{\otimes 8q} \otimes \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} \left|C_{(Y,X)}\right\rangle |Y\rangle |X\rangle \tag{9}$$

Step2: Prepare neighborhood pixels, that is, the neighborhood pixels preparation module is used to obtain the grayscale information of the neighborhood pixels and store them on $|0\rangle^{\otimes 8q}$. At this time, the neighborhood pixels and the central pixel share the same pair of qubit sequences $|Y\rangle|X\rangle$ for storing position information. Eq. (10) gives the quantum state after this step $|J\rangle$.

$$|J\rangle = |0\rangle^{\otimes 8q} \otimes \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} \begin{array}{l} \left|C_{(Y-1,X+1)}\right\rangle \left|C_{(Y-1,X)}\right\rangle \left|C_{(Y-1,X-1)}\right\rangle \left|C_{(Y,X-1)}\right\rangle \\ \left|C_{(Y+1,X-1)}\right\rangle \left|C_{(Y+1,X)}\right\rangle \left|C_{(Y+1,X+1)}\right\rangle \left|C_{(Y,X+1)}\right\rangle \\ \left|C_{(Y,X)}\right\rangle |Y\rangle |X\rangle \end{array} \tag{10}$$

Step3: Calculate distance, that is, the remaining $|0\rangle^{\otimes 8q}$ and the set of neighbor pixels is input into the distance calculation module. After calculating the distance between all the neighbor pixels and the center pixel, the distance information will be stored in these eight $|0\rangle^{\otimes q}$ qubit sequences. Eq. (11) gives the quantum state $|K\rangle$ processed by the distance calculation module.

$$|K\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} \begin{array}{l} \left|d_{(Y-1,X+1)}\right\rangle \left|d_{(Y-1,X)}\right\rangle \left|d_{(Y-1,X-1)}\right\rangle \left|d_{(Y,X-1)}\right\rangle \\ \left|d_{(Y+1,X-1)}\right\rangle \left|d_{(Y+1,X)}\right\rangle \left|d_{(Y+1,X+1)}\right\rangle \left|d_{(Y,X+1)}\right\rangle \\ \left|C_{(Y-1,X+1)}\right\rangle \left|C_{(Y-1,X)}\right\rangle \left|C_{(Y-1,X-1)}\right\rangle \left|C_{(Y,X-1)}\right\rangle \\ \left|C_{(Y+1,X-1)}\right\rangle \left|C_{(Y+1,X)}\right\rangle \left|C_{(Y+1,X+1)}\right\rangle \left|C_{(Y,X+1)}\right\rangle \\ \left|C_{(Y,X)}\right\rangle |Y\rangle |X\rangle \end{array} \tag{11}$$

Step4: Extract K-nearest neighbors, that is, the distance and grayscale value is input into the K-nearest neighbor extraction module. The Comparator module and the Swap module will sort the grayscale values in ascending order of distance. Eq. (12) gives the quantum state $|L\rangle$ processed by the K-nearest neighbor extraction module.

$$|L\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |d\rangle^{\otimes 8} |C_8\rangle |C_7\rangle |C_6\rangle |C_5\rangle |C_4\rangle |C_3\rangle |C_2\rangle |C_1\rangle |C_0\rangle |X\rangle \| Y\rangle \tag{12}$$

The nearest neighbor pixel sequence $|C_8\rangle \ldots |C_2\rangle|C_1\rangle$is the nearest neighbor pixel to the farthest neighbor pixel from right to left. $|d\rangle^{\otimes 8}$ represents an ascending sequence of distances.

Step5: Calculate mean value, that is, $|C_k\rangle \ldots |C_2\rangle|C_1\rangle$ is input into the K-nearest neighbor mean calculation module to calculate its mean, and the calculation result $|C_{k-mean}\rangle$ will be saved on $|C_k\rangle$.

Step6: Replace center pixel, that is, the Swap module is used for replacing $|C_{k-mean}\rangle$ with $|C_{(Y,X)}\rangle$, Eq. (13) gives the output quantum state $|I'\rangle$.

$$|I'\rangle = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1} \begin{array}{l} \left|C'_{(Y-1,X+1)}\right\rangle \left|C'_{(Y-1,X)}\right\rangle \left|C'_{(Y-1,X-1)}\right\rangle \left|C'_{(Y,X-1)}\right\rangle \\ \left|C'_{(Y+1,X-1)}\right\rangle \left|C'_{(Y+1,X)}\right\rangle \left|C'_{(Y+1,X+1)}\right\rangle \left|C'_{(Y,X+1)}\right\rangle \\ |C_{k-mean}\rangle\,|Y\rangle|X\rangle \end{array} \tag{13}$$

Step7: Finally, the quantum image can be retrieved or directly used as the input of other quantum image algorithms.

The complete quantum circuit for quantum image K-nearest neighbor mean filtering is shown in Fig. 14.

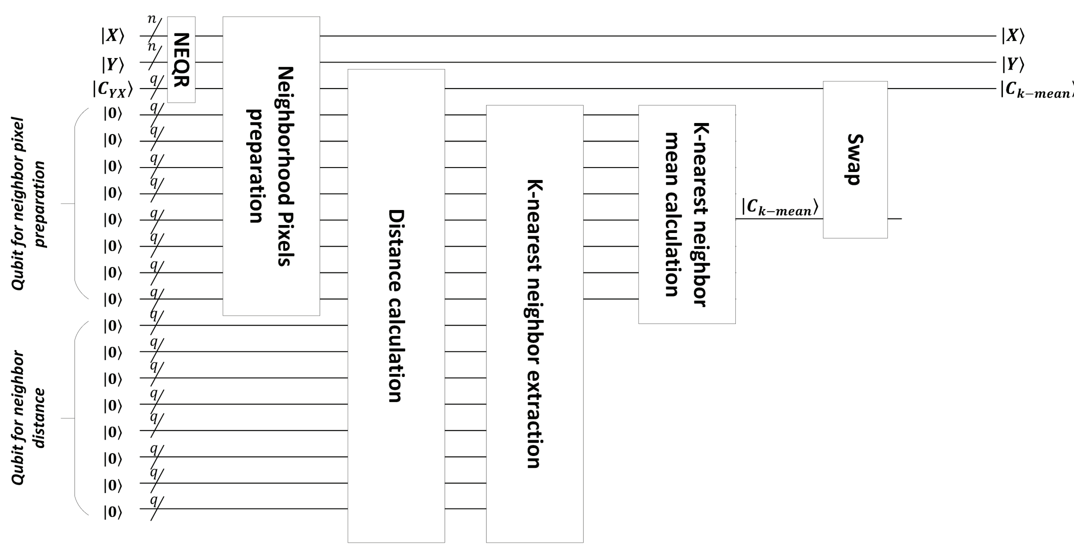


Fig. 14. The complete circuit of proposed quantum image filtering

## 4 Time complexity analysis

The time complexity of the quantum algorithm is correlated with the number of quantum gates used in the quantum circuit. According to the calculation method proposed in [26], we calculate the time complexity of the entire algorithm from bottom to top.

- **Swap module**. This module has $q$ Swap gates ($q$ refers to the number of qubits representing the greyscale information). A Swap gate can be composed of three C-NOT gates. Therefore, the time complexity of the Swap module is $O(3q)$.

- **Cyclic shift module**. The time complexity of this module is $O(n^2)$ given in [7].
- **Adder module**. This module has two sub-modules, UMA and MAJ. Both UMA and MAJ can be split into two C-NOT gates and one Toffoli gate. For the input of $q$-bit qubit sequences, this module needs to use $q$ pairs of UMA and MAJ modules and one C-NOT gate. Its time complexity is $O(14q + 1)$.
- **Comparator module**. This module uses the CGC and ICGC modules. Like the Adder module, the time complexity is $O(14q + 1)$.
- **Division-by-two module**. The $q$-bit Division-by-two module has two C-NOT gates and $q - 1$ Swap gates. Its time complexity is $O(3q - 1)$.
- **ADC module**. This module consists of a $(q + 1)$-bit subtractor (obtained by the inverse operation of the Adder module), $2q$ C-NOT gates, $q$ Swap gates, and one $q$-bit Cyclic shift module, and its time complexity is $O(14(q+1)+1+2q+3q+q^2) = O(q^2 + 19q + 15)$.
- **Sort module**. It consists of a $q$-bit Comparator module and two $q$-bit Swap modules, and its time complexity is $O(20q + 1)$.

In Fig. 17, the quantum circuit of the proposed method includes a total of ten Cyclic shift modules, eight ADC modules, twenty-eight Sort modules, $k-1$ Adder modules, and $\log_2 k$,(k=2,4,8) Division-by-two modules. Therefore, the total time complexity does not exceed $O(10n^2 + 8(q^2 + 19q + 15) + 28(20q + 1) + (k - 1)(14q + 1) + \log_2 k(14q + 1)) \approx O(n^2 + q^2)$. The corresponding classical filtering algorithm needs to process each pixel individually, and its time complexity does not exceed $O(2^{2n})$. It can be seen that the time complexity of our proposed quantum image filtering algorithm is only a second-order polynomial function of $n$. Therefore, our proposed quantum image filtering algorithm highly reduces the time complexity of classical filtering tasks.

## 5 Simulation experiments and analysis

The number of available qubits of the existing programmable quantum computer is not enough to realize the proposed quantum image filtering algorithm. We use classical computer simulation to complete the quantum image K-nearest neighbor mean filtering, and the experimental environment is supported by MATLAB R2014b. In the numerical simulation experiment, we selected the five $512R512$ test images shown in Fig. 15 for image processing, and the grayscale range is $[0, 2^7]$.

Firstly, salt and pepper noise with a density of 0.1 and Gaussian noise with a mean value of 0 and variance of 0.01 are added to the original image respectively, and then the above gray image is filtered by quantum K-nearest neighbor mean filter and corresponding classical filtering when $K = 4$ and the filter window is $3\times3$. Fig. 16 and Fig. figfig17 show the effects of different filtering algorithms on images contaminated by salt and pepper noise and Gaussian noise, respectively. As can be seen in Fig. 16, the quantum image filtering algorithm proposed in this paper can significantly filter salt and pepper noise, protect the boundaries of the image, and improve clarity.

The above is based on the intuitive effect of human vision that is not necessarily able to distinguish specific details. Next, we analyze the noise reduction performance of the quantum image K-nearest neighbor filtering proposed from the aspect of image quality by calculating the peak signal-to-noise ratio PSNR of the original image and the denoised


Fig. 15. The complete circuit of proposed quantum image filtering

image. Peak Signal-to-Noise Ratio (PSNR) is a commonly used measure of image quality in image processing. It is based on the grayscale value of image pixels to analyze the effect before and after image processing. Eq. (14) gives the calculation method of PSNR.

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{2^{2n}}\sum_{i=0}^{2^n}\sum_{j=0}^{2^n}\left[I_0(i,j)-I_P(i,j)\right]^2}\right) \tag{14}$$

In Eq. (14), $I_O$ and $I_P$ represent the original image and the processed image, respectively, and Table 1 gives the PSNR data of the five color images before and after the experiment.

In Table 1, $P_i(i = 0, 1, 2)$ represents the PSNR value calculated by the noise image, the image after the quantum filtering proposed, and the image after the corresponding classical filtering. After the quantum filtering proposed in this paper processes the salt and pepper noise image with a noise density of 0.1, the average PSNR value increases by 11.2785db, while the classical scheme increases by 10.9072db. The difference is only 0.3713db. After the quantum filtering deals with the image polluted by Gaussian noise with a mean value of 0 and variance of 0.01, the average PSNR value increases by 3.8887db, while the classical scheme increases by 3.8518db. The difference is only 0.0369db. It shows that the quantum image filtering proposed in this paper has a very obvious effect on the salt and pepper noise of the image, and also has a certain suppression effect on the Gaussian noise.

The reason for the above results is that the filtering scheme proposed in this paper inherits the advantages of the statistical sorting filter and the linear filter. The gray average of the K-nearest neighbor of the central pixel can effectively filter out the extreme value of pixels generated by the salt and pepper noise. At the same time, because the K-nearest neighbor belongs to the same region, it can ensure that the edge pixels are still in the region after processing, which can preserve the boundary point. For images polluted by Gaussian noise, the Gaussian noise is distributed on each pixel. The filtering scheme proposed in this paper selects K-neighbor pixels for the gray average, which can also suppress the Gaussian noise to a certain extent. However, because not all neighborhood pixels are selected, the Gaussian noise suppression effect will be reduced to a certain extent compared with the weighted mean filter.

Compared with the classical K-nearest neighbor filtering, the quantum filter proposed in this paper has basically the same PSNR value after processing the same image. The reason for the slight difference is that the Cyclic shift module used in this paper

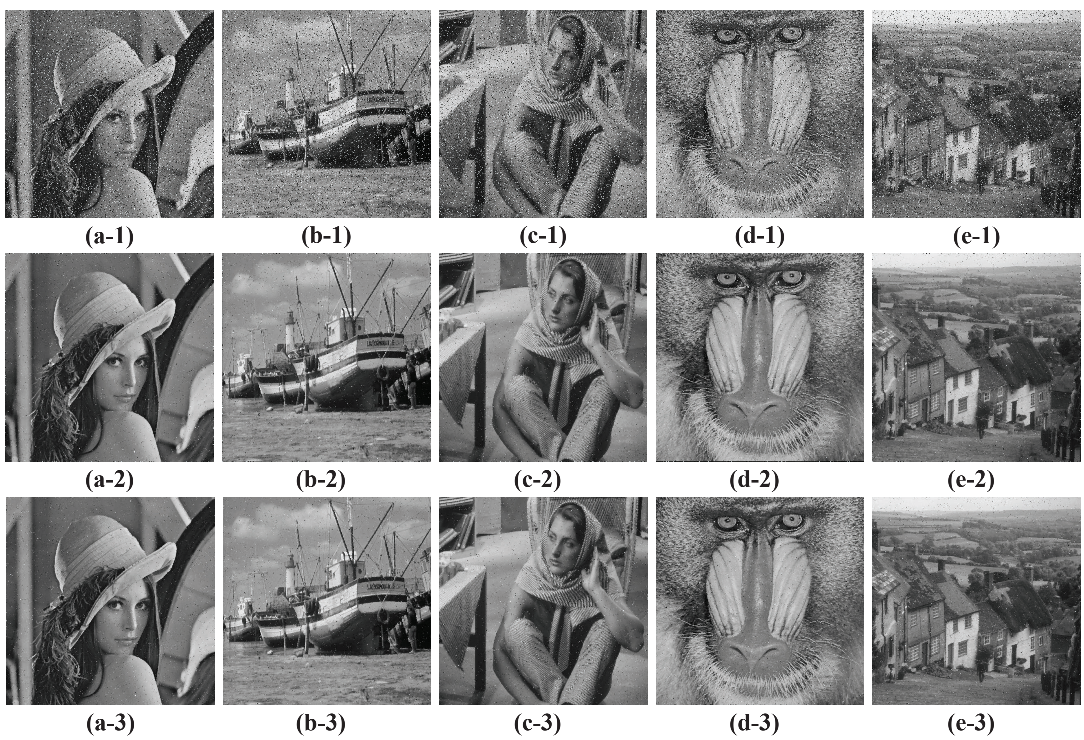


Fig. 16. The effect of salt and pepper noise pollution and filtering, where (a-1, b-1, c-1, d-1, e-1) is the image after adding salt and pepper noise with a density of 0.1, (a-2, b-2, c-2, d-2, e-2) is the image processed by the classical K-nearest neighbor mean filtering, (a-3, b-3, c-3, d-3, e-3) is the image processed by the proposed method.

can fill the edge pixels of the images, and then can filter the edge pixels, while the corresponding classical filtering generally does not process the edge pixels.

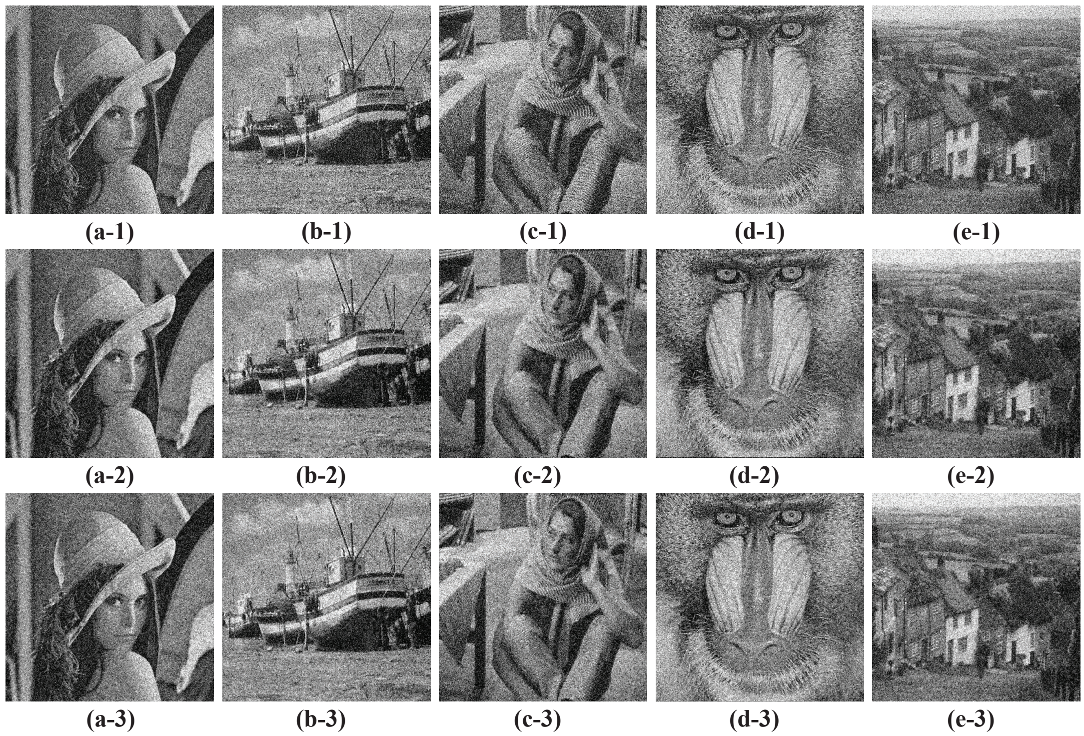


Fig. 17. The effect of Gaussian noise pollution and filtering, where (a-1, b-1, c-1, d-1, e-1) is the image after adding Gaussian noise with a mean value of 0 and variance of 0.01, (a-2, b-2, c-2, d-2, e-2) is the image processed by the corresponding classical filtering, (a-3, b-3, c-3, d-3, e-3) is the image processed by the proposed method.

## 6 Conclusion and future work

As an emerging interdisciplinary subject integrating quantum computing and image processing, quantum image processing can make full use of the excellent acceleration characteristics of quantum computing and will have a profound impact on image processing. This paper studied the boundary-preserving filtering based on quantum images and proposed a quantum image K-nearest neighbor mean filtering. And we designed the quantum circuit of the distance calculation module, the K-nearest neighbor extraction module, and the K-nearest neighbor mean calculation module. Then the time complexity of the proposed quantum algorithm is analyzed, and the time complexity is calculated to be $O(n^2 + q^2)$, which verifies the acceleration performance of the quantum algorithm. Finally, we carried out a simulation experiment. The relevant indicators show that the quantum image K-nearest neighbor mean filtering has the same effect as the classical K-nearest neighbor mean filtering. However, due to the properties of quantum superposition and quantum entanglement, in terms of processing speed and efficiency of image filtering, quantum image K-nearest neighbor mean filtering is better than classical K-nearest neighbor mean filtering.

## References


1. S.E. Venegas-Andraca and S. Bose(2003), *Storing, processing, and retrieving an image using quantum mechanics*, Quantum Information and Computation, International Society for Optics and Photonics, SPIE, Vol, 5105, pp. 137-147.
2. P.Q. Le and F. Dong(2011), *A flexible representation of quantum images for polynomial preparation, image compression, and processing operations*, Quantum Inf. Process, 10(1), pp. 63-84.
3. Haisheng锛孡i and Qingxin锛孼hu(2013), *Image storage, retrieval, compression and segmentation in a quantum system*, Quantum Inf. Process, 12(6), pp. 2269-2290.
4. S. Caraiman and V. Manta(2012), *Image processing using quantum computing*, 16th International Conference on System Theory, Control and Computing (ICSTCC), pp. 1-6.
5. Yi Zhang and Kai Lu(2013), *NEQR: a novel enhanced quantum representation of digital images*, Quantum Inf. Process, 12(8), pp. 2833-2860.
6. Yi Zhang and Kai Lu(2017), *A novel quantum representation of color digital images*, Quantum Inf. Process, 16(2), pp. 42.
7. PQ Le and AM Iliyasu(2011), *Strategies for designing geometric transformations on quantum images*, Theoretical Computer Science, 412(15), pp. 1406-1418.
8. Haisheng, Li and Qingxin, Zhu(2013), *Image storage, retrieval, compression and segmentation in a quantum system*, Quantum Inf. Process. 12(6), pp. 2269-2290.
9. PQ Le and AM Iliyasu(2012), *Watermarking and authentication of quantum images based on restricted geometric transformations*, Inf. Sci, 186, pp. 126-149.
10. ER Hancock(2015), *Local feature point extraction for quantum images*, Quantum Inf. Process, 14(5), pp. 1573-1588.
11. N. Jiang and L Wang (2015), *Quantum image scaling using nearest neighbor interpolation*, Quantum Inf. Process. 14(5), pp. 1559-1571.
12. S. Du and D. Qiu(2019), *Enhanced double random phase encryption of quantum images*, Results Phys. 13, pp. 102161.
13. S. Caraiman and V. Manta(2013), *Quantum image filtering in the frequency domain*, Advances in Electrocal and Computer Engineering, 13, pp. 77-84.
14. P. Li and H. Xiao(2018), *An improved filtering method for quantum color image in frequency domain*, International Journal of Theoretical Physics, 57, pp. 258-278.
15. S. Yuan and X. Mao(2017), *Quantum image filtering in the spatial domain*, International Journal of Theoretical Physics, 56(8), pp. 2495-2511.
16. S. Yuan and X. Mao(2018), *Improved quantum image filtering in the spatial domain*, International Journal of Theoretical Physics, 57(3), pp. 804-813.
17. P. Li and X. Liu(2017), *Quantum image weighted average filtering in spatial domain*, International Journal of Theoretical Physics, 56(11), pp. 3690-3716.
18. P. Li and X. Liu(2018), *Quantum image median filtering in the spatial domain*, Quantum Inf. Process, 17(3), pp. 1573-1332.
19. S.X. Jiang and R.G. Zhou(2019), *Improved Quantum Image Median Filtering in the Spatial Domain*, International Journal of Theoretical Physics, 58(7), pp. 2115-2133.
20. H.Y. Xia and Y.F. Xiao(2020), *Quantum circuit design of approximate median filtering with noise tolerance threshold*, Quantum Inf. Process, 19, pp. 183.
21. A.E. Ali and H. Abdel-Galil(2020), *Quantum image mid-point filter*, Quantum Inf. Process, 19, pp. 238.
22. Maxwell Henderson and Samriddhi Shakya(2019), *Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits*, arXiv:1904.04767.
23. Atchade-Adelomou and Alonso-Linaje(2022), *Quantum-enhanced filter: QFilter*, Soft Comput,

26(1), pp. 7167-7174.
24. S.A. Cuccaro and T.G. Draper(2004), *A new quantum ripple-carry addition circuit*, arXiv:quant-ph/0410184.
25. H. Xia and H. Li(2018), *An efficient design of reversible multi-bit quantum Comparator via only a single ancillary bit*, International Journal of Theoretical Physics, 57(12), pp. 3727-3744.
26. M.A. Nielsen and I.L. Chuang(2011), *Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th edn*, Cambridge University Press(New York).