SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2015, Vol. 58 012106:1–012106:13
doi: 10.1007/s11432-014-5158-9

QSobel: A novel quantum image edge extraction algorithm

ZHANG Yi[1,2], LU Kai[1,2*] & GAO YingHui[3]

[1]*Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha* 410073*, China;*
[2]*College of Computer, National University of Defense Technology, Changsha* 410073*, China;*
[3]*College of Electronic Science and Engineering, National University of Defense Technology, Changsha* 410073*, China*

Received September 23, 2013; accepted March 11, 2014; published online December 1, 2014

Abstract Edge extraction is an indispensable task in digital image processing. With the sharp increase in the image data, real-time problem has become a limitation of the state of the art of edge extraction algorithms. In this paper, QSobel, a novel quantum image edge extraction algorithm is designed based on the flexible representation of quantum image (FRQI) and the famous edge extraction algorithm Sobel. Because FRQI utilizes the superposition state of qubit sequence to store all the pixels of an image, QSobel can calculate the Sobel gradients of the image intensity of all the pixels simultaneously. It is the main reason that QSobel can extract edges quite fast. Through designing and analyzing the quantum circuit of QSobel, we demonstrate that QSobel can extract edges in the computational complexity of $O(n^2)$ for a FRQI quantum image with a size of $2^n \times 2^n$. Compared with all the classical edge extraction algorithms and the existing quantum edge extraction algorithms, QSobel can utilize quantum parallel computation to reach a significant and exponential speedup. Hence, QSobel would resolve the real-time problem of image edge extraction.

Keywords edge extraction, quantum image processing, FRQI, Sobel, computational complexity

Citation Zhang Y, Lu K, Gao Y H. QSobel: A novel quantum image edge extraction algorithm. Sci China Inf Sci, 2015, 58: 012106(13), doi: 10.1007/s11432-014-5158-9

1 Introduction

As a novel computing model, quantum computation can help store, process, and transfer information using the unique properties of quantum mechanics [1] such as the superposition state and the entangled state. Since 1982 when Feynman first proposed this concept [2], quantum computation has become a hot science topic. In recent years, we have witnessed significant theoretical improvements and some encouraging experimental results in this field. Especially, in 1994, Shor [3] designed a quantum integer factoring algorithm to find the secret key encryption of RSA (R. Rivest, A. Shamiv and L. Adleman) in polynomial time; meanwhile, in 1996, Grover [4] explored a quantum search algorithm for a database, which can reach quadratic speed. After these two most representative quantum algorithms, the excellent results motivated more researchers to explore the quantum world. Meanwhile quantum computation is considered to be a promising candidate to overcome the limitations of classical computing.

*Corresponding author (email: kailu@nudt.edu.cn)

Image edge extraction is an important technology of digital image processing [5]. It lies at the core of many applications such as pattern recognition, machine vision, feature detection and web retrieval. With the development of image sensors, the number of images has become huge and the size of images larger. Owing to the often overwhelming computational complexity of state-of-the-art algorithms, it is necessary to find better ways to store and process digital images with high accuracy and high real-time.

To cope with this problem of image edge extraction algorithms, the combination of quantum computation with image processing has been extensively investigated in the past decades. There are some researches on how to use quantum mechanics to store and process digital images, such as qubit lattice [6,7], entangled image [8,9], real ket [10], and a flexible representation of quantum image (FRQI) [11]. Some quantum edge extraction algorithms [12,13] are proposed based on qubit lattice. Qubit lattice utilizes one qubit to store the color information of one pixel, therefore the qubit lattice-based edge extraction algorithms have better accuracy and better robustness for noise. However, these algorithms cannot reduce the computational complexity, because qubit lattice does not utilize the parallel superposition state of quantum mechanics to store images.

In this paper, QSobel, a novel quantum image edge extraction algorithm based on quantum image model of FRQI is proposed. Because FRQI uses the superposition state of qubit sequence to store all the pixels of an image, the same operations can be done on all the pixels simultaneously. QSobel utilizes this excellent property of FRQI to calculate the Sobel gradients of the image intensity of all the pixels quite fast. Therefore, QSobel can reach an exponential speedup compared to all the classical edge extraction algorithms and the existing quantum edge extraction algorithms. Hence, QSobel would extract edges with high real-time for the actual applications.

The rest of this paper is organized as follows: In Section 2, we discuss related work. Then Section 3 gives a brief introduction to Sobel edge extraction algorithm, which is a famous classical algorithm. The theory and detailed design of QSobel are presented in Section 4. Tests and performance analyses of QSobel are reported in Section 5. Finally, we draw conclusions and outline possible future research paths in Section 6.

2 Related work

Recently, the merger of quantum computation and digital image processing has proved to be very fruitful in dealing with the high accuracy and high real-time problems of the actual image processing applications.

There are four existing quantum image models to store and process images, including qubit lattice, entangled image, real ket, and FRQI. In the model of qubit lattice [6,7], every pixel will be stored in a single qubit and all the pixel operations will be changed into the relative quantum operations of one qubit. Every quantum image will be represented as a qubit matrix. Entangled image [8,9] is similar with qubit lattice, but it utilizes the entangled state of quantum mechanics to demonstrate the relationship of pixels of images. The model of real ket [10] makes the image quartering iteratively and builds a balanced quad-tree index. Every pixel will be mapped into a ground state of 4-dimensional index qubit sequence. FRQI was proposed by Le et al. [11]. This quantum image representation model is expressed as Eq. (1) for an image with size of $2^n \times 2^n$. In FRQI, the position information of every pixel is stored in a ground state of 2-dimensional qubit sequence, and the color information is stored in the probability amplitude of a qubit, which is entangled with the qubit sequence. Figure 1 shows an FRQI quantum image with size of 4×4.

$$|I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} (\cos\theta_{YX} |0\rangle + \sin\theta_{YX} |1\rangle) |YX\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX}\rangle |YX\rangle. \quad (1)$$

Qubit lattice and entangled image are the most similar to the classical digital images, and it is easy to find the quantum equivalent operations of the classical image processing ones. The models of real ket and FRQI utilize the superposition state of qubit sequence to store images, therefore they can process the information of all the pixels simultaneously. Compared with real ket, FRQI maintains a a pixel matrix

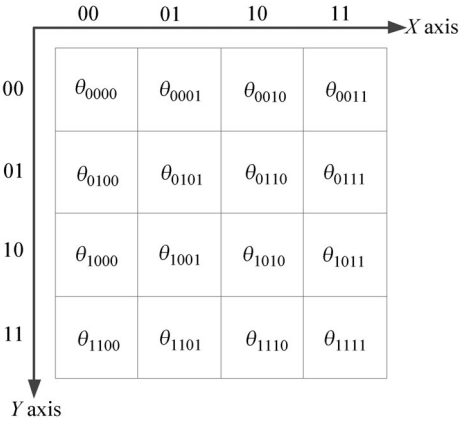

Figure 1 An FRQI quantum image with size of 4×4.

so that it is more suitable to design the quantum image processing algorithms, especially algorithms that are relative to the pixel neighborhood window.

Image edge extraction is an important issue of digital image processing. In an image, the edges are the pixels at which the color intensity of the image has discontinuities. Based on this property, there are many famous edge extraction algorithms such as Sobel [14], Prewitt [15], Kirsch [16], Canny [17], and wavelet-based edge detector [18]. In order to find all the discontinuous pixels, all the classical algorithms need to analyze and calculate the gradient of the image intensity of every pixel. Therefore, none of them can finish the whole work in the computational complexity of less than $O\left(2^{2n}\right)$ for an $2^n \times 2^n$ image. This is the main reason that the classical edge extraction algorithms will be low on real-time with the sharp increase of the image data.

By now, there are some quantum edge extraction algorithms; in general, all the existing algorithms are based on the model of qubit lattice. Tseng and Hwang [12] proposed an algorithm that combined Sobel algorithm and the model of qubit lattice, but it was just a quantum equivalent algorithm of Sobel and it has no performance improvements. Fu [13] utilized the theory of fuzzy entropy to extract edges for an enhanced qubit lattice image, and the proposed algorithm could show more weak edges and have better robustness for noise. Hence this quantum algorithm has a significant devotion to the medical image processing. Although these quantum edge extraction algorithms may obtain better results than the classical ones, they do not resolve the overwhelming computational complexity problem as the image data increases because of qubit lattice model. Their computational complexities remain as still $O\left(2^{2n}\right)$ for an $2^n \times 2^n$ image.

3 Classical Sobel edge extraction algorithm

Sobel algorithm is one of the famous and representative classical edge extraction algorithms. Many spinoffs and ramifications of Sobel algorithm have been applied widely. In this section, the theory of Sobel will be briefly introduced as an example of the classical edge extraction algorithms [5,14].

Mask computation is a common technology to calculate a rather inaccurate approximation of the gradient of the image intensity of every pixel, but it still sufficiently precise to be of practical use in most applications. Sobel algorithm utilizes the pixel neighborhood window to calculate approximations of the gradient of the image intensity. Figure 2(a) depicts the pixel neighborhood window. Sobel uses the masks shown in Figure 2(b) and Figure 2(c) to calculate the two approximations of the derivatives, one for horizontal changes, and one for vertical.

Firstly, Sobel will get approximations of the derivatives of X-axis and Y-axis, respectively. As shown in Eq. (2), G_x and G_y denote how to utilize the masks to calculate the two derivatives. Eq. (3) gives the relationship of the gradient g and the two derivatives. By comparing with a gradient threshold T,

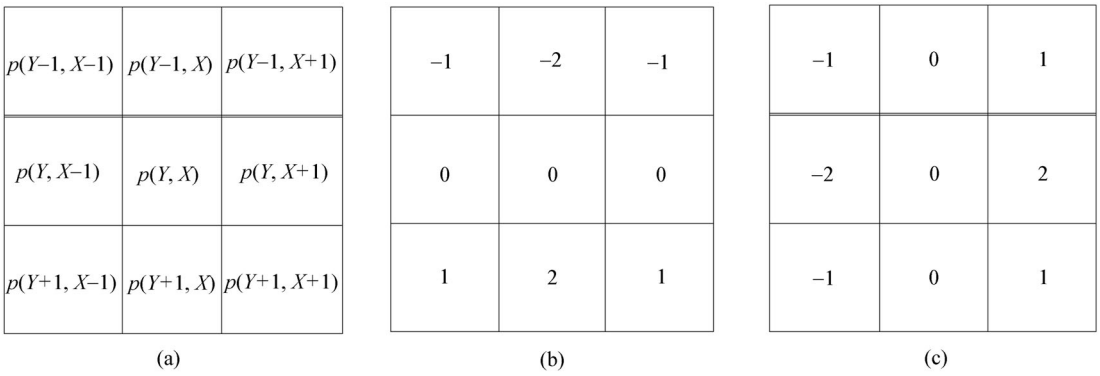

Figure 2 (a)The pixel neighborhood window; (b) and (c) are two masks of Sobel algorithm.

the pixel will be considered as a part of edge if $g \geqslant T$.

$$\begin{aligned} &G_x = (p(Y+1, X+1) + 2p(Y+1, X) + p(Y+1, X-1)) \\ &-(p(Y-1, X+1) + 2p(Y-1, X) + p(Y-1, X-1)), \\ &G_y = (p(Y+1, X+1) + 2p(Y, X+1) + p(Y-1, X+1)) \\ &-(p(Y+1, X-1) + 2p(Y, X-1) + p(Y-1, X+1)), \end{aligned} \tag{2}$$

$$g = [G_x^2 + G_y^2]^{1/2}. \tag{3}$$

The main procedure is to calculate the intensity gradient of each pixel using the Sobel masks. Because every pixel needs to be processed, the computational complexity of Sobel algorithm is $O\left(2^{2n}\right)$ for an $2^n \times 2^n$ image. Similar to Sobel, other classical edge extraction algorithms would utilize different masks or methods to calculate the approximation of the intensity gradients. Therefore, we can conclude that all the classical algorithms cannot extract edges for an $2^n \times 2^n$ image in computational complexity of less than $O\left(2^{2n}\right)$.

4 QSobel edge extraction algorithm

Based on Sobel edge extraction algorithm in section 3 and the quantum image model of FRQI, a novel quantum edge extraction algorithm, which is named QSobel is designed in this section. Firstly, we define two unitary operations of the shift transformations of FRQI. Then, the operations will be utilized to calculate the intensity gradient of every pixel in the FRQI image simultaneously. At last, the result of edge extraction will be stored in a new FRQI image.

4.1 *X*-Shift and *Y*-Shift transformations

4.1.1 *Definition*

From Eq. (1), it is known that the model of FRQI maintains a 2-dimensional pixel matrix. Therefore, if we shift the whole image, every pixel will access the information of its neighborhood simultaneously. For example, make a one-unit shift downwards for an image, the pixel will be transformed $p(Y, X)$ into $p(Y-1, X)$.

At the beginning, the two transformations will be defined as follows.

Definition 1. $U(x\pm)$: X-Shift transformation based on a FRQI image with a size of $2^n \times 2^n$ is defined in Eq. (4).

$$U(x\pm)\left|I\right\rangle = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\left|C_{YX}\right\rangle\left|Y\right\rangle\left|(X\pm 1) \bmod 2^n\right\rangle. \tag{4}$$

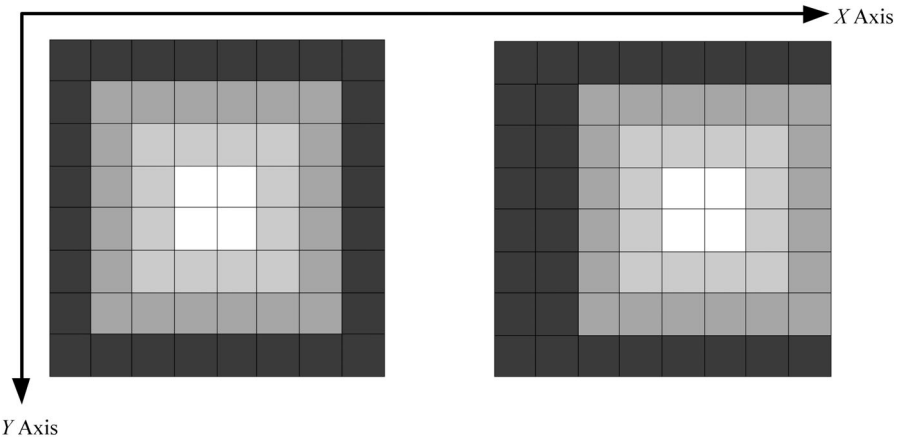

Figure 3 The example of the transformation $U(x+)$ for an image with size of 8×8.

By using X to replace $X \pm 1$, we can get the relationship as Eq. (5):

$$\sum_{X=0}^{2^n-1} |C_{YX}\rangle \, |(X \pm 1) \bmod 2^n\rangle = \sum_{X=0}^{2^n-1} |C_{YX'}\rangle \, |X\rangle, \tag{5}$$

where $X' = (X \mp 1) \bmod 2^n$, it means

$$U(x\pm) \, |I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX'}\rangle \, |Y\rangle \, |X\rangle. \tag{6}$$

It could be seen that all the pixels of the image get the left or right neighborhood pixel values in X axis by X-shift transformation.

Symmetrically, we can define Y-Shift transformation.

Definition 2. $U(y\pm)$: Y-Shift transformation based on a FRQI image with size of $2^n \times 2^n$ is defined in Eq. (7):

$$U(y\pm) \, |I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{YX}\rangle \, |(Y \pm 1) \bmod 2^n\rangle \, |X\rangle. \tag{7}$$

It is also known that:

$$U(y\pm) \, |I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |C_{Y'X}\rangle \, |Y\rangle \, |X\rangle, \tag{8}$$

where $Y' = (Y \mp 1) \bmod 2^n$. It is obvious that all the pixels of the image can get the updown neighborhood pixel values in Y axis by Y-shift transformation.

Figure 3 gives the example of the transformation $U(x+)$ for an image with size of 8×8. It is shown that all pixels shift one unit rightwards and the pixels in the column of the right edge move to the left edge after the operation.

Then we will analyze the unitarity and computational complexity of $U(x+)$ as the example of all these shift transformations in the next subsections.

4.1.2 *Unitarity*

Whether the shift transformations could be done in quantum mechanics is dependent on the unitarity. Then we will give the proof of the unitarity of the shift transformation $U(x+)$. Table 1 depicts the different operations and transition matrices of $U(x+)$ for the different widths in X axis of the images.

From Table 1, we can conclude that when we utilize n qubits to store the information of X axis of the pixels, $U(x+)$ will make a loop shift operation for the n-length qubit sequence. Its transition matrix would be shown as Eq. (9):

Table 1 The different operations and transition matrices of $U(x+)$ for the different widths in X axis of different images

Width	Storage	Operation	Transition matrix
2 pixels	1 qubit	$\|0\rangle \to \|1\rangle$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
		$\|1\rangle \to \|0\rangle$	
4 pixels	2 qubits	$\|00\rangle \to \|01\rangle$	$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
		$\|01\rangle \to \|10\rangle$	
		$\|10\rangle \to \|11\rangle$	
		$\|11\rangle \to \|00\rangle$	
8 pixels	3 qubits	$\|000\rangle \to \|001\rangle$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
		$\|001\rangle \to \|010\rangle$	
		$\|010\rangle \to \|011\rangle$	
		$\|011\rangle \to \|100\rangle$	
		$\|100\rangle \to \|101\rangle$	
		$\|101\rangle \to \|110\rangle$	
		$\|110\rangle \to \|111\rangle$	
		$\|111\rangle \to \|000\rangle$	

$$\begin{bmatrix} \mathbf{0}^T & 1 \\ I_{2^n-1} & \mathbf{0} \end{bmatrix}, \tag{9}$$

where I_{2^n-1} is an identity matrix with a size of $(2^n-1)\times(2^n-1)$, $\mathbf{0}$ is a (2^n-1)-dimensional vector and $\mathbf{0}^T$ is the transposed vector of $\mathbf{0}$.

From Eq. (9), we find that $U_{(x+)}U_{(x+)}^T = I$. Therefore, the transition matrix is unitary [19].

4.1.3 *Computational complexity*

Next, we will discuss the computational complexity of the transformation $U(x+)$ for an n-length qubit sequence.

It is reported that every arbitrary multi-qubit operation can be decomposed into some single-qubit gates, 2-qubit gates and 3-qubit gates [20]. In general, the number of these simple quantum gates represents the estimate of the computational complexity of the quantum operation. Therefore, we will focus on how to decompose the unitary transformation $U(x+)$.

For an image with a size of $2^n \times 2^n$, the X axis will be displayed as Eq. (10).

$$|x\rangle = |x_{n-1}x_{n-2}...x_1x_0\rangle\,. \tag{10}$$

Therefore, the transformation $U(x+)$ will be decomposed as Eq. (11).

$$\forall x_i, U_{x+}(x_i) = \overline{x_i} \;\text{ iff }\; x_{i-1}x_{i-2}...x_1x_0 = 11...11. \tag{11}$$

It means every qubit in the qubit sequence will be controlled by all the qubits below, and this is just the function of K Control-Not gate ($k-Cnot$). Figure 4 gives the detailed quantum circuit of the transformation $U(x+)$ for an n-length qubit sequence.

From Figure 4, when we make the operation $U(x+)$ for an n-length qubit sequence, the actual operation can be decomposed into n steps, and in the kth step sub-operation to deal with the kth qubit. In the quantum circuit, x_0 needs a single-qubit reversal gate because it will be overturned without any control during shifting. While x_k will be operated by a $k-Cnot$ gate, which is a $k+1$-qubit gate. Xu et al. [21] have studied that a $k-Cnot$ gate can be constructed by a set of $2-Cnot$ gates (which is a kind of the common 3-qubit gate, called Toffoli gate). It is remarkable that Xu's method need $k-2$ assistant qubits to construct the quantum circuit of a $k-Cnot$ gate. Fortunately, when we make the transformation $U(x+)$, all the qubits, which are used to represent the position of Y axis can be utilized to be the assistant

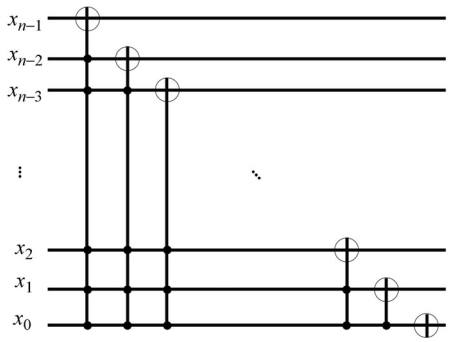

Figure 4 The detailed quantum circuit of the decomposition of $U(x+)$ for an n-length qubit sequence.

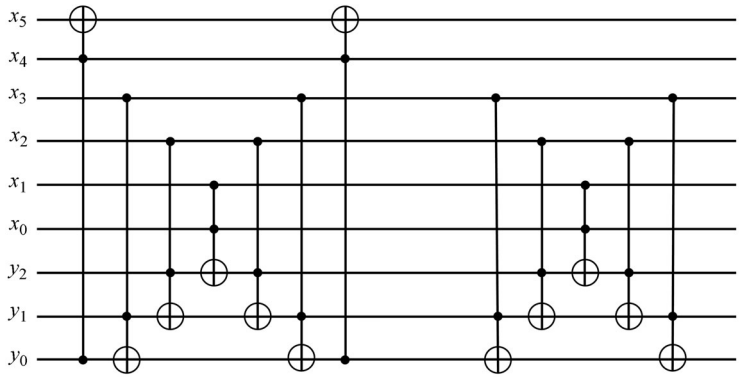

Figure 5 The quantum circuit which utilizes Xu's method [21] to construct a $5-Cnot$ gate.

qubits (The states of these assistant qubits do not change before and after the whole circuit, therefore the qubits of Y-axis can assist the operation $U(x+)$). Hence, it means we can utilize some $2-Cnot$ gates to construct a $k-Cnot$ gate without any extra qubits. Figure 5 gives the quantum circuit of Xu's method to construct a $5-Cnot$ gate.

In Figure 5, the left circuit controls the change of the x_5 qubit. It is obvious that when all the qubits below are equal to $|1\rangle$, the x_5 qubit would overturn. While the right circuit mainly recovers the state of all the assistant qubits. For a $k-Cnot$ gate, the number of $2-Cnot$ gates, which are used to construct the left circuit is $(k-2)\times 2+1$, while the number of $2-Cnot$ gates for the right circuit is $(k-3)\times 2+1$. Therefore, in order to construct a $k-Cnot$ gate, a set of $2-Cnot$ gates with a number of $4k-8$ are needed, as well as some assistant Y-axis qubits with number of $k-3$.

Through the analysis of the quantum circuit in Figure 4, it is known that if we only use single-qubit reversal gate, 2-qubit gate, and 3-qubit Toffoli gate to construct the whole circuit of the transformation $U(x+)$, we need one single-qubit reversal gate, one 2-qubit gate, and some 3-qubit Toffoli gates. And the total number of 3-qubit Toffoli gates is as follows:

$$1+\sum_{i=3}^{n-1} number(i-Cnot) = 1+\sum_{i=3}^{n-1}(4i-8) = n^2-5n+7. \tag{12}$$

Although we do not ensure that Xu's method can find the best way to decompose the unitary transformation $U(x+)$ into the least simple gates, it is demonstrated that the total computational complexity of the transformation $U(x+)$ for an n-length qubit sequence is no more than $O(n^2)$.

4.2 Mask computation

In this subsection, we will utilize the two aforementioned shift transformations to calculate the intensity gradients of all the pixels in an image simultaneously according to the Sobel masks and the neighborhood window.

Table 2 Computation prepared algorithm

Input: Original image: I_{xy}, which is as Eq. (1). $\|I\rangle = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{YX}\rangle\|YX\rangle$
Step 1: U_{y+}, shift I_{xy} one-unit downwards, then $I_{xy-1} = U_{y+}(I_{xy}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{Y-1X}\rangle\|YX\rangle$
Step 2: U_{x+}, shift I_{xy-1} one-unit rightwards, then $I_{x-1y-1} = U_{x+}(I_{xy-1}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{Y-1X-1}\rangle\|YX\rangle$
Step 3: U_{y-}, shift I_{x-1y-1} one-unit upwards, then $I_{x-1y} = U_{y-}(I_{x-1y-1}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{YX-1}\rangle\|YX\rangle$
Step 4: U_{y-}, shift I_{x-1y} one-unit upwards, then $I_{x-1y+1} = U_{y-}(I_{x-1y}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{Y+1X-1}\rangle\|YX\rangle$
Step 5: U_{x-}, shift I_{x-1y+1} one-unit leftwards, then $I_{xy+1} = U_{x-}(I_{x-1y+1}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{Y+1X}\rangle\|YX\rangle$
Step 6: U_{x-}, shift I_{xy+1} one-unit leftwards, then $I_{x+1y+1} = U_{x-}(I_{xy+1}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{Y+1X+1}\rangle\|YX\rangle$
Step 7: U_{y+}, shift I_{x+1y+1} one-unit upwards, then $I_{x+1y} = U_{y+}(I_{x+1y+1}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{YX+1}\rangle\|YX\rangle$
Step 8: U_{y+}, shift I_{x+1y} one-unit upwards, then $I_{x+1y-1} = U_{y+}(I_{x+1y}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{Y-1X+1}\rangle\|YX\rangle$
Step 9: $U_{y-}U_{x+}$, shift I_{x+1y-1} to the original position, $I_{xy} = U_{y-}U_{x+}(I_{x+1y-1}) = \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\|C_{YX}\rangle\|YX\rangle$

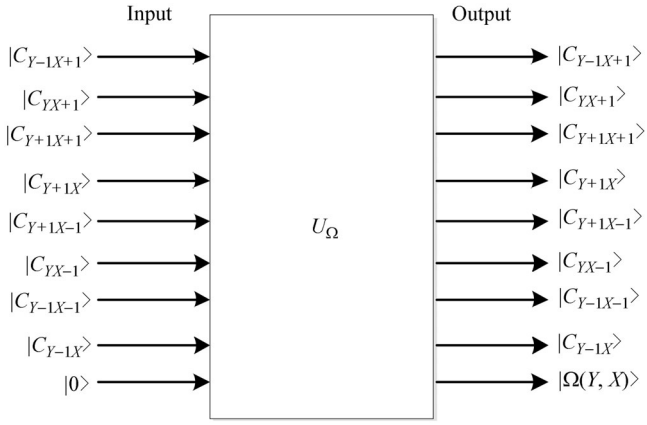

Figure 6 The quantum black box U_Ω.

Figure 2(b) and (c) show the Sobel masks. In order to get the intensity gradient of every pixel, it needs to utilize the masks to calculate the approximate gradient according to its 3×3 neighborhood window. The color information of the neighborhood window of every pixel can be obtained simultaneously by the shift transformations in a certain order. Then the detailed computation prepared algorithm that can do this task will be given as Table 2.

From computation prepared algorithm, it could be seen that after Step 1 to Step 8, the shift operations are utilized to get the color information of the neighborhood pixels of all the pixels simultaneously. There are totally 6 Y-Shift operations and 4 X-shift operations. Therefore, the computational complexity of computation prepared algorithm is still $O(n^2)$.

In order to calculate the approximate gradient of image intensity of all the pixels, a quantum black box U_Ω is designed, which utilizes the color qubits of the neighborhood pixels in computation prepared algorithm. Figure 6 shows the function of the quantum black box U_Ω.

In this quantum black box, it works as Eq. (13).

$$U_\Omega(\hat{C} \otimes |0\rangle) = \hat{C} \otimes |\Omega(Y,X)\rangle\,, \tag{13}$$

where $\hat{C}$ and $|\Omega(Y,X)\rangle$ are presented in Eqs. (14) and (15), respectively:

$$\hat{C} = |C_{Y-1X+1}\rangle\otimes|C_{YX+1}\rangle\otimes|C_{Y+1X+1}\rangle\otimes|C_{Y+1X}\rangle\otimes|C_{Y+1X-1}\rangle\otimes|C_{YX-1}\rangle\otimes|C_{Y-1X-1}\rangle\otimes|C_{Y-1X}\rangle\,; \tag{14}$$

$$|\Omega(Y,X)\rangle = \begin{cases} |0\rangle, \ (G_x(Y,X)^2 + G_y(Y,X)^2)^{1/2} < T, \\ |1\rangle, \ (G_x(Y,X)^2 + G_y(Y,X)^2)^{1/2} \geqslant T. \end{cases} \tag{15}$$

In Eq. (15), $G_x(i,j)$ and $G_y(i,j)$ are the approximations of the derivatives of X-axis and Y-axis, respectively as Eqs. (16) and (17).

$$G_x(Y,X) = (C_{Y-1X+1} + 2C_{YX+1} + C_{Y+1X+1}) - (C_{Y-1X-1} + 2C_{YX-1} + C_{Y+1X-1}), \tag{16}$$

$$G_y(Y,X) = (C_{Y+1X-1} + 2C_{Y+1X} + C_{Y+1X+1}) - (C_{Y-1X-1} + 2C_{Y-1X} + C_{Y-1X+1}). \tag{17}$$

Through the operation of the quantum black box U_Ω, we have finished the computation of the Sobel gradients of all the pixels in the whole image and stored the values into a color qubit $|\Omega(Y,X)\rangle$, which will be entangled with the position qubit sequence to form a new result image. Similar to the classical Sobel algorithm, in the new result image, the color qubit of all the pixels in the edge is $|1\rangle$, which represents white. On the contrary, the color qubit of other pixels will be $|0\rangle$, which represents black.

4.3 QSobel edge extraction

Through the last two subsections, all the detailed work for our novel quantum edge extraction algorithms, QSobel, has been introduced. In this subsection, the whole workflow of QSobel, as well as the integrated quantum circuit will be discussed.

4.3.1 *Workflow of QSobel*

In order to cope with the real-time problem of the classical edge extraction algorithms, a novel quantum edge extraction algorithm, QSobel, is proposed in this paper. The workflow of QSobel for a digital image with a size of $2^n \times 2^n$ is shown here as an example.

Step 1: Firstly, the digital image should be quantized into a FRQI quantum image as Eq. (1). In order to store this quantum image, we need a $2n+1$-qubit quantum register. Meanwhile, 9 extra qubits for the next operations are needed to store the color information of the neighborhood pixels of all the pixels and the result of QSobel.

Step 2: Perform computation prepared algorithm. In the algorithms, after Steps 1–8, we can get the relative color qubits of the neighborhood pixels of the whole image and store them into an extra qubit. The main operations in this step are X-Shift and Y-Shift transformations.

Step 3: When we have stored all the eight color qubits of the 3×3 neighborhood window, the quantum black box U_Ω is utilized to calculate the Sobel gradients of all the pixels simultaneously and store the result into another extra qubit $|\Omega(Y,X)\rangle$. This qubit can construct a new quantum image by entangling with the position qubit sequence.

Similar with the theory of Sobel edge extraction algorithms, in the resulting quantum image, the color qubit of the pixels in the edges will be $|1\rangle$, which represents white. On the contrary, the color qubit of the pixels not in the edges is $|0\rangle$, which means black. Therefore, QSobel has distinguished the different classes of the pixels and extracted the edges of the quantum image.

4.3.2 *Quantum circuit of QSobel*

The whole quantum circuit of QSobel can be partitioned into 3 stages as shown in Figure 7.

Stage 1: The work of Step 1 of QSobel is done in this stage. We need to prepare the quantum image in the model of FRQI and enough assistant qubits. From Figure 7, the whole circuit needs $2n+10$ qubits totally, including $2n+1$ qubits for storing the quantum image and 9 extra qubits, respectively. In the initial, we construct the FRQI quantum image with the method of Ref. [11]. Meanwhile, all the assistant qubits are initialized to be $|0\rangle$. The whole state of the quantum circuit after Stage 1 is as follows:

$$|\Psi\rangle = \sum (\bigotimes_{k=1}^{9} |0\rangle_k) |C_{YX}\rangle |Y\rangle |X\rangle . \tag{18}$$

Stage 2: This section of circuit is relative to the work of Step 2 of QSobel. All the color qubits of the neighborhood pixels of the image will be obtained and stored. For example, after the first operation U_{y+},

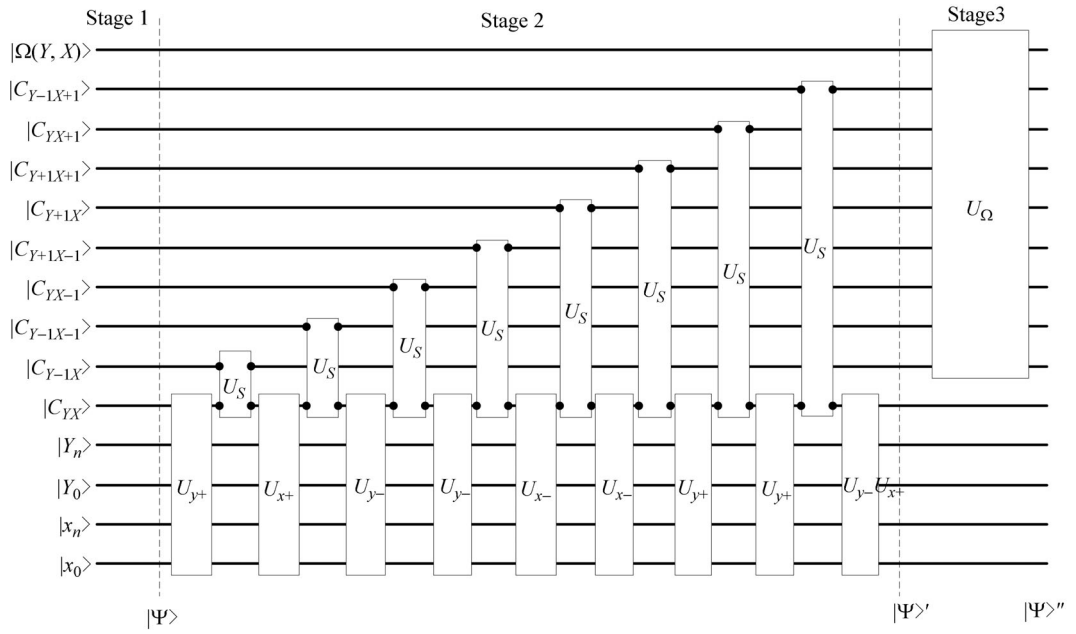

Figure 7 The whole quantum circuit of QSobel.

the image is changed to I_{xy-1} and the color qubit $|C_{YX}\rangle$ is changed to $|C_{Y-1X}\rangle$. Utilizing the quantum operation U_S as Eq. (19), the color qubit has been stored. In order to obtain all the color qubits of the 3×3 neighborhood window, all the 9 steps of computation prepared algorithm would be done in this stage. After this stage of the quantum circuit, the whole state of the quantum circuit is as Eq. (20).

$$U_S(|C\rangle\,|0\rangle) = |C\rangle\,|C\rangle\,, \tag{19}$$

$$\begin{aligned}|\Psi\rangle' = \sum (&|0\rangle\,|C_{Y-1X+1}\rangle\,|C_{YX+1}\rangle\,|C_{Y+1X+1}\rangle\,|C_{Y+1X}\rangle\\ &|C_{Y+1X-1}\rangle\,|C_{YX-1}\rangle\,|C_{Y-1X-1}\rangle\,|C_{Y-1X}\rangle\,|C_{YX}\rangle\,|Y\rangle\,|X\rangle)\,.\end{aligned} \tag{20}$$

Stage 3: In this stage of the quantum circuit, the quantum black box will be done to finish the work of QSobel. In $|\Psi\rangle'$, all the color qubits of the 3×3 neighborhood window have been prepared for the quantum black box U_Ω to calculate the Sobel gradients of all the pixels and store the result in another assistant qubit $|\Omega(Y,X)\rangle$. After the operation of U_Ω, the whole state of the quantum circuit is as Eq. (21).

$$\begin{aligned}|\Psi\rangle'' = \sum (&|\Omega(Y,X)\rangle\,|C_{Y-1X+1}\rangle\,|C_{YX+1}\rangle\,|C_{Y+1X+1}\rangle\\ &|C_{Y+1X}\rangle\,|C_{Y+1X-1}\rangle\,|C_{YX-1}\rangle\,|C_{Y-1X-1}\rangle\,|C_{Y-1X}\rangle\,|C_{YX}\rangle\,|Y\rangle\,|X\rangle)\,.\end{aligned} \tag{21}$$

Therefore, in $|\Psi\rangle''$, $|\Omega(Y,X)\rangle$ and the position qubit sequence can construct a new result quantum image as $|\Psi\rangle'''$ shown in Eq. (22). In this new image, the color qubit of the pixels in the edges will be $|1\rangle$, while the color qubit of the pixels not in the edges is $|0\rangle$. Therefore, QSobel has extracted the edges of the quantum image.

$$|\Psi\rangle''' = \sum |\Omega(Y,X)\rangle\,|Y\rangle\,|X\rangle\,. \tag{22}$$

5 Tests and performance analyses

In this section, the differences in the performance between QSobel and other edge extraction algorithms are compared.

First of all, in order to analyze the computational complexity of QSobel, we will discuss according to the quantum circuit in the last section. Here consider a digital image with a size of $2^n \times 2^n$ as an example.

In Stage 1, the quantum circuit will make the construction of the FRQI quantum image $|\Psi\rangle$. From the

Table 3 The performance comparisons between QSobel and other edge extraction algorithms for a digital image with a size of $2^n \times 2^n$

Algorithm	Storage	Quantum image model	Computational complexity of quantum image construction	Computational complexity of edge extraction
Sobel	$2^{2n} * 8bit$	-	-	$O(2^{2n})$
Canny	$2^{2n} * 8bit$	-	-	$O(2^{2n})$
Tseng	$2^{2n} qubit$	Qubit Lattice	$O(2^{2n})$	$O(2^{2n})$
Fu	$2^{2n} qubit$	Qubit Lattice	$O(2^{2n})$	$O(2^{2n})$
QSobel	$(2n+10) qubit$	FRQI	$O(2^{4n})$	$O(n^2)$

Sobel and Canny are classical algorithms. Therefore some items are null.

Table 4 Thresholds of Sobel and QSobel when extracting edges for the test images

Test image	Thresholds of Sobel and QSobel
Rice	0.071
Peppers	0.087
Lena	0.071
Camera	0.143

introduction of Ref. [11], it is known that the computational complexity of this stage is $O(2^{4n})$ because some operations will be done for every pixel one-by-one.

In section 4.1.3, we know that every shift transformation will cost the time of $O(n^2)$. Meanwhile, from Figure 7, there are totally 10 shift transformations and 8 quantum operations of U_S in Stage 2. Because U_S is an operation with a constant complexity of $O(1)$, it is obvious that the computational complexity of Stage 2 is $O(n^2)$.

The mainly work of Stage 3 is the quantum black box U_Ω, which is a 9-qubit operation. Meanwhile, we can see that the computational cost of Stage 3 is constant, and it does not change as does the size of the image. Therefore, the computational complexity of Stage 3 is $O(1)$.

Note that the time cost of quantum image construction of FRQI is long, however, it is not considered as a part of the quantum image processing. Therefore, if the quantum image has been constructed, our proposed algorithm QSobel can extract the edges for a FRQI quantum image in computational complexity of $O(n^2)$. Hence, we can give the conclusion that the novel quantum edge extraction algorithm, QSobel, can reach an exponential speed than all the classical edge extraction algorithms and all the existing quantum edge extraction algorithms.

Table 3 demonstrates the performance comparisons between QSobel and other edge extraction algorithms including two famous classical algorithms (Sobel [14] and Canny [17]) and two existing quantum edge extraction algorithms (Tseng [12] and Fu [13]) for a digital image with a size of $2^n \times 2^n$. Because QSobel is based on the quantum image model FRQI, it shows better performance. In the aspect of storage, QSobel needs $2n + 10$ qubits, which has an exponential decrease than others. Meanwhile, we have analyzed that QSobel can extract the edges in complexity of $O(n^2)$ and it is the fastest algorithm of all.

In order to test our new algorithm, we used a classical computer to simulate QSobel to do the edge extraction. All the simulations were coded in C language and ran in the processor of Intel Core2 Quad CPU Q6600. Four common test images such as Rice, Peppers, Lena, and Camera were chosen.

Through setting some certain thresholds as Table 4, all the edges of these test images will be extracted as Figure 8. In the near future, when quantum computer can be used for the actual applications, QSobel will show its unbelievable performance.

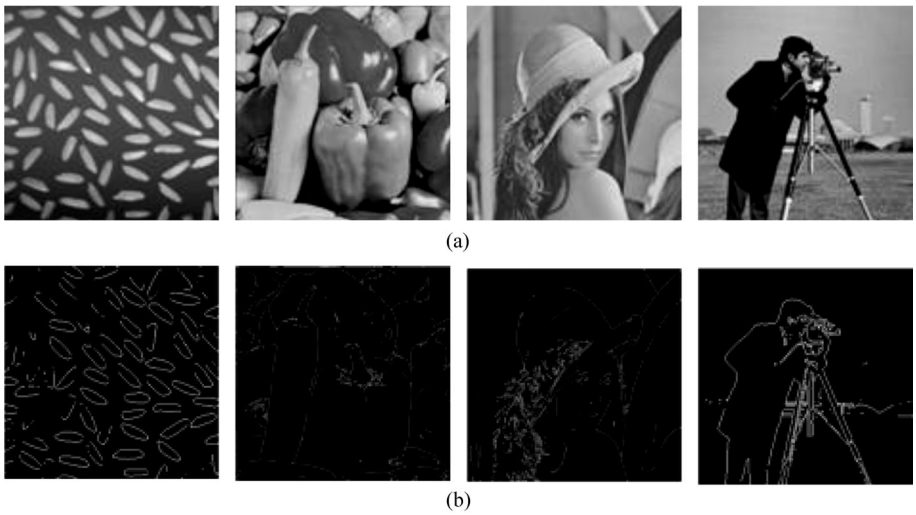

Figure 8 (a) Four common and original test images. (b) The result images of QSobel edge extraction by the thresholds in Table 4.

6 Conclusion

With the sharp increase of the image data in the actual applications, most of the classical digital image processing algorithms would take impractically longer, including the edge extraction algorithms on which this paper mainly focuses. In this paper, QSobel, a novel quantum edge extraction algorithm is designed, which combines the quantum image model of FRQI and Sobel edge extraction algorithm. FRQI utilizes the superposition state of qubit sequence to store all the pixels of the image, therefore QSobel, which is based on FRQI can calculate the Sobel graidents of all the pixels simultaneously. Through designing the quantum circuit of QSobel, it is reported that QSobel can extract edges in the computational complexity of $O(n^2)$ for a FRQI quantum image with a size of $2^n \times 2^n$. Compared with all the classical edge extraction algorithms and the existing quantum edge extraction algorithms, QSobel can utilize the quantum computation to reach an exponential speedup and the same result with the classical Sobel algorithm. Hence, as far as I know, QSobel will be the fastest edge extraction algorithms of all by now.

Because of the drawbacks of Sobel, QSobel may be deficient in robustness for noise and as a result QSobel may have serious errors when processing some complex image. However, we can utilize the approach of QSobel to design other quantum edge extraction algorithms, such as quantum Canny [17], quantum Susan [22]. Therefore, in that sense, QSobel can give us a significant approach to design quantum counterparts for the classical image processing algorithms.

In this paper, this drastic improvement of QSobel is liable on the peculiar properties of the quantum image model FRQI, which utilizes the superposition state of qubit sequence to store all the pixels of an image. Therefore, we can process the information of all the pixels simultaneously. However, there are still some drawbacks in this quantum image model. Firstly, FQRI image stores all the color information in the probability amplitude of a qubit [11], so we could not utilize the quantum measurement to obtain the exact and classical color information to retrieve the original classical digital image. It means this quantum image model is not fit for image retrieving.. Recently some new quantum image models are published [23,24], which are worth being focused on in the near future. Secondly, the quantum image processing based on FRQI needs to make all the operations unitary, that is, the procedure must be invertible. Therefore, the image convolution operation [25] and the arbitrary angle rotation operations [26] (Since the coordinate discretization and gray-scale quantization in the digital image processing, information loss will occur during the arbitrary angle rotation operations) cannot be done in FRQI images. In conclusion, our future researches will emphasis on designing a novel quantum image representation to improve FRQI for quantum image processing.

Acknowledgements

This work was supported in part by the National High-tech R&D Program of China (863 Program) (Grant Nos. 2012AA01A301 and 2012AA010901). And it was partially supported by the National Natural Science Foundation of China (Grant Nos. 61170261 and 61103082). Moreover, it is a part of the Innovation Fund Sponsor Project of Excellent Postgraduate Student (Grant Nos. B120601 and CX2012A002). We thank the anonymous reviewers for their helpful feedback.

References

1 Nielsen M A, Chuang I L. Quantum Computation and Quantum Information. Cambridge: Cambridge Univ Press, 2000
2 Feynman R. Simulating physics with computers. Int J Theor Phys, 1982, 21: 467–488
3 Shor P W. Algorithms for quantum computation: Discrete logarithms and factoring. In: Proceedings of 35th Annual Symposium on Foundations of Computer Science, Los Almitos, USA, 1994. 124–134
4 Grover L. A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual Symposium on the Theory of Computing, Philadelphia, USA, 1996. 212–219
5 Rafael C G, Richard E W, Steven L E. Digital Image Processing, 4th ed. Beijing: House of Electronics Industry Press, 2002
6 Venegas-Andraca S E, Ball J L. Storing images in engtangled quantum systems. arXiv:quant-ph/0402085, 2003
7 Venegas-Andraca S E, Bose S. Storing, processing and retrieving an image using quantum mechanics. In: Proceedings of the SPIE Conference on Quantum Information and Computation, 2003. 137–147
8 Venegas-Andraca S E, Ball J L, Burnett K, et al. Processing images in entangled quantum systems. Quantum Inf Process, 2010, 9: 1–11
9 Venegas-Andraca S E, Bose S. Quantum computation and image processing: New trends in artificial intelligence. In: Proceedings of the International Conference on Artificial Intelligence, 2003. 1563–1564
10 Latorre J I. Image compression and entanglement. arXiv:quant-ph/0510031, 2005
11 Le P Q, Dong F, Hirota K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. Quantum Inf Process, 2011, 10: 63–84
12 Tseng C, Hwang T. Quantum digital image processing algorithms. In: Proceedings of the 16th IPPR Conference on Computer Vision, Graphics and Image Processing, 2003. 827–834
13 Fu X, Ding M, Sun Y, et al. A new quantum edge detection algorithm for medical images. In: Proceedings of Medical Imaging, Parallel Processing of Images and Optimization Techniques. SPIE Vol.7497, 2009
14 Sobel L. Camera Models and Machine Perception. Stanford: Stanford Univ Press, 1970
15 Prewitt J. Object Enhancement and Extraction. New York: Picture Process and Psychopictoric Press, 1970. 75–149
16 Kirsch R A. Computer determination of the constituent structure of biological images. Comput Biol Med, 1971, 18: 113–125
17 Canny J. A computational approach to edge detection. IEEE TPAMI, 1986, 8: 679–697
18 Niya J M, Aghagolzadeh A. Edge detection using directional wavelet transforms. In: Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference, Ajaccio, 2004. 1: 281–284
19 Horn R A, Johnson C R. Matrix Analysis. Cambridge: Cambridge Univ Press, 1985
20 Lloyd S. Almost any quantum logic gate is universal. Phys Rev Lett, 1995, 75: 346–349
21 Xu X, Xiao F. Application of dichotomy in decomposition of multi-line quantum logic gate (in Chinese). J Southeast Univ, 2010, 40: 928–931
22 Qu Z, Wang P, Gao Y. Randomized SUSAN edge detector. Opt Eng Lett, 2011, 11: 1–4
23 Zhang Y, Lu K, Gao Y, et al. NEQR: A novel enhanced quantum representation of digital images. Quantum Inf Process, 2013, 12: 2833–2860
24 Zhang Y, Lu K, Gao Y, et al. A novel quantum representation for log-polar images. Quantum Inf Process, 2013, 12: 3103–3126
25 Lomont C. Quantum convolution and quantum correlation algorithms are physically impossible. arXiv:quant-ph/0309070, 2003
26 Le P Q, Iliyasu A M, Dong F, et al. Strategies for designing geometric transformations on quantum images. Theor Comput Sci, 2011, 412: 1406–1418