



# Improved Quantum Image Median Filtering in the Spatial Domain

SheXiang Jiang<sup>1,2,3</sup> · Ri-Gui Zhou<sup>1,2</sup>  · WenWen Hu<sup>1,2</sup> · YaoChong Li<sup>1,2</sup>

Received: 11 November 2018 / Accepted: 23 March 2019 / Published online: 9 April 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

In some image processing algorithms, such as those for image feature extraction and segmentation, filtering is a significant pre-processing step to remove noises and improve image quality. An improved quantum image median filtering approach is proposed, and its corresponding quantum circuit is designed in this work. The main idea of the approach is that first the classical image is converted into a quantum version based on the novel enhanced quantum representation (NEQR) of digital images, and then a unique quantum module is designed to realize the median calculation of neighborhood pixels for each pixel point in the image. Finally, in order to improve the filtering effect, extremum detection is employed to distinguish noises from true signals. The experimental results show that a competitive filtering performance is obtained compared with previous methods. In addition, a network complexity analysis of the quantum circuit suggests that the proposed filtering approach can perform enormous speed-up over its corresponding classical counterparts.

**Keywords** Quantum image processing · Quantum image filtering · Median filtering · Extremum detection

## 1 Introduction

The quantum computation model was proposed by Feynman in 1982 [1]. Moore's law states that computer performance will double roughly every 2–3 years [2]. However, this rule will no longer work effectively when electronic components cannot continue to shrink. The

---

✉ Ri-Gui Zhou  
rgzhou@shmtu.edu.cn

<sup>1</sup> College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China

<sup>2</sup> Research Center of Intelligent Information Processing and Quantum Intelligent Computing, Shanghai 201306, China

<sup>3</sup> School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001 Anhui, China

emergence of quantum technology provides potential to solve this issue. This new technology can help us design much smaller embedded electronic components to make computation more efficient and faster [3], with several outstanding merits, such as quantum superposition and entanglement. According to these merits, in 1994, Peter Shor designed a quantum algorithm to achieve integer factoring in polynomial time [4]. This was followed closely by Grover's quadratic speed-up database searching algorithm [5]. These two representative quantum algorithms provide strong evidence supporting quantum computers as being much more powerful than classical computers.

In recent years, with the rapid development of quantum technology, quantum computation has been applied in various fields of computer science, especially in image processing, which is devoted to utilizing quantum computers to perform image processing tasks. Quantum image processing is still in its early stage, and thus is facing several fundamental problems, such as how to represent and store an image in quantum computers appropriately, and how to implement image processing algorithms. In the past several decades, multiple quantum models have been proposed to represent image information, including the Qubit Lattice [6], Entangled Image [7], Real Ket [8], Flexible Representation of Quantum Images [9], Novel Enhanced Quantum Representation (NEQR) [10], and Quantum Image Representation for Log-polar Images [11]. Moreover, many quantum image processing algorithms based on these representation models have been proposed, such as quantum image scaling [12–14], quantum image feature extraction [15, 16], quantum image transformation [17–19], quantum image matching [20–22], quantum image compression [23], quantum image segmentation [23, 24], quantum image watermarking [25–29], and quantum image encryption [30–32].

Owing to the rapid development of modern technology, digital images have become ubiquitous in our daily life. During capture and transmission, however, digital images are likely to be degraded by various noises, and filtering may be one of the best ways to remove noises. Hence, filtering plays a crucial role in image processing, which is a significant pre-processing step in image feature extraction and segmentation. Furthermore, filtering enables some complex image processing to be realizable, such as image matching and image searching.

Unfortunately, the current research achievements on quantum image filtering are still scarce. This is because correlation or convolution operations are required in most filtering approaches, but they have been demonstrated as impossible in quantum implementations due to the constraint of linearity [33]. In 2013, Simona et al. [34] proposed the frequency domain filtering of quantum images, in which the Fourier transform was used to turn a quantum image from the spatial domain to the frequency domain; meanwhile, the inverse Fourier transform was used to perform the transformation from the frequency domain to the spatial domain. In order to avoid quantum convolution, they employed the quantum oracle to realize filtering behavior, but the detailed circuit of the black box was lacking. Later, Yuan et al. [35] proposed quantum image filtering in the spatial domain, and quantum addition was substituted for quantum multiplication. Only a year later, two drawbacks of this method were pointed out by [35]: (1) it is essential to know the specific filter coefficients before the filtering operation; and (2) the method has a limitation on integer filter coefficients but not decimal ones. Yuan et al. [35] also gave an improved version that takes full advantage of the quantum multiplication and resolves these two problems. Additionally, Li et al. proposed three approaches related to quantum image filtering, namely, quantum image weighted average filtering in the spatial domain [36], quantum image median filtering in the spatial domain [37], and an improved filtering method for the quantum color image frequency domain [38].

Analogous to Ref. [37], the present paper also concentrates on image median filtering in the spatial domain. However, the differences are as follows: (1) the proposed approach uses less elementary quantum gates than that of [37]; (2) in order to store neighborhood pixels, Ref. [37] employs a set of eight images based on NEQR, including position information and color information, while our scheme stores only the color information of the neighborhood pixels entangled with position information of the original image; and (3) we use the extremum detection approach to distinguish between noises and normal signal points, which can improve filtering effectiveness.

This paper is organized as follows. Section 2 briefly introduces some necessary theoretical basis, including NEQR model, median filtering, and noise detection. In Section 3, several core quantum modules are introduced; these are used to design the integrate quantum circuit for image filtering. Section 4 describes the proposed approach and quantum circuit in detail. Section 5 gives the circuit complexity analysis and experimental results. Finally, the conclusions are drawn in Section 6.

## 2 Preliminaries

### 2.1 NEQR

For a gray image with the size  $2^n \times 2^n$  and grayscale range  $2^q$ , NEQR encodes the position information into  $2n$  qubits and encodes the grayscale intensity into  $q$  qubits, as represented by the following equation:

$$|I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX}\rangle |YX\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX}^{q-1} C_{YX}^{q-2} \dots C_{YX}^0\rangle |YX\rangle, \tag{1}$$

where  $|C_{YX}\rangle \in \{0, 1, \dots, 2^q - 1\}$ , and  $C_{YX}^{q-1}, C_{YX}^{q-2}, \dots, C_{YX}^0 \in \{0, 1\}$ . Figure 1 shows an example of a  $2^2 \times 2^2$  gray image and its NEQR representation.

### 2.2 Median Filtering

Median filtering is an order-statistics filter, and has been widely used in image processing to remove noises. The underlying principle of the filter is to use the median of the pixels encompassed by the filtering window to replace the pixel value at the center of the filtering window. The pixels encompassed by the filtering window also are known as neighborhood pixels. The window size may be any  $n \times n$ , where  $n$  is generally an odd number; the size of  $3 \times 3$  can usually meet filtering expectations.

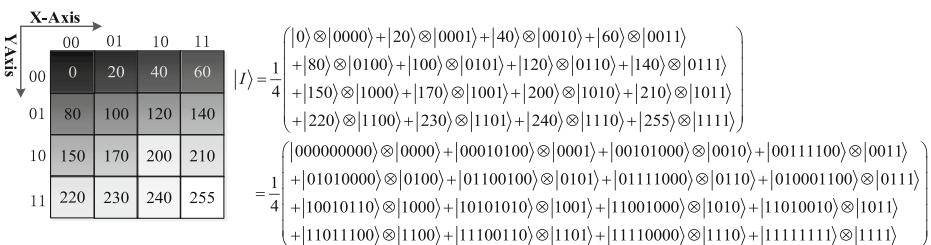


Fig. 1 An example of a  $2^2 \times 2^2$  gray image and its NEQR representation

Sorting is deemed to be the most common method for obtaining the median of neighborhood pixels, such as the bubble sorting used in [37]. However, this type of sorting is of high network complexity. To avoid this issue, a unique algorithm to acquire the median of the neighborhood pixels at a faster speed is proposed in Algorithm 1. The median, maximum, and minimum of neighborhood pixels are easily obtained through this algorithm, which will play key role in noise detection.

---

**Algorithm 1. The median, maximum, and minimum calculation of neighborhood pixels**

Input: the neighborhood pixels

Output: the median, maximum, and minimum of the neighborhood pixels

Step 1: Column sorting

Step 2: Row sorting

Step 3: Right diagonal sorting

---

To demonstrate the algorithm in detail, assume that the size of the filtering window is  $3 \times 3$ , which includes nine pixels, namely  $A_i$ ,  $i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ; this is shown in Fig. 2(a). The detailed process is described below.

Step 1. Perform column sorting based on ascending order. The new nine pixels are calculated in Fig. 2(b), namely,  $B_i$ ,  $i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . The relationship between  $B_i$  and  $A_i$  is as follows:

$$\begin{aligned}
 B_1 &= \min[A_1, A_4, A_7] \\
 B_2 &= \min[A_2, A_5, A_8] \\
 B_3 &= \min[A_3, A_6, A_9] \\
 B_4 &= \text{median}[A_1, A_4, A_7] \\
 B_5 &= \text{median}[A_2, A_5, A_8] \\
 B_6 &= \text{median}[A_3, A_6, A_9] \\
 B_7 &= \max[A_1, A_4, A_7] \\
 B_8 &= \max[A_2, A_5, A_8] \\
 B_9 &= \max[A_3, A_6, A_9]
 \end{aligned} \tag{2}$$

Step 2. Perform row sorting based on ascending order. The new nine pixels are calculated in Fig. 2(c), namely,  $C_i$ ,  $i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . The relationship between  $C_i$  and  $B_i$  is as follows:

$$\begin{aligned}
 C_1 &= \min[B_1, B_2, B_3] \\
 C_2 &= \text{median}[B_1, B_2, B_3] \\
 C_3 &= \max[B_1, B_2, B_3] \\
 C_4 &= \min[B_4, B_5, B_6] \\
 C_5 &= \text{median}[B_4, B_5, B_6] \\
 C_6 &= \max[B_4, B_5, B_6] \\
 C_7 &= \min[B_7, B_8, B_9] \\
 C_8 &= \text{median}[B_7, B_8, B_9] \\
 C_9 &= \max[B_7, B_8, B_9]
 \end{aligned} \tag{3}$$

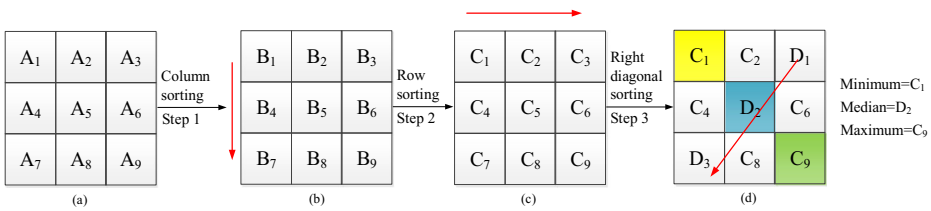


Fig. 2 An example demonstrating Algorithm 1

Step 3. Perform right diagonal sorting based on ascending order. The ascending order of  $C_3$ ,  $C_5$ , and  $C_7$  is denoted as  $D_1$ ,  $D_2$ , and  $D_3$ , as shown in Fig. 2(d). The relationship between  $D_i$  and  $C_i$  is as follows:

$$\begin{aligned}
 D_1 &= \min[C_3, C_5, C_7] \\
 D_2 &= \text{median}[C_3, C_5, C_7]. \\
 D_3 &= \max[C_3, C_5, C_7]
 \end{aligned}
 \tag{4}$$

According to Steps 1,2, and 3, a conclusion can be drawn. Inspired from Ref. [39], the supporting proofs are given here.

**Conclusion**  $C_1$  is the minimum of the nine pixels;  $C_9$  is the maximum of the nine pixels;  $D_2$  is the median of the nine pixels;  $C_2$ ,  $C_4$  and  $D_1$  are in the interval (minimum, median);  $C_6$ ,  $C_8$  and  $D_3$  are in the interval (median, maximum).

**Proof** (1)  $C_1$  is the minimum of the nine pixels, and  $C_9$  is the maximum of the nine pixels.

According to equations (2) and (3),

$$\begin{aligned}
 C_1 &= \min[B_1, B_2, B_3] = \min[\min[A_1, A_4, A_7], \min[A_2, A_5, A_8], \min[A_3, A_6, A_9]] \\
 C_9 &= \max[B_7, B_8, B_9] = \max[\max[A_1, A_4, A_7], \max[A_2, A_5, A_8], \max[A_3, A_6, A_9]]
 \end{aligned}
 \tag{5}$$

Therefore,  $C_1$  is the minimum of the nine pixels, and  $C_9$  is the maximum of the nine pixels.

(2)  $C_2$ ,  $C_4$  in the interval (minimum, median);  $C_6$  and  $C_8$  are in the interval (median, maximum).

Taking  $C_2$  for example, since  $C_2$  and  $C_3$  are both derived from the set of  $\{B_1, B_2, B_3\}$ , assume  $C_2 = B_i$ ,  $C_3 = B_j$ ,  $i, j \in \{1, 2, 3\}$ ,  $i \neq j$ . According to equation (2), for  $B_k$ ,  $k \in \{1, 2, 3\}$ , there exists at least two pixels larger than or equal to themselves, namely,  $B_{k+3}$ ,  $B_{k+6}$ . For  $C_3$ ,  $C_3 = B_j \leq B_{j+3} \leq B_{j+6}$ , and for  $C_2$ ,  $C_2 = B_i \leq B_{i+3} \leq B_{i+6}$ . According to equation (3),  $C_2 \leq C_3$ , and thus for  $C_2$  there are at least five pixels larger than or equal to itself, namely  $B_j$ ,  $B_{j+3}$ ,  $B_{j+6}$ ,  $B_{i+3}$ ,  $B_{i+6}$ . Hence,  $C_2$  is in the interval (minimum, median). Adopting a similar method, the result can be proved that  $C_4$  is in the interval (minimum, median);  $C_6$  and  $C_8$  are in the interval (median, maximum).

(3)  $D_2$  is the median of the nine pixels;  $D_1$  is in the interval (minimum, median);  $D_3$  is in the interval (median, maximum).

According to (1) and (2) described above, the six pixels of  $C_1, C_2, C_4, C_6, C_8,$  and  $C_9$  cannot be the median of the nine pixels and the median should be one of  $C_3, C_5,$  and  $C_7$ . Then, according to equation (4),  $D_2$  must be the median of the nine pixels,  $D_1$  should be in the interval (minimum, median), and  $D_3$  should be in the interval (median, maximum).

To verify the correctness of the algorithm, an example is given in Fig. 3.

In Step 1 and Step 2 of the algorithm, nine comparisons and nine swaps are used, respectively. In Step 3, three comparisons and three swaps are used. Hence, the whole algorithm uses only 21 comparisons and 21 swaps, which is superior to bubble and selection methods using 36 comparisons and 36 swaps.

### 2.3 Noise Detection

In terms of salt and pepper noise, true signals like edge details are removed if we use the median filtering to process all of the pixels in the image, which may cause blurring. To alleviate this problem, extremum detection is employed to determine whether a pixel is a noise point before filtering. In the extremum detection, the distinct criterion of the true signal  $S$  and the noise  $N$  is given as follows: for a given image, if the grayscale value of a pixel is the maximum or minimum of its neighborhood pixels, then the point is regarded as a noise; otherwise, it is a true signal. This is illustrated by equation (6).

$$x_{ij} \in \begin{cases} N & x_{ij} = \min(W[x_{ij}]), \max(W[X_{ij}]) \\ S & \min(W[x_{ij}]) < x_{ij} < \max(W[X_{ij}]) \end{cases} \quad (6)$$

According to the above criterion, all of the pixels in the image are first classified into two sets,  $N$  and  $S$ . Then, the median filtering is applied to process the pixels of category  $N$  while the pixels in category  $S$  are maintained. This can be described as follows:

$$Y_{ij} = \begin{cases} \text{median}(W[x_{ij}]), & x_{ij} \in N \\ x_{ij}, & x_{ij} \in S \end{cases} \quad (7)$$

### 3 Quantum Circuits of Modules

In this section, several modules used in this paper will be introduced. These modules are the Comparator module [40], Cycle Shift module [18], Swap module [36], Parallel Controlled-NOT module, Sort module, and Maximum-Median-Minimum module. The main function and corresponding quantum circuits of these modules are presented below.

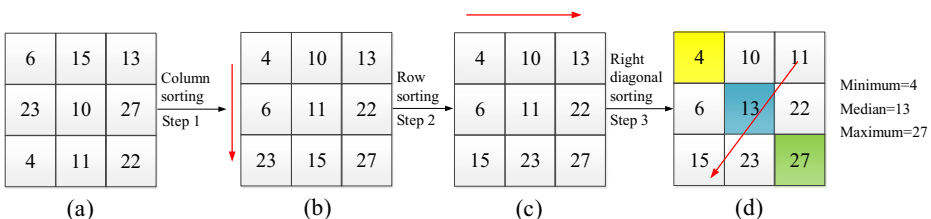


Fig. 3 An example verifying the correctness of Algorithm 1

### 3.1 Comparator Module

Comparator is used to compare two positive integers. This module was designed by Wang et al. [40]. In this paper, it is applied to compare grayscale values of two pixels. In order to compare two grayscale values of  $C_{YX}$  and  $C_{Y'X'}$ , where  $C_{YX} = C_{YX}^{q-1} C_{YX}^{q-2} \dots C_{YX}^1 C_{YX}^0$ ,  $C_{Y'X'} = C_{Y'X'}^{q-1} C_{Y'X'}^{q-2} \dots C_{Y'X'}^1 C_{Y'X'}^0$ ,  $C_{YX}^i, C_{Y'X'}^i \in \{0, 1\}$ , and  $i = q - 1, q - 2, \dots, 0$ . Its quantum circuit is shown in Fig. 4.

As shown in Fig. 4, the output qubits in the Comparator module  $|e_1\rangle$  and  $|e_0\rangle$  are used to denote the comparison results as follows:

- if  $e_1 e_0 = 00$ , then  $C_{YX} = C_{Y'X'}$ ;
- if  $e_1 e_0 = 01$ , then  $C_{YX} < C_{Y'X'}$ ;
- if  $e_1 e_0 = 10$ , then  $C_{YX} > C_{Y'X'}$ .

### 3.2 Cycle Shift Module

Le et al. [18] designed a quantum image Cyclic Shift transformation that can shift the whole image along the X-axis or Y-axis. In this paper, this module is used to obtain neighborhood pixels of the pixel point (Y, X). The transformation expressions on an NEQR image with the size of  $2^n \times 2^n$  are shown in equation (8). For their corresponding quantum circuits and detailed description, we kindly refer readers to reference [18]. Figure 5 presents an example demonstrating the operations of  $CS_{x+}$ ,  $CS_{x-}$ ,  $CS_{y+}$ ,  $CS_{y-}$ .

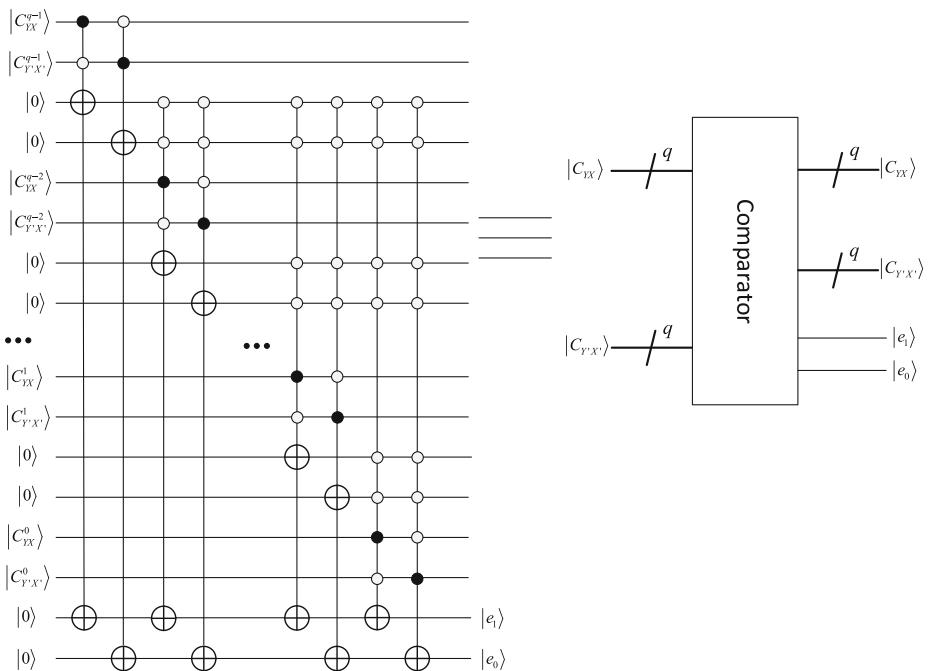


Fig. 4 Quantum circuit for the Comparator module

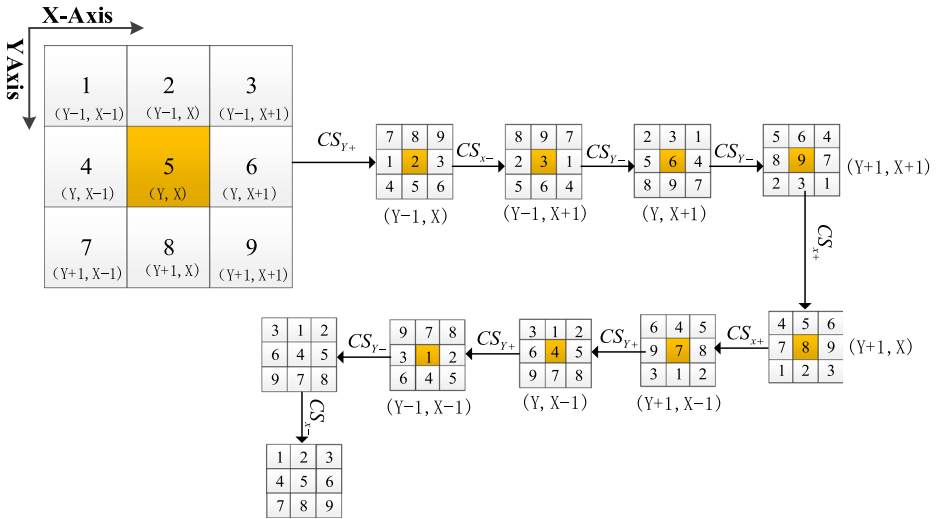


Fig. 5 An example demonstrating the Cycle Shift operation

$$\begin{aligned}
 CS_{X+}|I\rangle &= \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX'}\rangle|Y\rangle|(X+1)\bmod 2^n\rangle, X' = (X-1)\bmod 2^n \\
 CS_{X-}|I\rangle &= \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX'}\rangle|Y\rangle|(X-1)\bmod 2^n\rangle, X' = (X+1)\bmod 2^n \\
 CS_{Y+}|I\rangle &= \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{Y'X}\rangle|(Y+1)\bmod 2^n\rangle|X\rangle, Y' = (Y-1)\bmod 2^n \\
 CS_{Y-}|I\rangle &= \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{Y'X}\rangle|(Y-1)\bmod 2^n\rangle|X\rangle, Y' = (Y+1)\bmod 2^n
 \end{aligned}
 \tag{8}$$

As shown in Fig. 5, neighborhood pixels are acquired after a sequence of Cycle Shift operations,  $CS_{Y+}$ ,  $CS_{X-}$ ,  $CS_{Y-}$ ,  $CS_{X+}$ ,  $CS_{Y+}$ ,  $CS_{X-}$ ,  $CS_{Y-}$ ,  $CS_{X+}$ , which includes the nine pixels of  $(Y-1, X)$ ,  $(Y-1, X+1)$ ,  $(Y, X+1)$ ,  $(Y+1, X+1)$ ,  $(Y+1, X)$ ,  $(Y+1, X-1)$ ,  $(Y, X-1)$ ,  $(Y-1, X-1)$ , and  $(Y, X)$ . Additionally, the original image must be restored through successive shifts of  $CS_{Y-}$  and  $CS_{X-}$ .

### 3.3 Swap Module

According to Ref. [36], the function of this module is to swap two non-negative integers. In this paper, it is used to exchange two grayscale values. For example, the qubit  $|C_{YX}\rangle$  stores the grayscale value  $C_{YX}^{q-1} C_{YX}^{q-2} \dots C_{YX}^1 C_{YX}^0$  and the qubit  $|C_{Y'X'}\rangle$  stores the other grayscale value  $C_{Y'X'}^{q-1} C_{Y'X'}^{q-2} \dots C_{Y'X'}^1 C_{Y'X'}^0$ . After the module is executed, the values of  $C_{YX}$  and  $C_{Y'X'}$  are interchanged. Its quantum circuit is shown in Fig. 6.

### 3.4 Parallel Controlled-NOT Module

Inspired by Ref. [14], in this paper, this module is used to make a copy of the grayscale value  $|C_{YX}\rangle = |C_{YX}^{q-1} C_{YX}^{q-2} \dots C_{YX}^1 C_{YX}^0\rangle$  into the auxiliary qubits  $|0\rangle^{\otimes q}$ . This module consists of  $q$  Controlled-NOT gates, as illustrated in Fig. 7.



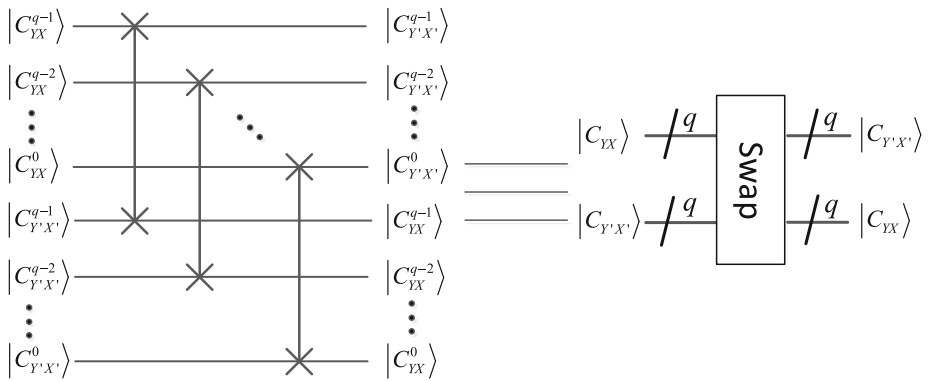


Fig. 6 Quantum circuit for the Swap module

### 3.5 Sort Module

The function of this module is to sort three integers to obtain the maximum, median, and minimum of them. To this end, three Comparator modules and three Swap modules are exploited. Figure 8 illustrates an example sorting of three integers of  $A$ ,  $B$ , and  $C$ . First, we use the Comparator to compare  $A$  and  $B$ , and then we determine whether to swap  $A$  and  $B$  depending on the value of  $e_1e_0$ . If it is 01, then swap  $A$  and  $B$  using the Swap module; else, do nothing. After this process,  $A$  is larger than  $B$ . Later, the operation is used to perform sorting of  $A$  and  $C$ , and it is also used to perform sorting of  $B$  and  $C$ . The outputs of this module are denoted as  $A'$ ,  $B'$ , and  $C'$  respectively, in which  $A'$  is the maximum,  $B'$  is the median, and  $C'$  is the minimum.

### 3.6 Maximum-Median-Minimum Module

The function of this module is to calculate the maximum, median, and minimum of neighborhood pixels. For this end, Ref. [37] employed bubble sorting using 30 Comparator modules and 30 Swap modules. Based on the analysis presented in Section 2.2, we design an improved quantum circuit utilizing lesser basic modules than that used in Ref. [37]. Figure 9 presents the quantum circuit of this module, which consists of seven Sort modules. According to the circuit of the Sort module, it is easy to see that there are 21 Comparator modules and 21 Swap

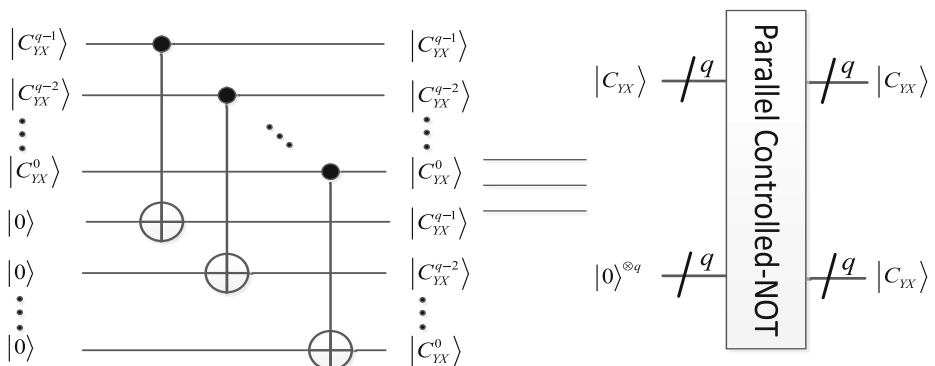


Fig. 7 Quantum circuit for the Parallel Controlled-NOT module

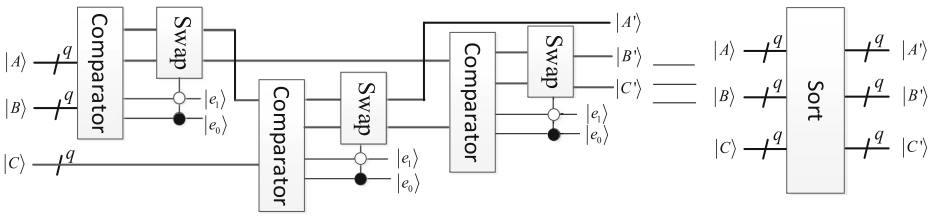


Fig. 8 Quantum circuit for the Sort module

modules. On this point, this approach is superior to Ref. [37]. As shown in Fig. 8, the inputs of  $|C_{Y-1,X}\rangle$ ,  $|C_{Y-1,X+1}\rangle$ ,  $|C_{Y,X+1}\rangle$ ,  $|C_{Y+1,X+1}\rangle$ ,  $|C_{Y+1,X}\rangle$ ,  $|C_{Y+1,X-1}\rangle$ , and  $|C_{Y,X-1}\rangle$ ,  $|C_{Y-1,X-1}\rangle$ ,  $|C_{Y,X}\rangle$  encode the neighborhood pixels, and the outputs of  $|Max\rangle$ ,  $|Med\rangle$  and  $|Min\rangle$  denote the maximum, median, and minimum of the neighborhood pixels, respectively. Here, it should be noted that other garbage qubits of the output are ignored.

### 4 Improved Quantum Image Median Filtering

In this section, the operation process of the proposed scheme is explained in detail. Here, it is assumed that the size of images is  $2^n \times 2^n$  and the grayscale range is  $2^q$ . Without loss of generality, this scheme adopts the  $3 \times 3$  filtering window. The workflow of this scheme is given first, and then we give a detailed description and step-by-step design of the circuit. Finally, the complete quantum circuit design is presented.

#### 4.1 Scheme Workflow

For this scheme, the original image is first converted to the quantum version based on NEQR, and then we use the Cycle Shift module and Parallel Controlled-NOT module to turn the neighborhood pixels into  $9q$  ancillary qubits. Next, we utilize extremum detection to determine the noise points, and perform median filtering on the noise points. Finally, we restore the filtered classical image through quantum measurements. The workflow of this scheme is presented in Fig. 10.

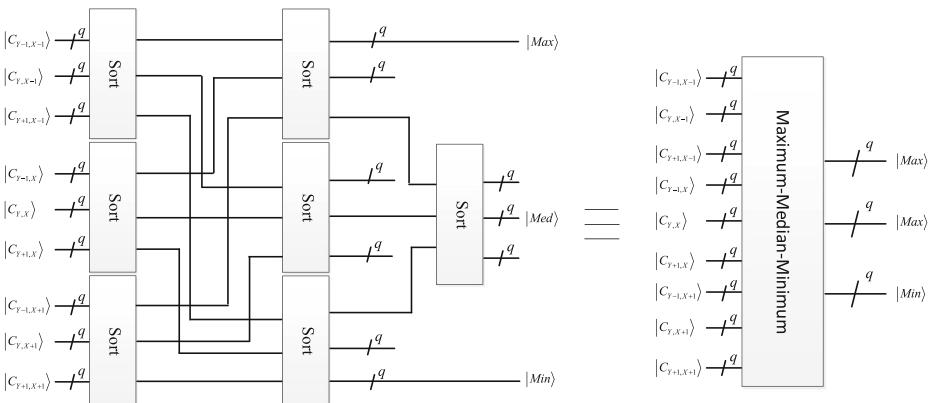


Fig. 9 Quantum circuit for the Maximum-Median-Minimum module

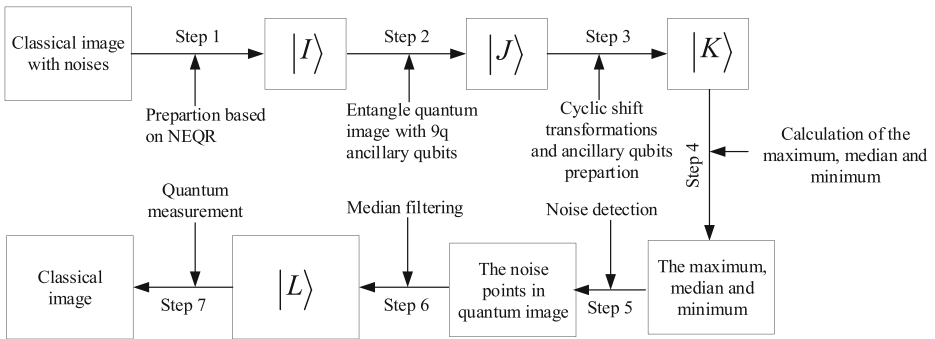


Fig. 10 Workflow of the improved quantum image median filtering

### 4.2 Step-By-Step Description of the Algorithm

Here we provide a more explicit explanation of the scheme presented in Fig. 10. The proposed scheme can be broken down into the following seven steps:

Step 1. We convert the classical grayscale image with noises to the quantum version  $|I\rangle$  based on NEQR, which encodes the grayscale value into eight qubits denoted as  $|C_{YX}\rangle = \bigotimes_{i=0}^7 |C_{YX}^i\rangle$ . The NEQR expression of a gray image with the size of  $2^n \times 2^n$  is written as

$$\begin{aligned}
 |I\rangle &= \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX}\rangle |YX\rangle \\
 &= \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX}^7 C_{YX}^6 \cdots C_{YX}^0\rangle |YX\rangle,
 \end{aligned}
 \tag{9}$$

where  $|YX\rangle = |y_{n-1}y_{n-2}\cdots y_1y_0\rangle |x_{n-1}x_{n-2}\cdots x_1x_0\rangle$ ,  $y_i, x_i \in \{0, 1\}$ . The first  $n$ -qubit  $|y_{n-1}y_{n-2}\cdots y_1y_0\rangle$  encodes the vertical coordinates, and the second  $n$ -qubit  $|x_{n-1}x_{n-2}\cdots x_1x_0\rangle$  encodes the horizontal coordinates.

Step 2. Nine  $|0\rangle^{\otimes 9}$  ancillary qubits are prepared to store the neighborhood pixels. To take full advantage of the parallelism of quantum computation, the ancillary qubits are the tensor product with the original quantum image states, expressed as follows:

$$\begin{aligned}
 |J\rangle &= |0\rangle^{\otimes 9q} \otimes |I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |0\rangle^{\otimes 9q} |C_{YX}\rangle |YX\rangle \\
 &= \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |0\rangle^{\otimes q} \cdots |0\rangle^{\otimes q} |C_{YX}\rangle |YX\rangle
 \end{aligned}
 \tag{10}$$

Step 3. We prepare the nine  $|0\rangle^{\otimes 9}$  ancillary qubits with the neighborhood pixel values through a sequence of Cyclic Shift transformations and Parallel Controlled-NOT operations on  $|J\rangle$ . First, ten Cycle Shift modules are used to obtain the neighborhood pixel values, and then nine Parallel Controlled-NOT modules are exploited to

encode these pixel values into the qubits  $|0\rangle^{\otimes q}$ . To simplify the design of the complete quantum circuits of this work, the function of this step is defined as a module, which we call the Neighborhood Preparation module. The resulting image  $|K\rangle$  is represented by equation (11), and the corresponding quantum circuit design is presented in Fig. 11.

$$|K\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{Y-1,X}\rangle \otimes |C_{Y-1,X+1}\rangle \otimes |C_{Y,X+1}\rangle \otimes |C_{Y+1,X+1}\rangle \otimes |C_{Y+1,X}\rangle \otimes |C_{Y+1,X-1}\rangle \otimes |C_{Y,X-1}\rangle \otimes |C_{Y-1,X-1}\rangle \otimes |C_{Y,X}\rangle \otimes |C_{Y,X}\rangle |Y\rangle |X\rangle \quad (11)$$

In equation (11), the nine auxiliary qubits of  $|C_{Y-1,X}\rangle, |C_{Y-1,X+1}\rangle, |C_{Y,X+1}\rangle, |C_{Y+1,X+1}\rangle, |C_{Y+1,X}\rangle, |C_{Y+1,X-1}\rangle, |C_{Y,X-1}\rangle, |C_{Y-1,X-1}\rangle, |C_{Y,X}\rangle$ , store the nine neighborhood pixels of  $|C_{Y,X}\rangle$ .

- Step 4. In this step we calculate the maximum, median, and minimum of the neighborhood pixels using the Maximum-Median-Minimum module. The maximum and minimum are the inputs for noise detection in Step 5. The median is the input for median filtering in Step 6.
- Step 5. This step is devoted to noise detection, as is depicted as a Noise Detection module. As discussed in Section 2.3, filtering is performed only on the detected noise points rather than on all pixel points; this allows us to avoid the blurring issue caused by the standard median filtering. In the proposed approach, if the pixel value is same to the maximum or minimum of the neighborhood pixels, this pixel will be determined as a noise point. Its quantum circuit is shown in Fig. 11, in which some garbage output qubits are ignored for circuit simplicity. As denoted in Fig. 12, the output qubit  $|e_{20}\rangle$  is essential to this module, which encodes the result of noise detection. If  $e_{20}=1$ , then the current pixel is a noise point; otherwise it is not a noise point.
- Step 6. According to Step 5, if  $e_{20}=1$ , the current pixel value is the maximum or minimum of its neighborhood pixels, that is to say, it is a noise point and it will be replaced by the median of its neighborhood pixels through a Swap module.

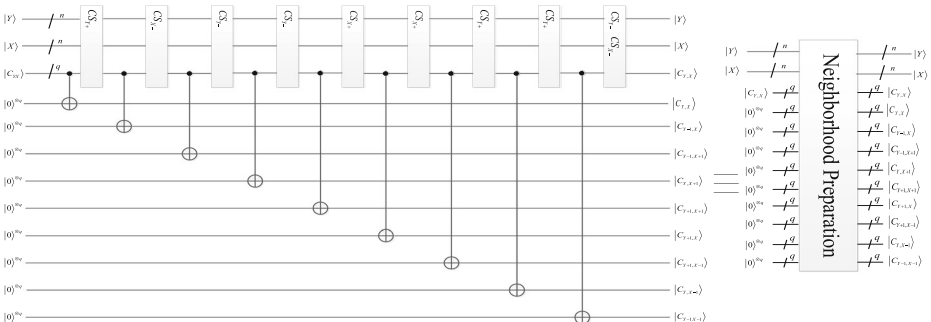


Fig. 11 Quantum circuit for the preparation of the ancillary qubits

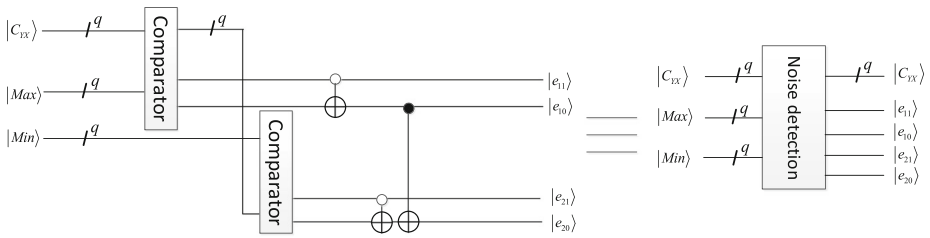


Fig. 12 Quantum circuit for the Noise Detection module

Step 7. The last step of image processing is usually image retrieval. As is well known, the quantum measurement is the only way to recover the classical image from the quantum version, and it is still difficult to convert the quantum image to the classical image by one measurement. However, image filtering is usually used as a pre-processing stage for other image processing algorithms, such as those employed in image feature extraction and segmentation. That is to say, image retrieval is not necessary after filtering. The result of filtering can be used as input to the next quantum image processing algorithms. To conclude, the quantum measurement can be regarded as an unnecessary step. If it is necessary to perform image retrieval, we kindly refer readers to reference [10], which presents a detailed retrieval approach.

### 4.3 Complete Quantum Circuit of the Proposed Scheme

According to the principle of the proposed filtering method and the algorithm description in Section 4.2, the complete quantum circuit is shown in Fig. 13. The circuit is composed of one Neighborhood Preparation module, one Maximum-Median-Minimum module, one Noise Detection module, and one Swap module.

The output of  $|\hat{C}_{YX}\rangle$  encodes the pixel values in the image after the filtering process, and the filtered image can be represented as follows:

$$|L\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |\hat{C}_{YX}\rangle |YX\rangle. \tag{12}$$

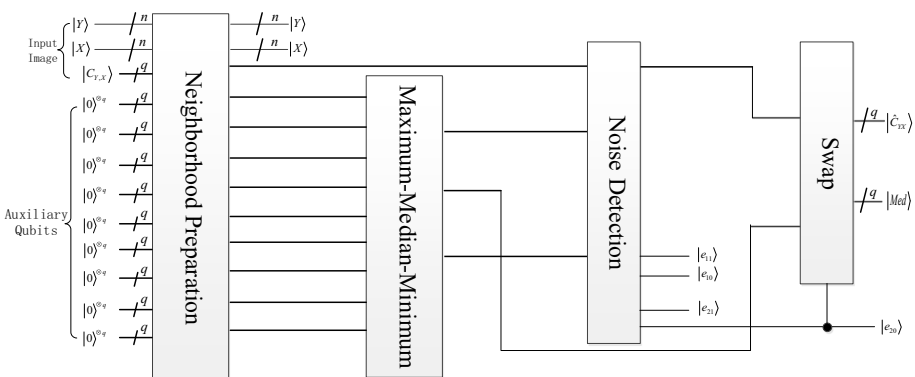


Fig. 13 Complete circuit for the improved quantum image median filtering

## 5 Circuit Complexity and Simulation Experiments Analysis

In this section, we will first discuss the circuit complexity of the proposed method. Then, we analyze the results of simulation experiments and compare them those of other image filtering algorithms.

### 5.1 Circuit Complexity

In quantum image processing, the network complexity lies mainly on the number of elementary gates used [41], such as the NOT gate, Hadamard gate, Controlled-NOT gate, and other basic unitary operators. Here, the complexity analysis will be discussed by module first, and the complexity of the entire quantum circuit is calculated afterward. It needs to be noted that the complexity of quantum image preparation is not considered.

- (1) Comparator module. According to the analysis in [41], the computational complexity of this module is no more than  $O(q^2)$ . For the detailed calculation, readers can refer to [41].
- (2) Cycle Shift module. For an image with a size of  $2^n \times 2^n$ , the Cycle Shift transformation will cost a time of  $O(n^2)$  [15].
- (3) Swap module. As is well known, a swap gate can be broken down into three Controlled-NOT gates. Hence, the quantum Swap module requires  $3q$  Controlled-NOT gates and will cost a time of  $O(3q)$ .
- (4) Quantum Parallel Controlled-NOT module. As seen in Fig. 7, this module consists of  $q$  Controlled-NOT gates. Hence, the complexity of this module is  $O(q)$ .
- (5) Sort module. As shown in Fig. 8, this module is composed of three Comparator modules and three Swap modules. Hence, the complexity of the module is  $O(3q^2 + 9q)$ .
- (6) Preparation Neighborhood module. As shown in Fig. 11, ten Cycle Shift modules and nine Parallel Controlled-NOT modules are employed to form the Preparation Neighborhood module. Hence, the complexity of this module is  $O(10n^2 + 9q)$ .
- (7) Maximum-Median-Minimum module. As seen in Fig. 9, this module consists of seven Sort modules. According to the analysis of equation (5), this module will cost a time of  $O(21q^2 + 63q)$ .
- (8) Noise Detection module. As presented in Fig. 12, this module is made up of two Comparator modules and three basic logical gates. Hence, this module's complexity is  $O(6q)$ .

Finally, as shown in Fig. 13, the entire quantum circuit consists of four modules: Neighborhood Preparation module, Maximum-Median-Minimum module, Noise Detection module, and Swap module. According to equations (1) to (8), it is easy to find that the complexity of the whole quantum circuit is no more than  $O(10n^2 + 9q + 21q^2 + 63q + 9q + 3q) \approx O(10n^2 + 21q^2)$ . At this point, the investigated scheme is superior to that proposed in Ref. [37].

### 5.2 Simulation Experiments and Analysis

Since a practical and useful quantum computer is not in our grasp right now, the simulation experiments are performed on a classical computer with an Intel (R) Core (TM) i5-7200U CPU@2.70GHz 8.00GB RAM, 64-bit operating system, and MATLAB 2016b. To simulate the Hilbert space, quantum states are denoted by vectors, and unitary transforms are denoted by unitary matrices. A detailed analysis of the experimental results is presented below.

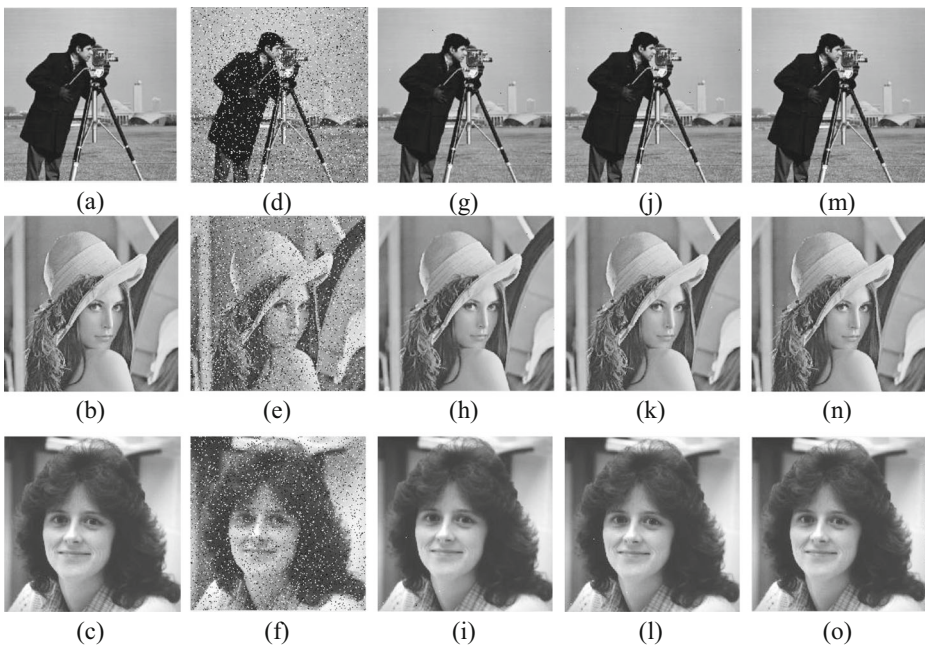
The three gray images named “Cameraman,” “Lena,” and “Woman\_darkhair” are used in the experiment and the size of each image is  $512 \times 512$ . The filtering effects for salt and pepper noise are given in Fig. 14.

As illustrated in Fig. 14, there are few differences between the classical environment and quantum environment for the proposed method. However, the filtering effect of the proposed method is better than that of [37]. To illustrate the visual effectiveness of the filtered image, the peak signal-to-noise ratio (PSNR) is used:

$$PSNR = 10\log_{10} \left( \frac{255^2}{\frac{1}{2^{2n}} \sum_{i=0}^{2^n} \sum_{j=0}^{2^n} [I_{ori}(i, j) - I_{NF}(i, j)]^2} \right), \quad (13)$$

where  $I_{ori}$  and  $I_{NF}$  denote the original image and the noise or filtered image, respectively. PSNR values of the noise images and filtered image are shown Table 1.

From Table 1, we can see that the proposed filtering approach is more effective than standard median filtering used in [37]. Additionally, compared to the classical counterpart, the PSNR obtained by the proposed method is larger modestly. However, according to the network complexity analysis presented in Section 5.1, the computational complexity of the proposed filtering approach is the second-order polynomial function of image size  $n$ , which can achieve an exponential speed-up than the classical counterpart with a time complexity of  $O(2^{2n})$ .



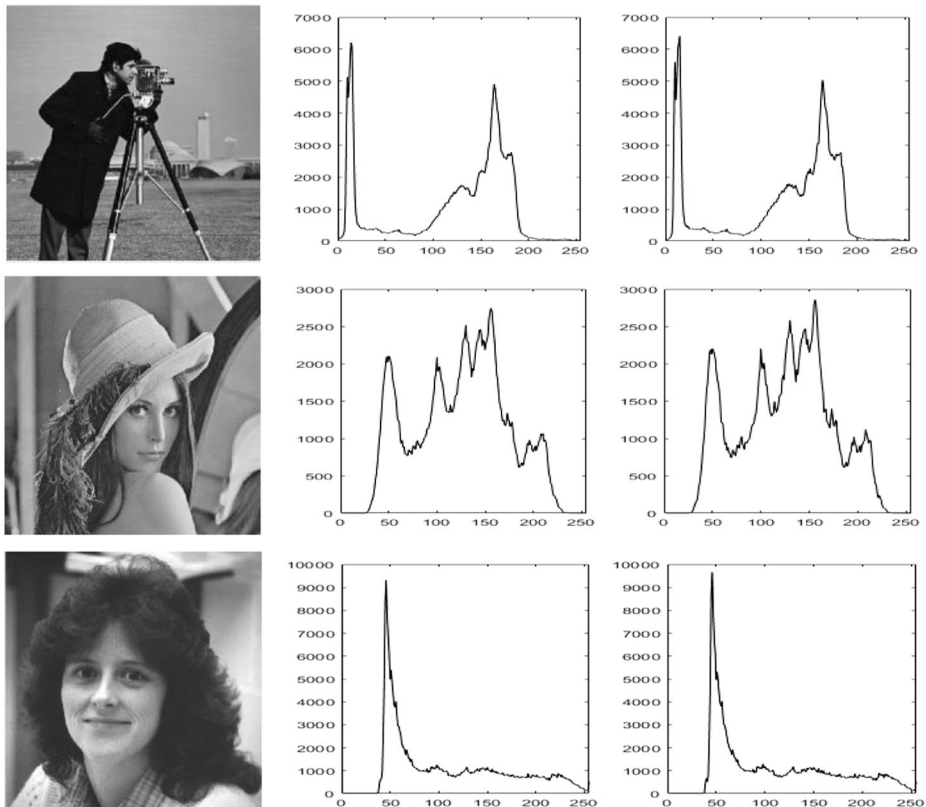
**Fig. 14** De-noising effects of filtering on salt and pepper noise with a density of 0.10. (a–c) Original images, (d–f) images after superimposing the salt and pepper noise, (g–i) filtering results for (d–f) using the method proposed in [37], (j–l) filtering results for (d–f) using the proposed scheme, and (m–o) respective filtering results for (d–f) on the classical counterpart

**Table 1** PSNR values of the noise image and filtered images (dB)

Test images	Noise image	Reference [37]	Proposed method	Classical counterpart
Cameraman	15.0900	34.3407	38.6905	37.4579
Lena	15.4662	33.6405	37.6166	36.5337
Woman_darkhair	14.9723	38.3772	42.0368	39.2694

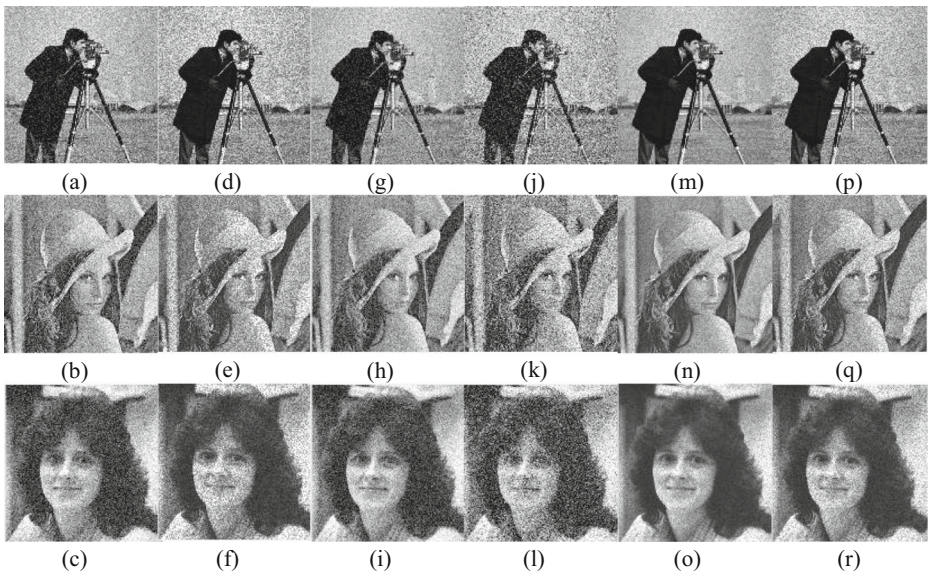
Figure 15 shows histograms of the three original images and the corresponding filtered images using the proposed scheme. It is clear that the proposed filtering method does not cause much damage to the quality of the original image.

According to the description of the median filtering in Section 2.2, the main function of the median filtering is to replace those pixels that are of significant difference from the surrounding pixels with values closest to those of the surrounding pixels. Therefore, the median filtering is efficient for removing isolated noise, such as salt and pepper noise. For other noises, such as Gaussian noise and Speckle noise, the result of filtering is that some noises fail to be removed, as shown in Fig. 16.



**Fig. 15** Histograms of the images: histograms of the original images are depicted in the second column; and histograms of the filtered images are depicted in the third column





**Fig. 16** De-noising effects of filtering on Gaussian noise and Speckle noise. **a–c** images after superimposing the Gaussian noise with mean 0 and variance of 0.05, **(d–f)** images after superimposing the Speckle noise with mean 0 and variance of 0.05, **(g–i)** filtering results for **(a–c)** using the standard median filtering [37], **(j–l)** filtering results for **(a–c)** using the proposed scheme, **(m–o)** filtering results for **(d–f)** using the standard median filtering [37], and **(p–r)** filtering results for **(d–f)** using the proposed scheme

As illustrated in Fig. 16, for Gaussian noise and Speckle noise, neither the standard median filtering [37] nor the proposed scheme can remove noises completely. Therefore, the median filtering is not appropriate to remove these two kinds of noise.

## 6 Conclusions

Image filtering has been widely applied in image processing. It is used to eliminate noises to improve the quality of images, and is considered as a significant pre-processing step before other image processing algorithms are used, such as those utilized for image feature extraction and segmentation. This paper presents an improved quantum image median filtering in the spatial domain, implemented mainly to remove isolated noise such as salt and pepper noise. As is well known, the median filtering can effectively filter these impulse noises, but if all pixels in the image are processed using the median filtering, the original image can be damaged. Hence, this scheme adopts extremum detection to distinguish noises and true signals as much as possible before using the filtering operation. Moreover, we design a more efficient quantum circuit for obtaining the median of the neighborhood pixels; this circuit uses less quantum elementary gates than other existing methods.

**Acknowledgments** This work is supported by National Key R&D Plan under Grant No. 2018YFC1200200; the National Natural Science Foundation of China under Grant No.61463016 and No.61763014; Science and technology innovation action plan of Shanghai in 2017 under Grant No.17510740300.

## References

1. Feynman, R.P.: Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982)
2. Nagy, M., Akl, S.G.: Quantum computation and quantum information. *Int. J. Parallel, Emergent Distrib. Syst.* **21**, 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**, 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. Venegas-Andraca, S., Bose, S.: Storing, processing, and retrieving an image using quantum mechanics. In: *Proceedings of SPIE Conference of Quantum Information and Computation*. 5105(8), 134–147(2003)
7. Venegas-Andraca, S., Ball, J.: Processing images in entangled quantum systems. *Quantum Inf. Process.* **9**(1), 1–11 (2010)
8. Latorre, J.: Image Compression and Entanglement. arXiv:quant-ph/0510031 (2005)
9. Le, P., Dong, F., Hirota, K.: A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* **10**, 63–84 (2011)
10. Zhang, Y., Lu, K., Gao, Y., Mao, W.: NEQR: a novel enhanced quantum representation of digital images. *Quantum Inf. Process.* **12**, 2833–2860 (2013)
11. Zhang, Y., Lu, K., Gao, Y., Xu, K.: A novel quantum representation for log-polar images. *Quantum Inf. Process.* **12**, 3103–3126 (2013)
12. Jiang, N., Wang, L.: Quantum image scaling using nearest neighbor interpolation. *Quantum Inf. Process.* **14**, 1559–1571 (2015)
13. Zhou, R., Hu, W., Liu, X., Fan, P., Luo, G.: Quantum realization of the nearest neighbor value interpolation method for INEQR. *Quantum Inf. Process.* **17**, 1–37 (2018)
14. Zhou, R.G., Hu, W.W., Fan, P., Ian, H.: Quantum realization of the bilinear interpolation method for NEQR. *Sci. Rep.* **7**(2511), (2017)
15. Zhang, Y., Lu, K., Xu, K., Gao, Y., Wilson, R.: Local feature point extraction for quantum images. *Quantum Inf. Process.* **14**, 1573–1588 (2015)
16. Zhang, Y., Lu, K., et al.: QSobel: a novel quantum image edge extraction algorithm. *Sci. China Inf. Scis.* **58**, 1–13 (2015)
17. Zhou, R.G., Tan, C.Y., Ian, H.: Global and local translation designs of quantum image based on FRQI. *Int. J. Theor. Phys.* **56**, 1382–1398 (2017)
18. Le, P.Q., Iliyasu, A.M., Dong, F., et al.: Strategies for designing geometric transformations on quantum images. *Theor. Comput. Sci.* **412**, 1406–1418 (2011)
19. Le, P.Q., Iliyasu, A.M., Dong, F., et al.: Fast geometric transformations on quantum images. *IAENG Int. J. Appl. Math.* **40**, (2010)
20. Luo, G.F., Zhou, R.G., Liu, X.A., Hu, W.W., Luo, J.: Fuzzy matching based on gray-scale difference for quantum images. *Int. J. Theor. Phys.* **57**, 2447–2460 (2018)
21. Jiang, N., Dang, Y.J., Wang, J.: Quantum image matching. *Quantum Inf. Process.* **15**, 3543–3572 (2016)
22. Dang, Y., Jiang, N., Zhang, W., Hu, H.: analysis and improvement of the quantum image matching. *Quantum Inf. Process.* **16**, 1–13 (2017)
23. Li, H.S., Zhu, Q.X., Lan, S., et al.: Image storage, retrieval, compression and segmentation in a quantum system. *Quantum Inf. Process.* **12**, 2269–2290 (2013)
24. Caraiman, S., Manta, V.I.: Image segmentation on a quantum computer. *Quantum Inf. Process.* **14**, 1693–1715 (2015)
25. Zhou, R.G., Hu, W.W., Fan, P., Luo, G.F.: Quantum color image watermarking based on Arnold transformation and LSB steganography. *Int. J. Quantum Inf.* **16**, 1850021 (2018)
26. Zhou, R.G., Hu, W.W., Fan, P.: Quantum watermarking scheme through Arnold scrambling and LSB steganography. *Quantum Inf. Process.* **16**(212), (2017)
27. Yang, Y.G., Wang, Y., Zhao, Q.Q.: Letter to the editor regarding “dynamic watermarking scheme for quantum images based on Hadamard transform” by song et al. *Multimedia Syst.* **22**, 271–272 (2016)
28. Yang, Y.G., Jia, X., Xu, P., Tian, J.: Analysis and improvement of the watermark strategy for quantum images based on quantum Fourier transform. *Quantum Inf. Process.* **12**, 2765–2769 (2013)
29. Heidari, S., Naseri, M.: A novel LSB based quantum image watermarking. *Int. J. Theor. Phys.* **55**, 4205–4218 (2016)
30. Heidari, S., Farzadnia, E.: A novel quantum LSB-based steganography method using the gray code for colored quantum images. *Quantum Inf. Process.* **16**, 1–28 (2017)

31. Jiang, N., Wang, L.: A novel strategy for quantum image steganography based on Moir pattern. *Int. J. Theor. Phys.* **54**, 1021–1032 (2015)
32. Jiang, N., Zhao, N., Wang, L.: LSB based quantum image steganography algorithm. *Int. J. Theor. Phys.* **55**, 107–123 (2016)
33. Lomont, C.: Quantum convolution and quantum correlation algorithms are physically impossible arXiv: quant-ph/0309070 (2003)
34. Caraiman, S., Manta, V.I.: Quantum image filtering in the frequency domain. *Adv. Electr. Comput. Eng.* **13**, 77–84 (2013)
35. Yuan, S., Lu, Y., Mao, X., Luo, Y., Yuan, J.: Improved quantum image filtering in the spatial domain. *Int. J. Theor. Phys.* **57**, 804–813 (2018)
36. Li, P., Liu, X., Xiao, H.: Quantum image weighted average filtering in spatial domain. *Int. J. Theor. Phys.* **56**, 3690–3716 (2017)
37. Li, P., Liu, X., Xiao, H.: Quantum image median filtering in the spatial domain. *Quantum Inf. Process.* **17**(49), (2018)
38. Li, P., Xiao, H.: An improved filtering method for quantum color image in frequency domain. *Int. J. Theor. Phys.* **57**, 258–278 (2018)
39. Han, T.: Research of fast median filtering algorithm and hardware implementation based on FPGA. *Chinese Journal of Electron Devices.* **40**, 697–701 (2017)
40. Wang, D., University, H: Kaifeng: Design of Quantum Comparator Based on extended general Toffoli gates with multiple targets. *Comput. Sci.* **39**, 302–306 (2012)
41. Wang, J., Jiang, N., Wang, L.: Quantum image translation. *Quantum Inf. Process Mathematic*, 1589–1604 (2015), 14

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.