Results in Physics 13 (2019) 102161

Contents lists available at ScienceDirect

Results in Physics

journal homepage: www.elsevier.com/locate/rinp

Enhanced double random phase encryption of quantum images

Shiping Du[a], Daowen Qiu[a,b,c,*], Paulo Mateus[b], Jozef Gruska[c]

[a] School of Data and Computer Science, Sun Yat-sen University, 510006 Guangzhou, China
[b] Instituto de Telecomunicações, Departamento de Matemática, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisbon, Portugal
[c] Faculty of Informatics, Masaryk University, Brno, Czech Republic

ARTICLE INFO

Keywords:
Quantum image
Information diffusion
Enhanced double random phase encryption

2000 MSC:
41A05
41A10
65D05
65D17

ABSTRACT

We improve the double random phase encryption (DRPE) algorithm to make the outcome of DRPE algorithm as uniformly mixed as possible via applying scrambling transform. So, the effectiveness of this method is reduced to prove the credibility of the scrambling transform. In doing so, our contributions include: (1) A transform set **Set** is constructed with some foundation transforms which have been proved to be numerous, to have lower complexity, and to be specified using rules. (2) A scrambling transform *P* constructed based on **Set** verifies the existence of the trend of the scrambling effect, and this trend means there exists an optimal scrambling transform that can be constructed with **Set** to uniformly mix the pixel information. The double-randomness of scrambling transform, the theoretic analysis and experimental results show the enhanced algorithm is effective.

Introduction

Quantum image as a hot topic has been discussed for a long time [1]. As an image, quantum image is also constituted by pixels [2]. So, pixel value and its coordinator determine a specified pixel information in a pixel matrix of a quantum image. Several different quantum states are used to express the state representation of quantum images, such as quantum image representation for log-polar images (QUALPI) [3], entangle state quantum image[4], flexible representation of quantum image (FRQI) [5,6].

The existing image encryption algorithms could be categorized into two kinds. The first category of the quantum image encryption method is scrambling transform [7]. This method encrypts the image information via scrambling the pixel information of the original image to get a meaningless image. This type operations appear in the fields of the classical image encryption, such as [8]. There are many such transforms, such as Fourier transformation [9], the Fresnel transform (FST) [10], the linear canonical transform (LCT) [11], the Gyrator transform (GT) [12] and the Hartley transform (HT) [13]. In addition, so called rotation operation is also an important transform introduced and discussed in [14].

The second category encryption method is more closely related to the particular quantum image state representation since the important information is stored in the pixel of the image [15], e.g., if the pixel of an images are quantumly represented by the amplitude, then, their encryption approach should be done by encrypting their amplitudes (similarly, see [15]), and if the pixel of the quantum image is denoted by the phase, then the encryption should modulate all the original phases to other values. Correspondingly, the pixel information of the original image is changed [16,17] (operation on the phase may trouble us on the phase estimation [18]). The early phase encryptions of the quantum image were inspired by Javadi who introduced the DRPE to implement the image encryption in the field of the quantum optics [19]. These operations actually are the main methods of image encryption processing in optic fields, the related documents also see [20,21]. This thinking has been used to develop a new algorithm to encrypt the quantum image through phase transform and Fourier transform [19,22]. Therefore, it is clear that, for different state representation of the quantum images, we need to devise the suitable method to realise the image encryption.

Three problems: the first, the original DRPE algorithm could be attacked under some conditions [23]. The second, adopting Fourier transform to improve DRPE encryption effect remains vulnerability [14,24], because inverse Fourier transform is a decryption key. The third, some paper points out that their encryption could obtain perfect scrambling effect with rotation operator is not exactly right [14]. We propose another enhanced DRPE (eDRPE) algorithm via applying a scrambling transform on the outcome of DRPE algorithm. Obviously, the new algorithm combines with two kinds of the encryption methods mentioned in the two categories above. That is, eDRPE method includes two steps. The first step is applying DRPE algorithm on the quantum image, and then we apply the scrambling transform on the outcome of

[*] Corresponding author at: School of Data and Computer Science, Sun Yat-sen University, 510006 Guangzhou, China.
E-mail address: issqdw@mail.sysu.edu.cn (D. Qiu).

https://doi.org/10.1016/j.rinp.2019.102161
Received 11 January 2019; Received in revised form 26 February 2019; Accepted 27 February 2019
Available online 11 March 2019

the first step. The critical point of implementing this method is to devise a strong scrambling transform operator to uniformly scramble the quantum pixel matrix. To this end, we will construct an operator set, and employ some combinatorial foundation transform to execute this scrambling transform. Contrasting with other methods, the advantage of eDRPE could provide a key with good operability, large enough space and flexible operation combination. These features will be described in Section "Advantages discussion".

This paper is organised as follows. In Section "Model of enhanced algorithm", we describe the model of eDRPE. In Section "Properties of the sub-operators of scrambling operators", a set which constitutes with some foundation transforms is constructed, and the main properties of this set are investigated. All these properties of the foundation transforms decide the characteristics of the scrambling transform. In Section "Operator construction and experimental results", an experiment using the transform constructed with the foundation transforms in Section "Properties of the sub-operators of scrambling operators" to verify the effectiveness of eDRPE is carried out. Section "Advantages discussion" is the comparison analysis of the present method and the previous. Conclusions are in the last section.

Model of enhanced algorithm

State representation of quantum images

In this paper, the state representation of the quantum image is specified as follows (note, as proposed in [24]),

$$|I(\theta_j)\rangle = \frac{1}{2^n}\sum_{j=0}^{2^{2n}-1} (|0\rangle + e^{i\theta_j}|1\rangle) \otimes |j\rangle, \tag{1}$$

where $\theta_j \in (0, \pi/2), j = 0, 1, \ldots, 2^{2n} - 1$. The relative phase $\theta_j \in \{\theta_1, \theta_2, \ldots, \theta_{2^{2n}-1}\}$ in $|0\rangle + e^{i\theta_j}|1\rangle$ encodes the pixel information, and $|j\rangle$, for $j = 0, 1, \ldots, 2^{2n} - 1$, which is a 2^{2n} dimensional vector, represents the position of a pixel.

Take the quantum image with $(3 + 3 + color)$ qubits as an example. The corresponding quantum circuit for the image can is presented in Fig. 1 (a). Fig. 1(b) is a corresponding pixel matrix.

The first 3 qubits of $(3 + 3 + color)$, $|y_2\rangle$, $|y_1\rangle$, and $|y_0\rangle$, are used to represent the y coordinate of the pixel of the quantum image. Next 3 qubits, $|x_2\rangle$, $|x_1\rangle$, and $|x_0\rangle$, are used to represent the x coordinate of the pixel of the quantum image. This is because each pixel in quantum image has its own specified coordinate. To see this clearly, we can see Fig. 1(b), which is a pixel matrix with size $8 * 8$. In this two dimensional plane, each number c_v represents a pixel of the quantum image (let $i, j \in \{1, 2, \ldots, 8\}$, $v = i * j$. So, $v \in \{1, 2, \ldots, 64\}$). Therefore, the coordinate (x, y) of each pixel is a (i, j) pair.

Each pixel v with coordinates $v = (i, j)$ has its special color, which will be labeled by c_v. Actually, it is $c_{i,j}$. Since one line of quantum circuit could represent a superposition state of $|0\rangle$and $|1\rangle$, to represent a quantum image with size 8*8, '3' qubits and '3' qubits are needed to distinguish x and y coordinate, respectively.

The essence of using quantum circuit transformation to implement the information hiding is coordinate transformation. That's to say, color of the pixel point is transformed followed by the change of the coordinate. However, the color itself is not changed, and only the coordinate are changed. So, in our requirement, we do not care the color change, but we are only concerned about the coordinate changes. This is why 'color' qubits to represent the color of the quantum image is enough here.

Geometry transform [25] of the quantum image is one important branch of image information processing. Image transform encrypts the image via exchanging the pixels in the pixel matrix, e.g., in Fig. 1, we can swap c_{62} and c_{48}, swap c_{60} and c_{32}, or swap c_{61} and c_{58}, and so on. The transform of the quantum image can be described with Fig. 2, where the assumed unitary operator U is an unitary geometry transformation operator.

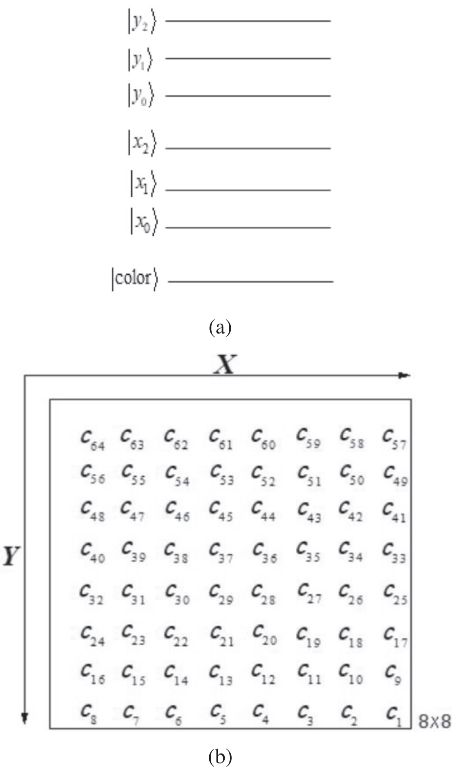

$|y_2\rangle$
$|y_1\rangle$
$|y_0\rangle$
$|x_2\rangle$
$|x_1\rangle$
$|x_0\rangle$
$|color\rangle$
(a)
X
Y
c_{64} c_{63} c_{62} c_{61} c_{60} c_{59} c_{58} c_{57}
c_{56} c_{55} c_{54} c_{53} c_{52} c_{51} c_{50} c_{49}
c_{48} c_{47} c_{46} c_{45} c_{44} c_{43} c_{42} c_{41}
c_{40} c_{39} c_{38} c_{37} c_{36} c_{35} c_{34} c_{33}
c_{32} c_{31} c_{30} c_{29} c_{28} c_{27} c_{26} c_{25}
c_{24} c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17}
c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9
c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1
8x8
(b)

Fig. 1. (a) is an example of a $2^3 \times 2^3$ quantum image, where x and y are horizontal and vertical coordinate of the pixel point of the quantum image. (b) is an example of pixel information matrix of the quantum image with size $2^3 \times 2^3$. $c_i (i \in [1, 64])$ denotes the pixel at corresponding position. For a 8×8 matrix, it has 64 pixel point in total. We encode them as $c_1, c_2, \ldots, c_{64}$ sequentially.

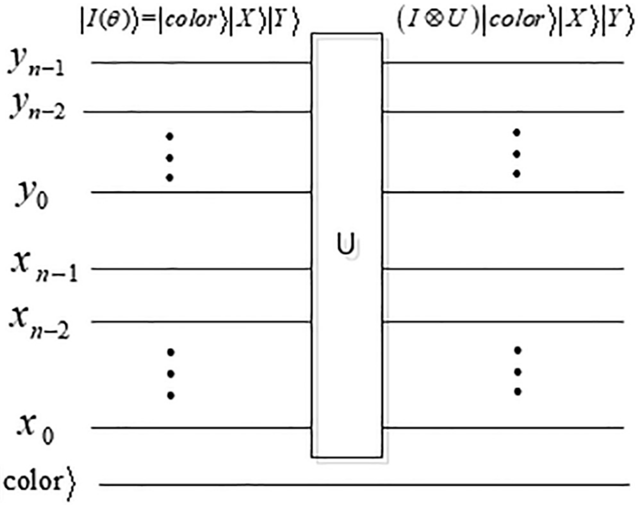

$|I(\theta)\rangle = |color\rangle|X\rangle|Y\rangle$
$(I \otimes U)|color\rangle|X\rangle|Y\rangle$
y_{n-1}
y_{n-2}
y_0
x_{n-1}
x_{n-2}
x_0
$|color\rangle$
U

Fig. 2. Geometry operation of the quantum image. $|I(\theta)\rangle$ is the original image, $U|I(\theta)\rangle$ is the transform result though unitary operation U.

Enhanced double random phase encoding

Our enhanced algorithm adds a transform on the outcome of DRPE algorithm [19] to scramble the information for overcoming the vulnerability mentioned in [23]. Assume that the operator to transform the quantum image is P, then the steps of eDRPE algorithm can be described as follows.

Step 1. A pply the first key $U(\varphi_j)$ with phases φ_j, $(j = \{1, 2, \ldots, 2^{2n}\})$, and let the result be $|\psi_1\rangle$, then we have

$$|\psi_1\rangle = (U(\varphi_j) \otimes I)\frac{1}{2^n}\sum_{j=1}^{2^{2n}} (|0\rangle + e^{i\theta_j}) \otimes |j\rangle \tag{2}$$

$$= \frac{1}{2^n}\sum_{j=1}^{2^{2n}} (|0\rangle + e^{i(\theta_j+\varphi_j)}|1\rangle) \otimes |j\rangle. \tag{3}$$

$U(\varphi_j)$ is defined by Eq. (2).

Step 2. Execute the quantum Fourier transform(QFT) on $|\psi_1\rangle$ to get $|\psi_2\rangle$, which can be represented as

$$|\psi_2\rangle = (I \otimes QFT)(|\psi_1\rangle)$$

where the operation of QFT is defined as

$$QFT: |j\rangle \rightarrow \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{i2\pi jk/M}|k\rangle. \tag{4}$$

QFT is already well known in quantum information processing. Quantum analogue of the classical Fourier transformation is widely used in the classical information and especially in image processing. The quantum circuit for quantum Fourier transformation has been describe by Yao in [26].

Step 3. Apply the second key $U(\theta_j')$ with phases θ_j', $(j = \{1, 2, \ldots, 2^{2n}\})$.

$$|\psi_3\rangle = (U(\theta_j') \otimes I)|\psi_2\rangle \tag{5}$$

$$= \frac{1}{2^n}\sum_{j=1}^{2^{2n}} (|0\rangle + e^{i(\theta_j+\varphi_j+\theta_j')}|1\rangle)\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} e^{i2\pi jk/M}|k\rangle.$$

The transform $U(\theta_j')$ satisfies the Eq. 5.

Step 4. Apply the inverse quantum Fourier transform, and suppose that the result is denoted by $|E\rangle$, we have $|E\rangle = QFT^{-1}(|\psi_3\rangle)$, and

$$|E\rangle = QFT^{-1}[(U(\theta_j') \otimes I)|\psi_2\rangle]. \tag{6}$$

Step 5. Apply the permutation P. That is,

$$P|E\rangle. \tag{7}$$

eDRPE algorithm could be summarized with a diagram as Fig. 3.

Corresponding to the encryption procedure above, the decryption algorithm could be described as the following steps. Three unitary operators $U(\theta_j)^{-1}$, $U(\varphi_j)^{-1}$ and P^{-1} are referred. They are actually the inverse transformation of $U(\theta_j)$, $U(\varphi_j)$ and P.

Step 1. Applying P^{-1}, we get $|E\rangle$.

$$|E\rangle \leftarrow P^{-1}P|E\rangle. \tag{8}$$

Step 2. Decryption using outer-ring Fourier operation with QFT, we get $|\psi_3\rangle$.

$$|\psi_3\rangle \leftarrow QFT|E\rangle. \tag{9}$$

Step 3. Decryption using $U(\theta_j')^{-1}$.

$$|\psi_2\rangle \leftarrow U(\theta_j')^{-1}|\psi_3\rangle. \tag{10}$$

Step 4. Decryption using the inverse Quantum Fourier Transform.

$$|\psi_1\rangle \leftarrow QFT^{-1}(|\psi_2\rangle). \tag{11}$$

Step 5. Decryption using the key $U(\varphi_j)^{-1}$.

$$U(\varphi_j)^{-1}|\psi_1\rangle. \tag{12}$$

The result of the Step 5 is the original image decrypted from Eq. (7).

Main idea

In Section "Enhanced double random phase encoding", we use P to encrypt the quantum image, so the encryption problem now is reduced to consider how to prove the effectiveness of the transform P used in our algorithm. In doing so, we define a functional set **Set** with three categories general operations: point swapping, column or row swapping, and sub-block swapping. We exemplify each element in **Set** with real quantum circuits. Because there are so many pixels in one pixel matrix, so the number of the quantum circuit to implement the different pixel transform also are numerous. That is, these circuits construct a big encryption key space. The operator P is constructed with the combination of these circuits. Then, owing to the sequence and the enough space, P is random which satisfies the requirements of uncertain form of the encryption key. Since the encryption key space is so big (when the quantities of the pixels is magnitude), the randomness of P guarantees the security of the encryption. Of course, these transforms in **Set** have their own important properties. All of these are discussed in Section "Properties of the sub-operators of scrambling operators". On the other side, we clarify that the uniform scrambling effect can not be obtained, because the pixel is a micro object, whose exact position can not easily be captured. Especially, the perfect scrambling effect may need lots of quantum transform circuits to implement the transform of some pixels in sometime, but this may lead to great overhead of the physical space to express the transform circuit (after all, the complexity of each

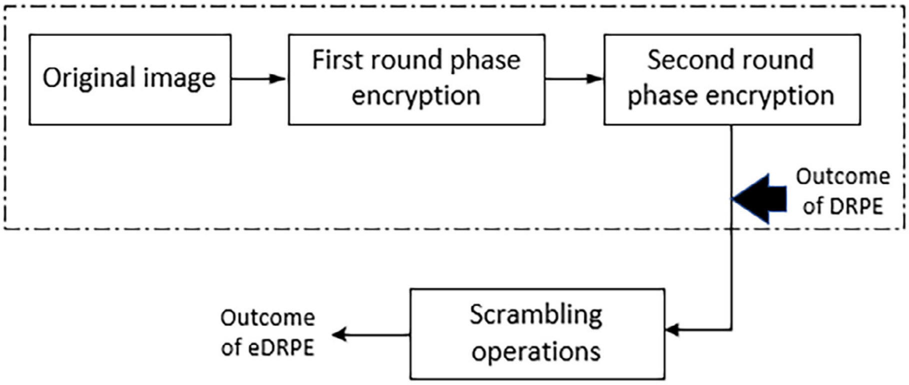
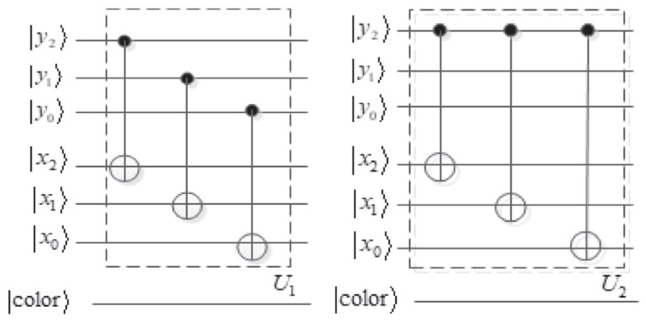


Fig. 3. Diagram of eDRPE algorithm. eDRPE scheme includes DRPE and scrambling operation based on DRPE two steps, where DRPE includes two operations: first round phase encryption and second round phase encryption.



(a) (b)

Fig. 4. Pixel matrix used to analyze the quantum image is as the right picture in Fig. 1. (a) implements the image row folding. (b) implements that the above half of the data in pixel matrix is folded right and left.

operator is $\boldsymbol{O}$(n), which see in Section "Properties of the sub-operators of scrambling operators"). However, the approximation of perfect scrambling is feasible. In our paper, we demonstrate a scrambling tendency when the combination of the scrambling operator becomes more complex (the detail see Section "Operator construction and experimental results"). That's to say, (based on the right operation procedure,) the more complex of the transform circuit, the better scrambling effect. The double-randomness of the scrambling operator is the security guarantee of the eDRPE.

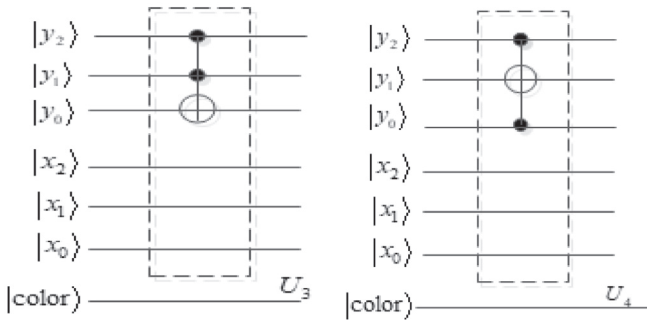
Properties of the sub-operators of scrambling operators

Transform sets

In this section, a couple of quantum circuits are devised to construct a foundation transform set, which includes different kinds of functions, see Figs. 4–6.

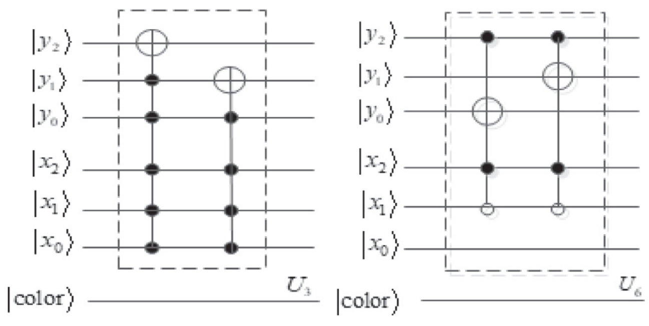
As a model to interpret the function of each specified transform, quantum images with $(3 + 3 + Color)$ qubits are taken as an example. That is, the quantum transforms are 8×8 pixel matrix which are to be applied on the state of quantum image to show the different actions of the transforms. We also can extend the qubits of the quantum image from 3 to n. Generally, see Fig. 2, where n qubits $y_0, y_1, \ldots, y_{n-1}$ and $x_0, x_1, \ldots, x_{n-1}$ denote Y and X axis coordination information of the quantum image, respectively.

Based on the assumption of $(3 + 3 + color)$ qubits quantum image, it is known from Fig. 1 (a) that the coordinate vectors of the quantum image could be expressed as $|y_2y_1y_0\rangle|x_2x_1x_0\rangle$. Let X be the Pauli operator, and I is a two dimensional identity matrix, then we define the operators U_1, U_2, U_3, U_4, U_5, U_6 individually. U_1 is defined as



(a) (b)

Fig. 5. Row folding(Pixel matrix used to analyze the quantum image is as the right picture in Fig. 1). Two quantum circuits implement row exchange, respectively. (a) U_3 exchanges the pixel information in row $1st$ and row $2st$. (b) U_4 exchanges the pixel information in row $1st$ and row $3rd$.



(a) (b)

Fig. 6. Pixel matrix used to analyze the quantum image is as the right picture in Fig. 1. (a) Function of U_5 implements the pixel exchange between the pixel point c_{64} and c_{56} with multiple K-Cnot gate. (b) exchange some elements in the inner of sub-block. Concretely, the local small sub-block is folded between row 1 row 2 and row 3 row 4 in column 3 and 4.

$$U_1 \equiv U_{1x}U_{1y}U_{1z}, \tag{13}$$

and U_{1x}, U_{1y}, U_{1z} are defined as follows, respectively.

$$\begin{cases} U_{1x} = |1\rangle\langle 1| \otimes\ I^{\otimes 2} \otimes\ X \otimes\ I \otimes\ I + \\ |0\rangle\langle 0| \otimes\ I^{\otimes 5}, \\ U_{1y} = I \otimes\ |1\rangle\langle 1| \otimes\ I \otimes\ X \otimes\ I \otimes\ I + \\ I \otimes\ |0\rangle\langle 0| \otimes\ I^{\otimes 4} \\ U_{1z} = I^{\otimes 2}|1\rangle\langle 1| \otimes\ I \otimes\ I \otimes\ X + \\ I^{\otimes 2}|0\rangle\langle 0| \otimes\ I^{\otimes 3}. \end{cases} \tag{14}$$

U_2 is defined as $U_2 = U_{2x}U_{2y}U_{2z}$. U_{2x}, U_{2y}, U_{2z} are defined below, respectively.

$$\begin{cases} U_{2x} = |1\rangle\langle 1| \otimes\ I^{\otimes 2} \otimes\ X \otimes\ I \otimes\ I + \\ |0\rangle\langle 0| \otimes\ I^{\otimes 5}, \\ U_{2y} = |1\rangle\langle 1| \otimes\ I^{\otimes 2} \otimes\ I \otimes\ X \otimes\ I + \\ |0\rangle\langle 0| \otimes\ I^{\otimes 5}, \\ U_{2z} = |1\rangle\langle 1| \otimes\ I^{\otimes 2} \otimes\ I \otimes\ I \otimes\ X + \\ |0\rangle\langle 0| \otimes\ I^{\otimes 5}. \end{cases} \tag{15}$$

Fig. 4 is the corresponding quantum circuits of U_1 and U_2. The function of Fig. 4 (a) folds a whole row, and Fig. 4 (b) folds upper half of the pixel matrix.

U_3 is defined as

$$U_3 = |11\rangle\langle 11| \otimes\ X \otimes\ I^{\otimes 3} + \sum_{y_2y_1 \in \{0,1\}^2, y_2y_1 \neq 11} |y_2y_1\rangle\langle y_2y_1| \otimes\ I^{\otimes 4}. \tag{16}$$

and U_4 is defined as

$$U_4 = |101\rangle\langle 111| \otimes\ I^{\otimes 3} + |111\rangle\langle 101| \otimes\ I^{\otimes 3} + \sum_{y_2y_1y_0 \in \{0,1\}^3, y_2y_0 \neq 11} |y_2y_1y_0\rangle \langle y_2y_1y_0| \otimes\ I^{\otimes 3}. \tag{17}$$

Fig. 5 is the quantum circuits of U_3 and U_4. The function of Fig. 5 (a) swaps two rows when $y_2y_1 = 11$, and Fig. 5 (b) also swaps the first rows and the third row.

U_5 is defined as

$$U_5 = U_{5x}U_{5y}, \tag{18}$$

and U_{5x}, U_{5y} are defined as follows, respectively.

$$\begin{cases} U_{5x} = |111111\rangle\langle 011111| + |011111\rangle\langle 111111| + \\ \sum_{y_1y_0x_2x_1x_0\in\{0,1\}^5, y_1y_0x_2x_1x_0\neq 11111} I\otimes\ |y_1y_0x_2x_1x_0\rangle\langle y_1y_0x_2x_1x_0|, \\ U_{5y} = I\otimes\ |11111\rangle\langle 01111| + I\otimes\ |01111\rangle\langle 11111| + \\ \sum_{y_0x_2x_1x_0\in\{0,1\}^4, y_0x_2x_1x_0\neq 1111} I^{\otimes 2}|y_0x_2x_1x_0\rangle\langle y_0x_2x_1x_0|. \end{cases}$$

U_6 is defined as

$$U_6 = U_{6x}U_{6y}, \tag{19}$$

and U_{6x}, U_{6y} are defined as follows,

$$\begin{cases} U_{6x} = |1\rangle\langle 1| \otimes\ I\otimes\ (|0\rangle\langle 1| + |1\rangle\langle 0|) \otimes\ |10\rangle\langle 10| + \\ \sum_{y_2y_1y_0x_2x_1x_0\in\{0,1\}^6, y_2x_2x_1\neq 110} |y_2y_1\rangle\langle y_2y_1| \otimes\ I\otimes \\ |x_2x_1x_0\rangle\langle x_2x_1x_0|, \\ U_{6y} = |1\rangle\langle 1| \otimes\ (|0\rangle\langle 1| + |1\rangle\langle 0|) \otimes\ I\otimes\ |10\rangle\langle 10| + \\ \sum_{y_2y_1y_0x_2x_1x_0\in\{0,1\}^6, y_2x_2x_1\neq 110} |y_2y_1\rangle\langle y_2y_1| \otimes\ I\otimes \\ |x_2x_1x_0\rangle\langle x_2x_1x_0|. \end{cases}$$

Fig. 6 is the quantum circuits of U_5 and U_6. The function of the left K-Cnot gate in Fig. 6(a) swaps two pixels, and the right K-Cnot gate realises 4 pixels in the first column exchanged each other. Fig. 6(b) implements sub block folding operation.

Let a transform set $\{U_1, U_2, U_3, U_4, U_5, U_6\}$ be **Set**. An unitary operator which swaps two specified pixels in pixel matrix of quantum image, is called point swapping (PS). An unitary operation transform which folds the pixels of a row or column in a pixel matrix of a quantum image is called row swapping or column swapping (RS/CS). Similarly, an unitary transform which folds a symmetric sub-block in a pixel matrix of a quantum image is called sub-block folding swapping (SBS).

We then ask what properties of **Set** have exactly? In order to answer this question, we have the following result.

Theorem 1. **Set** $= \{U_1, U_2, U_3, U_4, U_5, U_6\}$ *can be categorized into three kinds according to the function of the quantum circuits. The three kinds of the transforms are point swapping, row (or column) swapping, and sub-block swapping.*

Proof. According to the functional definition about PS, RS/CS and SBS, the transforms in **Set** could cover the functional distribution of all these circuits:

(1) All transforms implementing the PS operation form a set_1, $set_1 = \{U_5, ...\}$.
(2) All transforms implementing the RS/CS operation form a set_2, $set_2 = \{U_1, U_3, U_4, ...\}$.
(3) All transforms implementing the SBS operation form a set_3, $set_3 = \{U_2, U_6, ...\}$. Especially, the transform U_6 supports the operation of the arbitrary inner sub-block folding.

Since

$$set_1 \bigcup set_2 \bigcup set_3 = \textbf{Set}, \tag{20}$$

the result holds. Note that, PS exchanges two pixels which are in the same row or same column (The functions provided by PS is different from the operation of arbitrary two pixels swapping in Section "Arbitrary pixel swapping"). □

The elements in **Set** only provide a paradigm for explaining the foundation functions of these transforms. However, many other different quantum circuits can be added into **Set**, because all these circuits can also provide PS, SBS, and RS/CS functions. Actually, other transforms which are not yet included into **Set** are used to scramble the pixels at other different positions. That is to say, one quantum transformation-example circuit chosen from some set of **Set** only provides a specific functional-example of a class of functions. However, the quantum circuits in **Set** is enough for us to explore the typical characteristic of universal functional transform in reality. Though variants of such quantum circuits do not add into set **Set**, it can be used according to our real requirements in the applications, such as, some transforms applied in Fig. 10. Fig. 10 shows that, no matter how to change the form of the quantum circuits or how complicated the form of the transform operator is, the functional transform operators still could be generalized into **Set**.

Arbitrary pixel swapping

A set $\mathbf{set_c}$ is called complete transform set, if $\mathbf{set_c}$ is a minimal set, and any of the transform results could be realised with the elements in this minimal set.

The goal of scrambling transform is to mix and re-distribute all pixels of a meaningful image to make the color of the final image meaningless thoroughly. The ideal case is when the final state of the transformation can be seen as uniformly mixed.

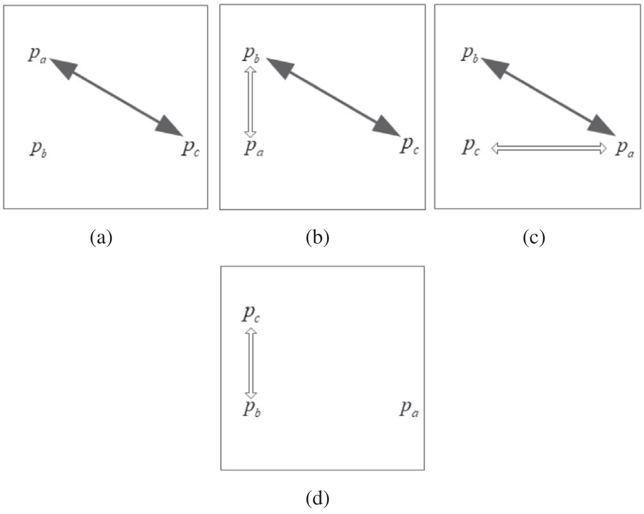
For convenience, we use a sign $\mathbf{x} \leftrightarrow \mathbf{y}$ to express the swapping operation of two values x and y, where x and y are the pixels in the same row or same column. That is, $\mathbf{x} \leftrightarrow \mathbf{y}$ defines a PS operation about x and y.

According to the definition above, and based on the features of **Set**, we have the following result.

Theorem 2. *Point swapping (PS) operation constructs a* $\mathbf{set_c}$. *That is,* $\mathbf{set_c} = \{PS\}$.

Problem reduction. Given two quantum images Q_a and Q_b, where Q_a is the original image, Q_b is a scrambling result via applying arbitrary transform operator(s) on Q_a. Let us consider now what is the minimal transform set which can realise any transform effect (such as Q_b) of pixel scrambling, if we find it in **Set**, it must be $\mathbf{set_c}$. Because the essence of scrambling transform for quantum image is the pixel exchange of quantum image, and note that the pixels in the same row or column can be swapped by the transform chosen from PS, so, the problem is reduced to consider how to transform those pixels which are not in the same row or column successfully.

Proof. Assume that Fig. 7 (a) is a pixel matrix **M**. Given two pixels, $P_a, P_c \in \mathbf{M}$. Note that, P_a and P_c in pixel matrix are not in the same row (column). Our goal is to implement the swapping the pixels P_a and P_c with basic transforms chosen from **Set** successfully. Because we have known that PS can swap arbitrary two pixels in the same column or same row, so, for any two pixels which are not in the same row or same



(a) (b) (c)

(d)

Fig. 7. Assume that the box is a pixel matrix M, where the box is full of the pixels of the image. $P_a, P_c \in M$. (a) describes the state of the two pixels p_a and p_c which are to be swapped, and p_b is prepared. (b) is the result of swapping p_a and p_b. (c) is the result of swapping p_a and p_c. (d) is the result of swapping p_b and p_c.

column, such as the two pixels P_a and P_c in Fig. 7, we exchange P_a and P_c with the Algorithm 1

```
Algorithm 1: swap($p_a$, $p_c$)
Input:A pixel matrix $M$, which includes pixels $p_a$, $p_c$, and $p_b$.
Output:A pixel matrix $M$, where $p_a$ and $p_c$ are swapped.
1 Choose a pixel,let it be $P_b$, which is in the same row with $P_c$, and same column with
    $P_a$, see Fig. 7(a)
2 $P_a \leftrightarrow P_b$, see Fig. 7 (b)
3 $P_a \leftrightarrow P_c$, see Fig. 7 (c)
4 $P_c \leftrightarrow P_b$, see Fig. 7 (d)
```

(Suppose that, how to find the medial pixel p_b between p_a and p_c is the work of operation system of quantum computer.) Now, we exchange p_a and p_c successfully through repeatedly using single PS operation. That is to say, arbitrary pixel point exchanged can be implemented with PS in four steps. Obviously, according to the analysis above, all the basic transforms employed to implement the swapping of two pixels which are non in same row or column are the elements of $\mathbf{set_c}$, since PS is the only one kind of operation involved in the transformation, $\mathbf{set_c} = \{PS\}$. □

Decomposition of complicated operator

Some quantum gates, such as Hardamard gate, Pauli gate, Toffoli gate, are the statistical objects when we consider the complexity of the quantum algorithm. The left quantum circuit in Fig. 8 is a K-Cnot gate. K-Cnot gate is an important transform, especially in PS operation. However K-Cnot gate is not a common quantum gate, so, before analyzing the complexity relation between K-Cnot gate and common quantum gates, we should decompose K-Cnot gate into common gates. The right quantum circuit in Fig. 8 gives an equivalent form of the K-Cnot quantum circuit using 2-Cnot gates [27] (Toffoli gates). Fig. 8 shows that one multiple controlled-not gate can be expressed with multiple Toffoli gates. The decomposition of K-Cnot quantum circuit shows a numerical relation that, the function of one K-Cnot gate is equivalent to $2K - 3$ ($K \in \mathbf{Z}^+$, $K \geqslant 2$) Toffoli gates [27].

Complexity of diffusion transform

From Section "Arbitrary pixel swapping", we know $\mathbf{set_c} \subset \mathbf{Set}$. Since the essence of all the transforms of quantum image is the pixel shifting, any transformation requirements can be realised by applying the elements of $\mathbf{set_c}$ multiple times. The consequent problem is that $\mathbf{set_c} = \{PS\}$ can resolve all the transform requirements, why we need other types of operation in **Set**? In order to deal with this question we are going to proceed in the following.

For conveniently clarify the theory in the following part, we will now try to introduce and explain the necessary conceptions in advance. If the data to be processed is a block (data pack) with columns of even length, which can be covered by one rectangle or square, then we call such a data block as regular data block (RDB).

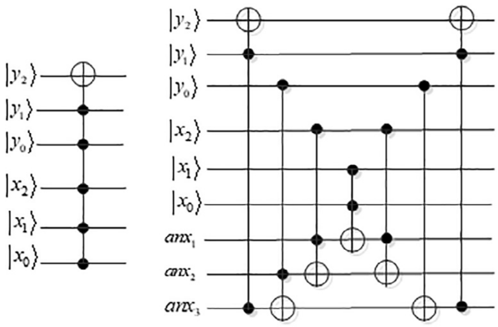


Fig. 8. The left circuit is a multiple K-Cnot gate. The right circuit is the equivalence of the quantum circuit of the left.

The block folding operation could quickly scramble the pixel information of the pixel matrix. The operation combining SBS with PS or the operation combing CS/RS with PS is called diffusion transform (DT).

Let $|RDB|$ denote the number of data elements in a data block RDB. Since SBS folds a RDB, the minimal of $|RDB|$ is satisfied with

$$|RDB| = 2. \tag{21}$$

In addition, the complement of $\mathbf{set_c}$ could be expressed as

$$\bar{\mathbf{set_c}} = \mathbf{Set} - \mathbf{set_c} = \{SBS,\ CS/RS\}. \tag{22}$$

According to the result of Section "Decomposition of complicated operator", we can know that, the complexity of the operation tFS, CS/RS, SBS are equivalent, and all of which are $\boldsymbol{O}$(n), the transforms provided in $\bar{\mathbf{set_c}}$ are the coarse-grained operation, and the transform in $\mathbf{set_c}$ is fine-grained operation. The transformation efficiency provided by $\mathbf{set_c}$ is less than the efficiency provided by $\bar{\mathbf{set_c}}$. About the complexity of DT and PS, we have the following result.

Theorem 3. *Assume that a pixel matrix is $M_{n\times n}$. Let the data to be processed be in the upper part or lower part in a $M_{n\times n}$ be a RDB. Suppose $|RDB|$ is $O(n^2)$, the complexity of SBS and PS to scramble all these pixels in RDB are $O(n)$ and $O(n^3)$, respectively.*

Proof. It could be got that from Section "Decomposition of complicated operator", PS operation needs $O(n)$ Toffoli gates. Since the steps of PS operation is proportional to $|RDB|$, and the data magnitude of pixel information reaches $O(n^2)$, the complexity of applying PS to complete this task is

$$O(n)^*O(n^2) = O(n^3). \tag{23}$$

Because of the equivalence of the left and the right quantum circuit in Section "Decomposition of complicated operator", the complexity of SBS operation (see Fig. 6, U_6) is $O(n)$. On the other hand, because the data to processed is a RDB, and this RDB satisfies $O(|RDB|) = O(n^2)$, e.g., if $|M_3| \approx \frac{1}{5}n^2$($|M_3|$ is column of even length required), that is M_3 includes $\frac{1}{5}M_{exp}$ pixel information (it is possible), then the folding task swaps $\frac{1}{10}M_{exp}$ pixels approximately, and only once SBS operation employed is sufficient. That is, to complete the task required, the complexity of applying SBS transform in our algorithm is

$$O(1)^*O(n) = O(n). \tag{24}$$

Therefore, the result holds. □

Example. In order to explain the transformation power of DT, we take the following problem as an example. Assume that we have a pixel matrix M_{exp} (see Fig. 9), which is a two dimensional (X-Y axis) plane. M_{exp} includes two parts, upper part and lower part. Three RDB, which are M_1, M_2, M_3 in the upper part, and three RDB which are M_4, M_5, and M_6 in the lower part. All these RDB in M_{exp} are with columns of even length. Some pixels p_i, ($i \in \{a, b, c, d, e, f, g, h\}$), are satisfied with

$$p_a, p_b, p_c, p_d \in M_1, \tag{25}$$

and

$$p_e, p_f, p_g, p_h \in M_6. \tag{26}$$

Assume that the local actions required:

Step 1. Exchange two - arbitrary pixels, one pixel is called $p_x \in M_{exp}$ (note $p_x \notin M_1$), another one is called $p_i \in M_1$.
Step 2. Folds M_1.
Step 3. Folds M_2.
Step 4. Folds M_3.
All these folding operation are along with the direction of the arrow in the upper part of the Fig. 9. Similarly, in the lower part

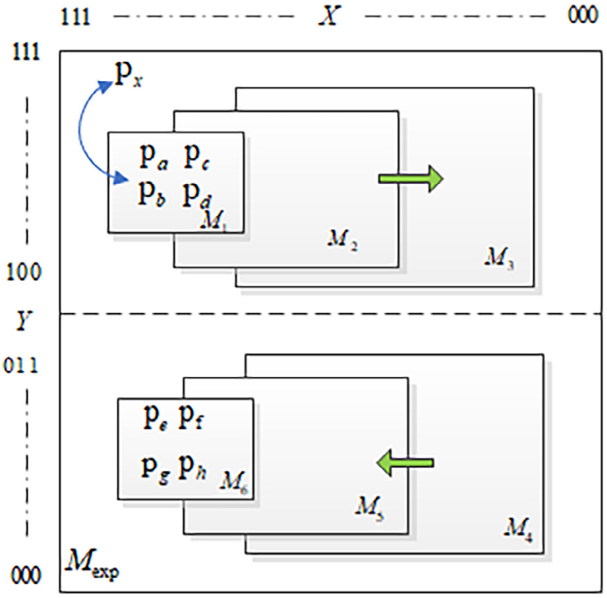


Fig. 9. M_{exp}, example for diffusion of the packed RDB pixel information.

of Fig. 9, assume that the local operation required:
Step 5. Fold RDB M_4.
Step 6. Fold RDB M_5.
Step 7. Fold RDB M_6.

All these folding operations are along with the direction of the arrow in the lower part of the Fig. 9. How can we deal with this problem simply?

One way can swap each pixel by using PS. That is to say, for each pixel in the rectangle, we execute the Algorithm 1 once. So to complete this work, more than

$$4(|M_1| + |M_2| + |M_3| + |M_4|)^* \frac{1}{2} \tag{27}$$

$$= 2(|M_1| + |M_2| + |M_3| + |M_4|) \tag{28}$$

and

$$4(|M_4| + |M_2| + |M_5| + |M_6|)^* \frac{1}{2} \tag{29}$$

$$= 2(|M_4| + |M_2| + |M_5| + |M_6|) \tag{30}$$

times calling PS operation are called for the upper part and lower part of M_{exp}, respectively. When $|RDB|$ is big, the total times of calling PS is numerous. Applying DT to implement the scrambling is another complexity result. Since the data block to be processed is a RDB, SBS or CS/RS operations could be employed to cope with the folding operation, e.g., seven steps folding operation only 6 times SBS folding operations are enough.

Pixel-shifting and possible periodic cycling

Assume that the pixel matrix of a quantum image is M, then multiple times applying the transforms in **Set** may lead to the circulating loop. For example, if we swap arbitrary two pixels p_a and p_b using some transform, such as U_1, within two steps, then the distribution of the pixel information goes back to its initial state. Algorithm 2 gives an example of such a simple loop. That is, applying the same operator twice on the same data may lead - that the transformed effect is offset.

Algorithm 2: Recycle(p_a, p_b)

```
Input: $M$, $p_a, p_b \in M$.
Output:The pixel matrix $M$, where $p_a$ and $p_b$ are not swapped.
$U_1$: $p_a \leftrightarrow p_b$, we get $p_b, p_a$
$U_1$: $p_b \leftrightarrow p_a$, we get $p_a, p_b$
```

In the scrambling transform, circulating loop may have more complicated form, e.g., take four elements, $p_a, p_b, p_c, p_d \in M$, as an example, assume that some operators U_1, U_2, U_3, U_4 are applied sequentially (the operation about the pixel shifting at different position is corresponding to the different unitary transform quantum circuit), Algorithm 3 shows an greater circulating. For the continuous data block (connected but not intersected), such issues may happen under some special conditions.

Algorithm 3: swap(p_a, p_b)

```
Input: $M$, $p_a, p_b, p_c, p_d \in M$.
Output: The pixel matrix $M$, where $p_a$ and $p_d$ are not swapped.
$U_1$: $p_a \leftrightarrow p_b$, we get $p_b, p_a, p_c, p_d$
$U_2$: $p_a \leftrightarrow p_c$, we get $p_b, p_c, p_a, p_d$
$U_3$: $p_a \leftrightarrow p_d$, we get $p_b, p_c, p_d, p_a$
$U_4$: $p_a \leftrightarrow p_b$, we get $p_a, p_b, p_c, p_d$
```

To avoid this case, we introduce DT at least once to non-symmetrically destroy the symmetry of the transform result. DT combining other transforms (assumed U_1, U_x) are applied to diffuse the local information to other places, then the risk of cycling is downgraded, see Algorithm 4.

Algorithm 4: swap(p_1, p_2)

```
Input: $M$, $p_1, p_2, ..., p_x, ..., p_n \in M$, and DT operator.
Output: The pixel matrix $M$, where $p_1$ and $p_x$ are swapped.
$U_1$: $p_1 \leftrightarrow p_2$
DT
…
$U_x$: $p_x \leftrightarrow p_1$
```

Operator construction and experimental results

Construction about scramble operator P

Section "Enhanced double random phase encoding" gives a scheme of encryption, where the critical point of implementing this algorithm is to design a good scramble operator P. Section "Properties of the sub-operators of scrambling operators" lists some properties or rules for constructing such transforms. Fig. 10 is an example of such a transform P based on the properties or rules. Apparently, Fig. 10 shows that P is a combinatorial operator constructed with 12 variants of transforms in **Set**. P will be used in Section "Enhanced double random phase encoding", where step 5 of encryption algorithm, and step 1 of its decryption are involved.

To show the potential of P, we apply P to Lena's gray image. Then, some characteristics of such a process are depicted in Fig. 11, where Fig. 11 (a) is the input of Lena's original image, and Fig. 11 (c) is the outcome after applying transformation P on Fig. 11 (a). Fig. 11 (b) and (d) are the outcomes of the histogram for image Fig. 11 (a) and Fig. 11 (c), respectively. It is obvious that Fig. 11 (b) and (d) are same. This means that, the function of the operator representing by Fig. 10 only shifts the positions of the pixels in the quantum image.

Experimental results and analysis

Encryption simulation with Matlab (R2016a, 64bit) on win64 based on DRPE algorithm, Fourier transform, and scrambling transform, outputs an experimental result, see Fig. 13 and Fig. 14. Since the state representation of quantum image captures a mapping relation between phase and the brightness of the pixel point, we should consider the color changes in the course of executing DRPE algorithm. Running

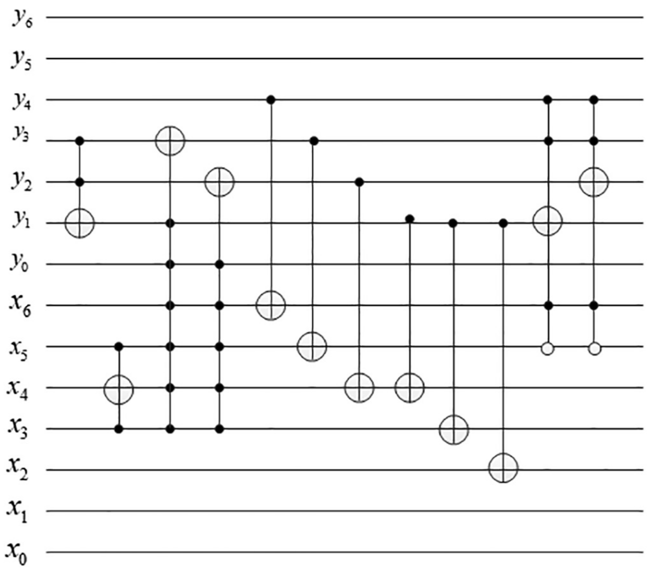
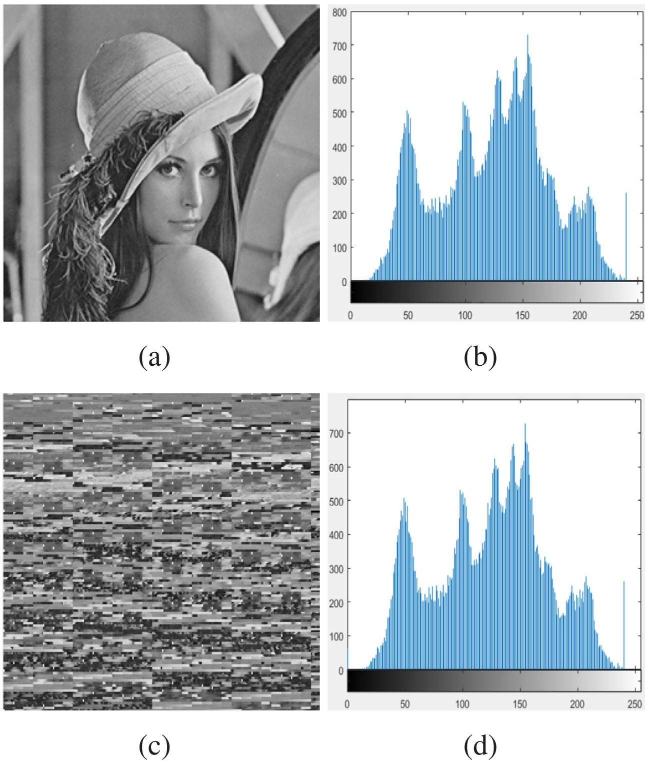


Fig. 10. Quantum circuit of unitary operator P will be used to implement the image transformation to enhance the DRPE algorithm.



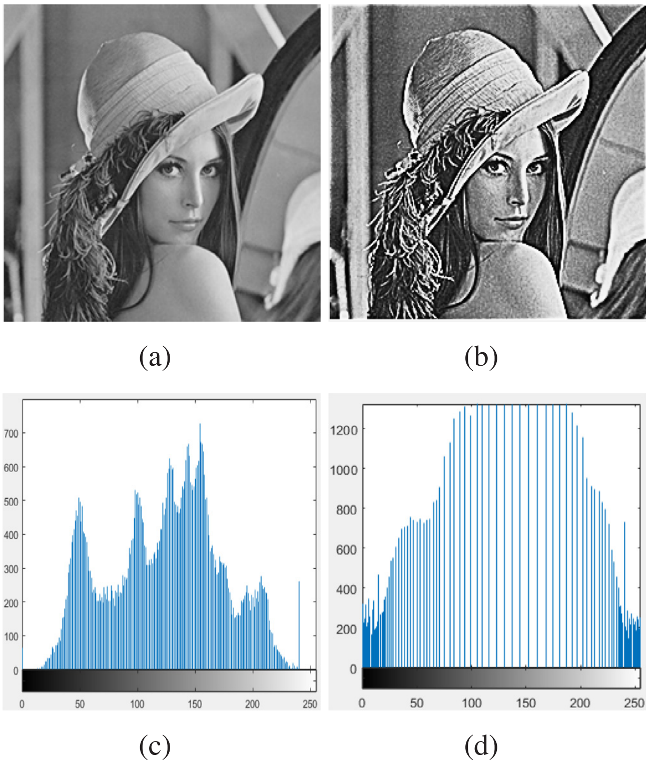
(a) (b)

(c) (d)

Fig. 11. (a) Experimental original image. (b) the histogram of the original image. (c) the transformed image of (a) with transform P in Fig. 10. (d) the histogram of the transformed image.

DRPE algorithm will lead to the color change of the quantum image. Here, high pass filter is introduced to simulate the color change of the quantum image. Compared with the histogram of the original image and a transformed image which is the outcome after applying DRPE algorithm, Fig. 12(a) is the original image, Fig. 12(b) is the simulating result of DRPE algorithm, Fig. 12(c) is the histogram of the original image, and Fig. 12 (d) is the histogram of the outcome of DRPE algorithm.

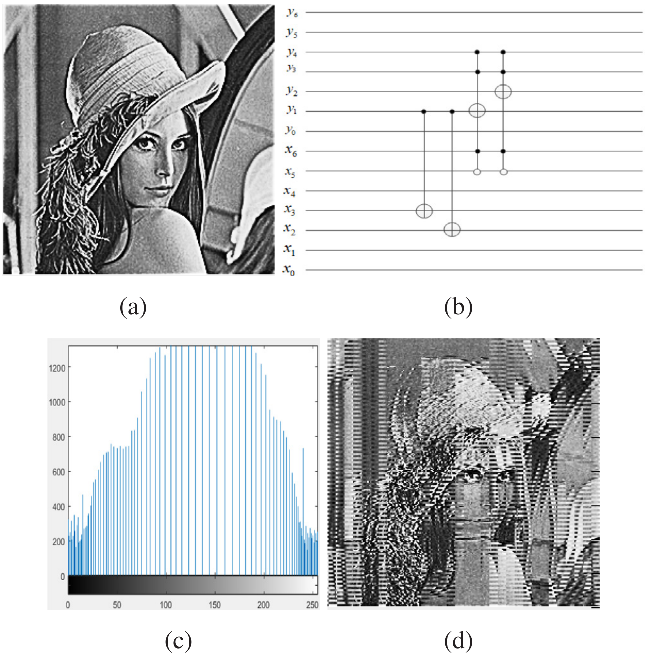
An transform is said to be good, if it leads to a good transformation (scrambling) effect (that is it is practically impossible to get the original image from an encrypted one). Theoretically, better effect means more complicated P, and more complicated P means bigger space occupied. So, the optimal transform can not be provided here for the limited size of the space. Because we can not completely give such an optimal transform and its optimal transformation effect directly, the only way that we can facilitate is to prove that there exists a tendency of transforming effect. Therefore, Fig. 10 is just an example used to prove the



(a) (b)

(c) (d)

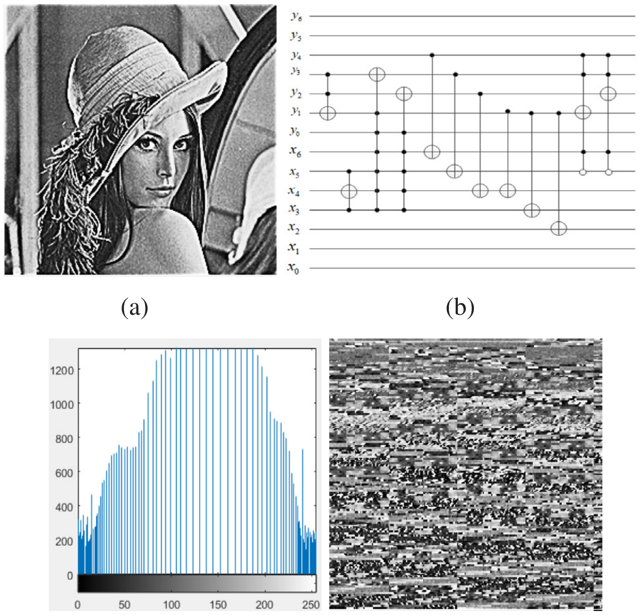
Fig. 12. (a) The original Lena's image. (b) Lena's image after simulating using DRPE. (c) The histogram of Fig. 12(a). (d) The histogram of Fig. 12(b).



(a) (b)

(c) (d)

Fig. 13. These figures are used to compare with Fig. 14. (a) Lena's image after simulating using DRPE. (b) The transform would be applied on Fig. 13(a). (c) The histogram of Fig. 13(a). (d) The result of applying Fig. 13 (b) on Fig. 13(a).



(a) (b)

(c) (d)

Fig. 14. These figures are used to compare with Fig. 13. (a) Lena's image after simulating using DRPE. (b) The transform would be applied on Fig. 12(a). (c) The histogram of Fig. 14(a). (d) The result of applying Fig. 14 (b) on Fig. 12(a).

existence of the transformation effect trends.

In doing so, we construct a partial transform P' via extracting partial components-transform from P. Fig. 13 (a) is such a transform P'. Then we can compare the transformation result of applying P' (see Fig. 13 (b)) and the result of applying P (see Fig. 14(b)). The traces in Fig. 13 (b) show that some of the pixels in pixel matrix are scrambled, but many pixels remain at its own original position. This is because that P' (namely Fig. 13 (a)) is too simple, and only a small number of pixels are moved.

Figs. 13(a) to 14(a) construct a trend. The result of applying Fig. 13(a) with Fig. 14(a) are Fig. 12(a) and Fig. 14(b), respectively. Fig. 13(b) to Fig. 14(b) also construct a trend. So, to compare Fig. 12(b) and Fig. 14(b) can get our conclusion. The actual operation results intuitively show the trend: as the number of scrambling operations increases, the pixels become more uniform, because the scrambling effect in Fig. 14(b) is better than Fig. 13(b).

All of these indicates that, the transform constructed with combinatorial operator chosen from the elements of **Set**, can effectively scramble the quantum image. Since scrambling operation is applied on the outcome of DRPE algorithm, the effectiveness of scrambling shows the effectiveness of eDRPE algorithm. On the other side, the transform has made the original image meaningless. Even though Fig. 14 is an outcome of an example-transform P, the analysis about how to attack this encryption method has been a hard problem.

Advantages discussion

Since eDRPE is an encryption algorithm, to discuss the strength of the encryption key is essential. In our scheme, the security of eDRPE is guaranteed by the longitudinal double-randomness: the randomness of choosing the example-operator from **Set**, and the randomness hidden in scrambling behavior of the example-operator itself. This feature is different from case of the single-randomness [9,11]. Taking **Set** discussed in Section "Properties of the sub-operators of scrambling operators", into consideration, we emphasize the following critical points of the advantages of the encryption key from two perspectives: comparison with theoretical and experimental analysis.

Comparison with theoretic analysis

Existence of the optimal operator

The effectiveness of encryption is the most important for encryption algorithm. The optimal effect for the transformation is that the pixel information of the quantum image is uniformly mixed after applying the scrambling transform. That is to say, the brightness distributed in the transformed pixel matrix is uniform everywhere. Since the pixel is a micro-concept, it is difficult to accurately control the brightness clustering where the dark of bright pixel is not uniformly distributed. Because $\mathbf{set_c}$ has perfect property, so the optimal operator exists. This can be concluded from the Theorem 2 theoretically. So, if the quantum circuit of the transform could be complicated enough allowed, then the transform with any complicated functional requirement could be implemented. That is, the optimal operator exists theoretically, and it can transform a quantum image to an uniform state.

In contrast to other similar and published algorithms, such as [28], where the authors end the experiment just via applying some transforms to the state of quantum image to get a running result, the detailed mathematical evidence is not obvious. The analysis in Section "Properties of the sub-operators of scrambling operators" shows that our algorithms are guaranteed by strict theoretical evidence.

Multiple times scrambling vs. twice scrambling

About the times of applying the scrambling transforms on the state of the quantum image, we summarize the following advantages:

(1) The operator space constructed with the various transforms chosen from the elements of **Set** is big enough. We can call **Set** a functional set (FS). That is,

$$FS = \{PS, CS/RS, SBS\}. \tag{31}$$

Obviously, the first, since shifting each pixel in a pixel matrix individually refers to a specified quantum transform circuit, for a pixel matrix with size $n \times n$, n^2 transforms is needed, which represents an uniquely functional behavior. The functions in all these n^2 transforms can be reduced to FS. The second, focusing on different block data in pixel matrix, we can construct different block transform. For the attackers, the block transform can lead to the position of the pixel unpredictable.

(2) Applying the same set of component-operators (foundation transform) with different consequence leads to different transformation results. So the same foundation transform with different consequence represents different multiple transforms.

(1) and (2) construct an big enough space. This means we can use sufficient keys to scramble the quantum image multiple times. Compared with the previous encryption algorithm for the quantum image in [29] where twice scrambling operation are used, our algorithm which allows multiple times applying encryption key shows a certain advantage. Based on this, the randomness provided by the sufficient big random transformation space (and its random combination) realises the enough ability against the attack.

Comparison of irregularity and sensitivity of keys

Song et al. in [28] use

$$x_{\delta+1} = \frac{4\eta'^2 x_\delta(1 - x_\delta)}{1 + 4(\eta'^2 - 1)x_\delta(1 - x_\delta)} \tag{32}$$

as a logistic map to generate key for diffusing scrambling the pixel information. The vulnerability of the algorithm is that if the attackers discover this rule, then the encryption method maybe ineffective. In our paper, we destroy this regular form, and substitute the previous operator generation method with the transform operators randomly chosen according to randomness rules. So, this increases the great difficulty for attackers beginning their works.

In addition, concerning the keys sensitivity, using a wrong decoding operator to decode an encrypted image is equivalent to an encryption operation once more. For the transform operator P, the decryption algorithm adopts the principle $PP^{\dagger} = I$, e.g., if we use U_1, U_2, and U_3 to construct an operator $P = U_1U_2U_3$ to encrypt a quantum image, then we should use $P^{\dagger} = U_3^{\dagger}U_2^{\dagger}U_1^{\dagger}$ to decrypt $P|\psi\rangle$, and we get the original quantum state $|\psi\rangle$ by using $|\psi\rangle = PP^{\dagger}|\psi\rangle$.

Efficiency of scrambling

Let G_{ps}, G_{rscs}, G_{sbs} be the granularity for three transforms PS, RS/CS, and SBS, respectively. According to the analysis in Section "Complexity of diffusion transform", these variables are satisfied with $G_{ps} < G_{rscs} < G_{sbs}$, and the complexity to finish one RDB folding task, $O(n)$ and $O(n^3)$ for SBS and PS are needed, respectively. That is to say, using this coarse-grain operation, the efficiency of implementing the random scrambling will be improved drastically. Because there exist the advantage of SBS operation, we, in this part, emphasize that the efficiency of our transform P is higher than previous methods [28] where the complexity is $O(n^2)$ for diffusion.

The typical characteristic of **Set** emphasized in this paper is $\mathbf{set_c} \subset \mathbf{Set}$. The special ability of $\mathbf{set_c}$ does not taken seriously in previous quantum transform algorithm [30,31]. $\mathbf{set_c}$ guarantees that there exists the uniformly mixed state that could be got theoretically. All these advantages about **Set** show that the transform P constructed with the elements of **Set** is strong credible.

Comparison with experimental methods

The serial experiments in Section "Operator construction and experimental results" prove that, the more complex transform circuit, the better the scrambling effect. This indicates that, the uniform scrambling effect could be obtained if a sufficient complex of the transform circuit is constructed. Contrasting with other papers, such as [14,28], where directly give an uniform mixed result, the encryption method of our paper is more convincing.

Conclusions

To overcome the vulnerability of DRPE algorithm encrypting the quantum image, we have applied an additional scrambling mechanism to make the outcome of DRPE algorithm well scrambled. A set **Set** has been constructed based on the foundation transforms, and all of its rich properties have been discussed. Since shifting each pixel or folding a data block in a pixel matrix refers to the specified quantum circuits, to construct such a combinatorial transform which can shift all the pixels should occupy a big physical space. Therefore, depicting such a completed circuit example set is impossible. Since $\mathbf{set_c} \subset \mathbf{Set}$ can optimally scramble the quantum image, for verifying scrambling tendency when the transform constructed based on **Set** becomes complicated more and more, the scramble effect becomes better and better Then we can conclude that the transform is effective and the optimal transform exists when we correctly use scrambling rule. In doing so, a complex transform P has been constructed using some transforms example in the elements of **Set**. For comparison analysis, a transform P (namely, Fig. 10, or Fig. 14) and its simplicity version(see Fig. 13(b)) have been constructed and applied on the same image individually. The experimental result has shown that, the more complicated scrambling quantum circuit, the better the scrambling effect. This has indicated that the effectiveness of the transform and the existence of the optimal transform constructed based on **Set** are verified rationally. The theoretic analysis has shown that the combinatorial operator used for image encryption based on **Set** have some advantages than the algorithms proposed before. All these characteristics determine that the key strength of the combinatorial transform operator based on **Set** is credible. We summarized the algorithm with the following viewpoints: (1) **Set** can provide potential uniform ability. (2) Owing to the high complexity of the K-Cnot gate, the transform circuit can not be infinitely complex. So the maximum possibility is that we only approximate uniform shifting pixels as much as possible. (3) Since the enough big encryption key space can be provided by **Set**, and the operator has the sensitivity for the decryption, the longitudinal double-randomness and the sensitivity of the transform guarantees the security of the encryption. (4) All these features can ensure the rationality and effectiveness of the eDRPE.

Acknowledgments

We would thank the anonymous reviewers for important suggestions that help us improve the quality of the manuscript. The first author would like to thank Professor Guangping He for useful suggestions. This work was partly supported by the National Natural Science Foundation of China (Nos. 61572532, 61876195, 61272058), the Natural Science Foundation of Guangdong Province of China (No. 2017B030311011), the Fundamental Research Funds for the Central Universities of China (No. 17lgjc24), and this work was also fund by FCT under contract IT (UID/EEA/50008/2019), PREDICT (PTDC/CCI-CIF/29877/2017), project QuantMining (POCI-01–0145-FEDER-031826).

References

[1] Srivasta Niraj, Müller Gerhard. Quantum images of hamiltonian chaos. Phys Lett A 1990;147(5):282–6.
[2] Kolobov MI. Quantum imaging. In: Conference on Lasers & Electro Optics & the Pacific Rim Conference on Lasers & Electro-optics; 2009.
[3] Yan Fei, Iliyasu Abdullah M, Venegas-Andraca Salvador E. A survey of quantum image representations. Quantum Inf Process 2016;15.
[4] Venegas-Andraca SE, Ball JL. Processing images in entangled quantum systems. Quantum Inf Process 2010;9(1):1–11.
[5] Yan Fei, Iliyasu Abdullah, Jiang Zhengang. Quantum computation-based image representation, processing operations and their applications. Entropy 2014;16(10):5290–338.
[6] le Phuc Q, Dong Fangyan, Hirota Kaoru. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. Quantum Inf Process 2011;10:63–84.
[7] Jiang Nan, Luo Wang, Wenya Wu. Quantum hilbert image scrambling. Int J Theor Phys 2014;53(7):2463–84.
[8] Chen Junxin, Zhu Zhiliang, Zhang Libo, Zhang Yushu, Yang Benqiang. Exploiting self-adaptive permutation-diffusion and dna random encoding for secure and efficient image encryption. Signal Process 2018;142:340–53.
[9] Haldun M. Ozaktas, M. Alper Kutay. The fractional fourier transform. In: 2001 European Control Conference; 2001, doi: 10.23919/ECC.2001.7076127.
[10] Gustafson Steven C. Book rvw: Introduction to fourier optics(second edition, by joseph w.goodman). Opt Eng 1995;35(4):1513.
[11] Mendlovic David, Zalevsky Zeev, Konforti Naim, Dorsch Rainer G, Lohmann Adolf W. Incoherent fractional fourier transform and its optical implementation. Appl Opt 1995;34(32):7615–20.
[12] Rodrigo Jos A, Alieva Tatiana, Calvo Mała L. Gyrator transform: properties and applications. Opt Express 2007;15(5):2190–203.
[13] Bracewell RN. Discrete hartley transform. J Opt Soc Am 1983;73(12):1832–5.
[14] Yang Yuguang, Jia Xin, Sun Sijia, Pan Qingxiang. Quantum cryptographic algorithm for color images using quantum fourier transform and double random-phase encoding. Inform Sci 2014;277(2):445–57.
[15] Song Xianhua, Wang Shen, Abd El-Latif Ahmed A, Niu Xiamu. Quantum image encryption based on restricted geometric and color transformations. Quantum Inf Process 2014;13(8):1765–87.
[16] Unnikrishnan G, Joseph J, Singh K. Optical encryption by double-random phase encoding in the fractional fourier domain. Opt Lett 2000;25(12):887–9.
[17] Zhang Shuqun, Karim Mohammad A. Color image encryption using double random phase encoding. Microwave Opt Technol Lett 2015;21(5):318–23.
[18] D'Ariano GM, Macchiavello C, Sacchi MF. On the general problem of quantum phase estimation. Phys Lett A 1998;248(2–4):103–8.
[19] Refregier Philippe, Javidi Bahram. Optical image encryption based on input plane and fourier plane random encoding. Opt Lett 1995;20(7):767.
[20] Naughton Thomas J, Hennelly Bryan M, Dowling Tom. Introducing secure modes of operation for optical encryption. J Opt Soc Am A Opt Image Sci Vision 2008;25(10):2608–17.
[21] Chen Wen, Javidi Bahram, Chen Xudong. Advances in optical security systems. Adv Opt Photon 2014;6(6):120–55.
[22] Zhou Nanrun, Tian Xianghua, Gong Lihua, Dong Jupei, Liao Qinghong. Quantum image encryption based on generalized arnold transform and double random-phase encoding. Quantum Inf Process 2015;14(4):1193–213.
[23] Frauel Yann, Castro Albertina, Naughton Thomas J, Javidi Bahram. Resistance of

the double random phase encryption against various attacks. Opt Express 2007;15(16):10253–65.

[24] Szczykulska Magdalena, Baumgratz Tillmann, Datta Animesh. Multi-parameter quantum metrology. Adv Phys: X 2016;1(4):621–39.

[25] Wang Shen, Song Xianhua, Niu Xiamu. A novel encryption algorithm for quantum images based on quantum wavelet transform and diffusion. Intell Data Anal Appl 2014;2:243–50.

[26] Yao Xiwei, Wang Hengyan, Liao Zeyang, et al. Quantum image processing and its application to edge detection: Theory and experiment. Phys Rev X 2017;7(3).

[27] Yi Lu, Kai Zhang, Gao YingHui. Qsobel: a novel quantum image edge extraction algorithm. Sci China (Inform Sci) 2015;58(1):1–13.

[28] Song Xianhua, Wang Shen, Ahmed A, El-Latif Abd, Niu Xiamu. Quantum image encryption based on restricted geometric and color transformations. Quantum Inf Process 2014;13(8):1765–87.

[29] Zhou Rigui, Hu Wenwen, Fan Ping, Luo Gaofeng. Quantum color image watermarking based on arnold transformation and lsb steganography. Int J Quantum Inform 2018;16(9):1850021.

[30] Naseri Mosayeb, Abdolmaleky Mona, Parandin Fariborz, et al. A new quantum gray-scale image encoding scheme. Commun Theor Phys 2018.

[31] Le Phuc Q, Iliyasu Abdullahi M, Dong Fangyan, Hirota Kaoru. Strategies for designing geometric transformations on quantum images. Theor Computer Sci 2011;412(15):1406–18.