Circuit-Based Modular Implementation of Quantum Ghost Imaging

FEI YAN\textsuperscript{1}, KEHAN CHEN\textsuperscript{1}, ABDULLAH M. ILIYASU\textsuperscript{2,3}, (Member, IEEE), AND KAORU HIROTA\textsuperscript{3,4}

\textsuperscript{1}School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
\textsuperscript{2}College of Engineering, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
\textsuperscript{3}School of Computing, Tokyo Institute of Technology, Yokohama 226-8502, Japan
\textsuperscript{4}School of Automation, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Abdullah M. Iliyasu (a.iliyasu@psau.edu.sa)

This study was sponsored by the Prince Sattam Bin Abdulaziz University, Saudi Arabia, via the Deanship for Scientific Research funding for the Advanced Computational Intelligence and Intelligent Systems Engineering (ACIISE) Research Group under Project 2019/01/9862.

ABSTRACT Although promising in terms of its applications in many facets of science and engineering; notably, in laser technology and remote sensing, ghost imaging is primarily impeded by its intense demands related to computational overhead, which impacts on the quality of output images. Advances in imaging and computing technologies have seen many efforts to overcome this perceived shortcoming. This study contributes towards ameliorating the earlier mentioned costs via implementation of ghost imaging from the perspective of quantum computing. Specifically, a quantum circuit implementation of ghost imaging is proposed wherein the speckle patterns and phase mask are encoded by utilizing the quantum representation of images. To accomplish this, we formulated several quantum modules, i.e. quantum accumulator, quantum multiplier, and quantum divider, and suffused them into our quantum ghost imaging (QGI) mechanism. Our study provides a new impetus to explore the implementation of ghost imaging using quantum computing resources.

INDEX TERMS Quantum computation, circuit implementation, quantum module, ghost imaging, quantum image processing, image encryption.

I. INTRODUCTION Leveraging on its immense potentials for applications that require minimal computing resources, speed, security, etc., quantum mechanics has exploded beyond its utility in optics (including laser technology [15] and remote sensing technology [29]) to exciting applications in computer science and engineering (e.g., machine learning [9] and artificial intelligence [28]). Naturally, this has also led to inter-disciplinary explorations, such as quantum ghost imaging (QGI) [4] and quantum image processing (QIP) [31], etc.

Ghost imaging is a technique employed to retrieve an object from the cross-correlation function of two separate beams, neither of which obtains information from the object [2]. One beam interrogates a target and then illuminates a single-pixel detector that provides no spatial resolution, while the other beam does not interact with the target, but it impinges on a high-resolution camera, hence yielding a multiple-pixel output [6]. The timeline for this sub-discipline’s development shows its modest beginning started in 1995, where the two beams of ghost imaging were formed from a stream of entangled photons [22]. The reconstruction of the image was attributed to the non-local quantum correlations between the photon pairs. For several years, ghost imaging was considered as an effect of quantum non-locality due to earlier experiments. Challenging this interpretation, Bennink et al. demonstrated ghost imaging using two classically correlated beams [3], following which, it was found that many of the features obtained with entangled photons could be reproduced with a classical pseudothermal light source. However, the nature of the spatial correlations exhibited with a pseudothermal source, and whether they can be interpreted as classical intensity correlations or are fundamentally non-local quantum correlations, is still debatable [8], [23], [24], [27]. Although, focusing on the problems and improvement of the spatial resolution, field of
view, and signal-to-noise ratio of the ghost imaging result, recent progress has revolved around using different types of light sources [26] and the implementations on different materials [30].

Even though ghost imaging has shown potential for applications demanding high detection sensitivity as well as high resolution, which are useful in the civil and military domains, development of the ghost imaging technique is constrained in terms of:

- Employment of large amounts of speckle patterns impacts on the quality of ghost imaging applications, whereas huge computational overhead is associated with the preparation and storage of these patterns.
- Interactions between speckle patterns and phase mask, which are integral components of ghost imaging, also impose additional computational demands that further stretch the overhead.
- Cross correlation between signals in the signal field and idler field which has further demand for computational resources that will further aggravate the overhead.

Quantum computing offers a fresh set of computing contraptions that utilize properties of parallelism and entanglement, etc. to speed up tasks, while demanding far less resources than their classical (i.e. digital of non-quantum) equivalent [5], [7], [25]. Among many other areas [12], these tools are used in the emerging sub-discipline of QIP [1]. Technically, QIP is focused on extending conventional image processing tasks and operations to the quantum computing framework [34]. While an introduction of our adapted quantum image model is presented in Section II, readers can refer to the growing literature in QIP, notably, the reviews in [11] and [35], for details on other pertinent advances in the QIP sub-discipline.

In this study, we attempt to utilize the proven potency of quantum information science; more specifically, we focus on aspects used in QIP, in a new paradigm for ghost imaging. The anticipated benefits accruing therefrom include:

- Use of a quantum register consisting of $n$ qubits capable of storing $2^n$ binary numbers provides exponential storage capability (in comparison with the classical register/storage), which provides a platform to circumvent the need for storage space for speckle patterns.
- Deployment of quantum computing operations to support decision regarding interactions between speckle patterns and phase mask provides a foothold to reduce the hitherto excessive computational complexity associated with ghost imaging.
- Furthermore, the cross correlation stage of ghost imaging would benefit from the parallelism inherent to quantum computing via the use of carefully crafted quantum arithmetic operations, i.e. quantum accumulator, quantum multiplier, and quantum divider.

The rest of the paper is organized as follows: in Section II, quantum image representation and the main quantum arithmetic operations of quantum adder and quantum comparator are introduced, following which, the quantum accumulator, quantum multiplier, and quantum divider are designed and proposed. In Section III, a complete QGI circuit network is designed, and with it, the rudiments of creating quantum speckle patterns, the interaction between the patterns as well as the quantum phase mask, and quantum computation of the cross correlation are established. In Section IV, the experiments heralding implementations of the realized quantum circuit are presented and analyzed in terms of the image quality. Finally, in Section V, we draw a few conclusions from the study and offer insights regarding its applications in quantum image encryption.

II. MODULAR APPROACH TO BASIC QUANTUM ARITHMETIC OPERATIONS

Traditionally, the arithmetic operations of addition, subtraction, multiplication, and division are employed to operate on two or more numbers. Considering their utility, it is important to extend classical execution of these operations to our QGI framework. In addition to the four traditional operations, we utilize the accumulator (ACC) and comparator (COM) operations as the six quantum arithmetic operations to support the execution of the proposed QGI protocol. First, we formalize, from established literature, the notion of an image as used on the quantum computing paradigm.

A. QUANTUM IMAGE REPRESENTATION

As defined in [34] and [11], QIP is devoted to “utilizing the quantum computing technologies to capture, manipulate, and recover quantum images in different formats and for different purposes.” The first step in accomplishing this requires a representation be conjured to encode images based on the quantum mechanical composition of any potential quantum computing hardware. Out of the available quantum image representations, in this study, the novel enhanced quantum representation (NEQR) of digital images [34] is utilized to represent both a speckle pattern and target image. This choice is attributed to its ability to support the use of the basis states of two qubit sequences for storing the color and position information about each pixel in the image [36]. Mathematically, the proposed model defines an image as:

$$|I(m, n, l)\rangle = \frac{1}{2^{m+n/2}} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} \bigotimes_{y=0}^{2^n-1} |c_{yx}^h\rangle |yx\rangle,$$  \hspace{1cm} (1)

where $|c_{yx}^h\rangle$ ($c_{yx}^h \in \{0, 1\}$) encodes the color information (whose range is $2^3$) of the pixel at position $|yx\rangle$, where $|yx\rangle = |y\rangle |x\rangle = |y_{m-1}y_{m-2} \ldots y_0\rangle |x_{n-1}x_{n-2} \ldots x_0\rangle$, $y_i, x_i \in \{0, 1\}$, $i = 0, 1, \ldots, m-1, j = 0, 1, \ldots, n-1$. An example of a $2 \times 2$ NEQR image and its quantum state is presented in [34], wherein, its preparation and retrieval procedures have been thoroughly discussed.

B. QUANTUM ADDER

The quantum addition operation is executed via the quantum adder circuit network (hereinafter called ADD module), which is considered a basic quantum arithmetic operation in
The quantum comparator (i.e. COM) circuit has been widely used in the quantum computing literature. Designed in [21] and as used in [33], the COM module in Fig. 2 compares two states producing two outputs $|y\rangle$ and $|y+x\rangle$. As presented in Fig. 1, a quantum adder consists of $2n−1$ “carry” and $2n$ “sum” sub-modules. Moreover, as discussed in [32], due to the fact that quantum gates are reversible, quantum subtraction could be implemented using a network of quantum adders. To show this, in the subtracter (i.e. SUB) module, a black bar is juxtaposed on the left side of the ADD module network.

C. QUANTUM COMPARATOR

The quantum comparator (i.e. COM) circuit has been widely used in the quantum computing literature. Designed in [21] and as used in [33], the COM module in Fig. 2 compares two states $|y\rangle$ and $|x\rangle$, where $|y\rangle = |y_{n-1}\ldots y_0\rangle$ and $|x\rangle = |x_{n-1}\ldots x_0\rangle$, $y_i, x_i \in \{0, 1\}$, $i = 0, 1, \ldots, n−1$. Qubits $|e_1\rangle$ and $|e_0\rangle$ are outputs of the comparison, which could take one of three outcomes:

- If $|e_1e_0\rangle = |10\rangle$, then $|y\rangle > |x\rangle$;
- If $|e_1e_0\rangle = |01\rangle$, then $|y\rangle < |x\rangle$;
- If $|e_1e_0\rangle = |00\rangle$, then $|y\rangle = |x\rangle$.

Therefore, when $|e_0\rangle = 0$, $|y\rangle \geq |x\rangle$; otherwise, $|y\rangle < |x\rangle$. Together with the discussion presented earlier in Section II-B, we conclude that the SUB module will work on $|y\rangle − |x\rangle$ only when $|e_0\rangle = 0$ (i.e. $|y\rangle \geq |x\rangle$).

D. QUANTUM ACCUMULATOR

Since the ghost imaging algorithm requires a series of pixel accumulation operations [6], we envision the need for a circuit network to accumulate contents of these pixels. Hence, in this subsection, we propose a rigorous paradigm for implementing the quantum accumulator (or simply ACC module).

![FIGURE 1. Circuit implementation of quantum ADD module (figure and descriptions adapted from [32]).](image1)

![FIGURE 2. Circuit implementation of quantum COM module (figure and descriptions adapted from [21]).](image2)

Hopefully, beyond its use here, the proposed ACC module will be useful for other protocols and applications in the quantum computing domain. Mathematically, the ACC module is designed to accomplish the following transformation:

$$\text{ACC}(0)|c_{y/x}\rangle|y\rangle|x\rangle = \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} |c_{y/x}\rangle|c_{y/x}\rangle|y\rangle|x\rangle.$$  (3)

As presented in Fig. 3, $|y\rangle$ and $|x\rangle$, i.e. $|y_{n-1}y_{n-2}\ldots y_0\rangle$ and $|x_{n-1}x_{n-2}\ldots x_0\rangle$, are $2n$ control qubits of the ADD modules, while $|c_{y/x}\rangle$ (which consists of $l$ qubits) retains different states in each pair of $|y_i,x_i\rangle$. The state $|y_i,x_i\rangle$ ranges from $|0\rangle\otimes|2n\rangle$ to $|1\rangle\otimes|2n\rangle$, i.e. from $|0\rangle$ to $|2^{2n}−1\rangle$. The $2^{2n}$ ADD modules are utilized to perform the summation of $|c_{y/x}\rangle$ in each state of $|y_i,x_i\rangle$. The additional $l$ qubits (which are initialized as states $|0\rangle^{\otimes l}$) are integrated into the circuit to record the accumulation outcome of each ADD module, and the combination of these results produce the final output of the whole operation.
E. QUANTUM MULTIPLIER

Quantum multiplier (or simply MUL module as we shall refer to it henceforth) is primarily targeted at executing the multiplication operation between two quantum states. Some often-used MUL operations include those in [18] and [16]. In this subsection, although formulated for widespread use, the MUL module is purposely refined to support efficient implementation of our proposed QGI protocol. As a premise, we start with the classical multiplication of two binary numbers \( y = y_{m-1} \ldots y_1 y_0 \) and \( x = x_{n-1} \ldots x_1 x_0 \), a process outlined in (4):

\[
xy = (y_{m-1} \ldots y_1 y_0)x_0 + (y_{m-1} \ldots y_1 y_0 0)x_1 + \ldots + (y_{m-1} \ldots y_1 y_0 0 \ldots 0)x_{n-1}
\]

\[
= yx_0 + (y0)x_1 + \ldots + (y0 \ldots 0)x_{n-1}. \quad (4)
\]

To elucidate, let \( y = 10101 \) (i.e. \( n=5 \)) and \( x = 1011 \) (i.e. \( x_3 = 1, x_2 = 0, x_1 = 1, x_0 = 1, \) and \( m=4 \)), then \( yx = (10101) \times (1011) = 11100111 \). The execution of this operation can be better comprehended via the stepwise implementation presented in (5):

\[
10101 \times 1011 = \sum_{i=0}^{3}(2^i y)(x_i)
\]

\[
= (2^0 y)(x_0) + (2^1 y)(x_1) + (2^2 y)(x_2) + (2^3 y)(x_3)
\]

\[
= (10101 0 ) + (10101 00 ) + (10101 000 ) + (10101 0000 )
\]

\[
= 10101 + 101010 + 1010100 + 10101000 = 11100111. \quad (5)
\]

Employing the ADD module (presented earlier in Section II-B) and (4), our proposed MUL operation is executed using (6):

\[
MUL|0\rangle|z\rangle|y\rangle|x\rangle = |yx\rangle|y0\ldots0\rangle|y\rangle|x\rangle,
\]

where a sequence of \(|0\rangle\)'s is used as input to record the product of \(|y\rangle\) and \(|x\rangle\), while \(|z\rangle\) (including \(m+n-1\) qubits) is used to store the temporary results in the multiplication process. As shown in the circuit network in Fig. 4, our MUL module is implemented via a concatenation of \(m\) ADD operations. The procedure for executing the MUL circuit that multiplies two binary numbers is outlined in the sequel.

(i) Input: Besides the input states \(|y\rangle\) and \(|x\rangle\), additional qubits of \(|z\rangle\) and \(|p\rangle\) (which consists of \(m+n-1\) and \(n\) qubits, respectively) are initialized as a sequence of \(|0\rangle\)'s, wherein \(|p\rangle\) is used to dynamically store the intermediate result of the multiplication in each step, while \(|z\rangle\) is used to prepare inputs of the ADD module as explained in (ii).
(ii) Iterative addition: During the \((i+1)\)th step, \(n\) Toffoli gates (controlled by the \(|y\rangle\) and \(|x_i\rangle\) for \(i = 0, 1, \ldots, m - 1\)) are applied on the state of \(|z\rangle\) to obtain an output \(|z'\rangle = |y\rangle|0\rangle^\otimes m|x_i\rangle\) which is regarded as an input to the ADD module (i.e. ADD\(_{i+1}\)) in this step, while the other input of ADD\(_{i+1}\) comes from the addition result of ADD\(_i\) (We note that it is initialized as \(|0\rangle^\otimes m\) when \(i=0\)).

For instance, in step 1 (i.e. \(i=0\)), if \(|x_0\rangle = |1\rangle\), we set \(|z_{n-1}\cdots z_0\rangle = |y_{n-1}\cdots y_1y_0\rangle\), so \(|z\rangle = |y\rangle\) in this case. Then, \(|z\rangle\) and \(|p\rangle\) \((p=0)\) are considered as two inputs of the ADD\(_1\). Following the addition operation, \(n\) Toffoli gates are employed to reset \(|z\rangle\) to its original states, i.e. \(|0\rangle^\otimes m+n-1\).

(iii) Output: An iterative approach is used to compute the product between \(|y\rangle\) and \(|x\rangle\) (including \(m+n\) qubits) such that \(|p\rangle = |p_{m+n-1}\cdots p_1p_0\rangle = |y\rangle|x_0\rangle + |y\rangle|0\rangle|x_1\rangle + \ldots + |y\rangle|0\rangle^{\otimes m-2}|x_{m-2}\rangle + |y\rangle|0\rangle^{\otimes m-1}|x_{m-1}\rangle\).

**F. Quantum Divider**

On classical computers, the binary division operation is expounded as a series of subtraction tasks. Consider, as an example, the classical division operation \(\frac{100110}{110}\) (as presented in Fig. 5), which can be executed via the following four steps:

Step 1: A subset comprising of the first three bits in the numerator (i.e. the dividend) “100” is taken as minuend, which is compared with the denominator (i.e. the divisor) “110” using the COM module (as presented in Section II-C). Since 100 < 110, the subtraction operation is not applicable. Therefore, the result from Step 1 becomes “100” with the next bit in the numerator (in this case “1”) added to make a sequence “1001” that serves as the new minuend sequence.

Step 2: The two binary sequences being compared are the outcome from Step 1 (i.e. “1001”) and the denominator (i.e. “110”). Since 110 < 1001, the SUB module (as presented in Section II-B) will be applied to perform the subtraction. The result of this subtraction is “11”, to which the next bit in the numerator sequence (i.e. “1” would be included to make the new minuend sequence, i.e. “111”).

Step 3: Similar to the previous steps, we use the COM module to compare two states. However, in this case, the minuend resulting from Step 2 (i.e. “111”) is compared with the divisor (i.e. “110”). Since 111 > 110, we proceed with the subtraction where 111 − 110 = 01. Finally, the last bit of the numerator (i.e. “0”) is juxtaposed with the outcome of this subtraction to make the bit string “010”, which serves as the minuend for the next step of the operation.

Step 4: Continuing here, we compare the minuend “010” with the divisor “110”, and since 010 < 110, the subtraction operation is not activated. Having exhausted the bits in the numerator sequence, our operation returns the last two bits of the minuend “10” as the reminder of the division operation. In the event of the opposite scenario, i.e. the minuend is greater than or equal to “110”, the SUB operation is used to obtain the difference that serves as input to subsequent steps of the division operation.

As outlined in the four steps above, the division \(\frac{100110}{110}\) produces a quotient “110” and a remainder of “10”. Figure 6 presents a pictorial implementation of the four steps outlined earlier. Based on them, we propose a quantum divider (or simply DIV module) to implement the division operation of our QGI applications.

An overview of the composition, formulation, and circuitry to implement the quantum DIV module that executes the division operation is presented forthwith. Consider a sequence \(|y\rangle = |y_{m-1}\cdots y_1y_0\rangle\) as the dividend (numerator) and another one \(|x\rangle = |x_{n-1}\cdots x_1x_0\rangle\) as the divisor (denominator) of a division operation. Then, using a depository, additional information \(|x'\rangle = |x_{m-1}'\cdots x_1'x_0'\rangle\) emanating from the stepwise execution of the subtraction operation (itself part of the quantum DIV operation) to divide \(|y\rangle\) by \(|x\rangle\), the result of which (i.e. the quotient) is returned as \(|q\rangle = |q_{m-n+1}\cdots q_1q_0\rangle\).

As presented in Step 1 of the DIV circuit (in Fig. 6), the first \(n\) CNOT gates are used to map the state of \(|x_{n-1}\cdots x_1x_0\rangle\) to \(|x_{m-1}'\cdots x_1'x_0'\rangle\). Subsequently, the COM module (presented earlier in Section II-C) is utilized to compare between states \(|y_{m-1}\cdots y_1y_0\rangle\) and \(|x_{m-1}'\cdots x_1'x_0'\rangle\). A useful state coming out of the COM module is \(|e_0\rangle\), whence a result \(|e_0\rangle = |0\rangle\) (i.e. \(|y\rangle \geq |x'\rangle\)) activates the \(e_0\)-controlled SUB module to obtain the subtraction outcome of \(|v_1\rangle = |y\rangle|x'\rangle\) (as the input of the COM module in Step 2). Otherwise, the SUB module is inactive, so the outcome remains \(|y\rangle\) (refer to full illustration at the bottom of Fig. 6). Following the SUB module, the first \(e_0\)-controlled CNOT gate is applied on \(|q_{m-n}\rangle\) to obtain the first (also leftmost) or most significant qubit of the division result (i.e. quotient), while the second CNOT gate ensures \(|e_0\rangle = |0\rangle\) before proceeding to Step 2. At the end of the Step 1, additional \(n\) CNOT gates are similarly used to reset state \(|x'\rangle\) to its initialized state, i.e. a sequence of \(|0\rangle^\otimes m\) entries, preparatory for its use in Step 2.

In Step 2, the first \(n\) CNOT gates assign the value \(|0\rangle^\otimes m-1\rangle\) to qubit \(|x'\rangle\) for comparison with the output from Step 1 (i.e. \(|v_1\rangle\)). Similarly, \(|e_0\rangle = |0\rangle\) indicates
Figure 6. Circuit implementation of quantum DIV module.

\[ |v_1 \rangle \geq |x' \rangle \], so the subtraction operation can be executed to produce an outcome \( |v_2 \rangle = |v_1 \rangle - |x' \rangle \). Following that, the next \( (n+2) \) CNOT gates are applied to: (1) obtain the second qubit of the division result, (2) guarantee \( |e_0 \rangle = |0 \rangle \) in the next step, and (3) reset \( |x' \rangle \) from \( |0x \rangle |0 \rangle ^{m-n-1} \) to \( |0 \rangle ^m \).

The final outcome of the DIV module (indicated as \( |q \rangle \) at the end of the circuit) is the iterative execution of the steps enumerated above. Meanwhile, the sequence \( |v_{m-n+1} \rangle = |v_{m-n} \rangle - |x' \rangle \) (technically, composed of \( |r_{n-1}r_{n-2} \ldots r_0 \rangle \) at the end of the circuit) is regarded as the remainder from the DIV operation.

Figure 7. Mathematical analysis of the setups of ghost imaging experiment.

III. CIRCUIT-BASED IMPLEMENTATION OF QGI

A. MATHEMATICAL FORMULATION OF GHOST IMAGING

Following the mathematical discussions in [2], the methodological outline of our proposed QGI experiment is described in Fig. 7. There are two speckle pattern sequences, i.e. \( I_{g_{k-1}} \ldots I_{g_0} \) and \( I_{d_{k-1}} \ldots I_{d_0} \), wherein, \( I_{k}^s \) and \( I_{k}^d \) indicate the speckle patterns in the signal and idler fields, respectively. The speckle patterns at the same position of two sequences are identical and each pattern consists of \( 2^m \times 2^n \) pixels. \( U_k (y, x) \) represents a pixel whose coordinate is \( (y, x) \) in the region of \( U \) (which is the field of view) at the \( k \)th sample.
pattern in the sequence (e.g., \(R^k\) in the signal field), where \(y \in \{0, 1, \ldots, 2^m - 1\}\), \(x \in \{0, 1, \ldots, 2^n - 1\}\), and \(k \in \{0, 1, \ldots, 2^s - 1\}\). We define a subregion \(A\) (where \(A \subseteq U\)) within which the object is located, i.e. the pixels in \(A\) are subsets of pixels in \(U\). Furthermore, we define a characteristic function \(F_k(y, x)\) in the region \(U\) such that \(\cup_k F_k(y, x) = \text{in the sub-region} A\), then \(F_k(y, x) = 1\); otherwise \(F_k(y, x) = 0\).

We further define a special vector \(W(k)\) whose \(k\)th element records the statistical weight for the \(k\)th speckle pattern in the sequence (of the signal field). Formally, \(W(k)\) could be calculated as:

\[
W(k) = \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^m-1} \mathcal{F}_k(y, x) \cup_k (y, x),
\]

where \(W(k)\) calculates the sum of pixels belonging to \(A\) in the \(k\)th speckle pattern that is the weight of the \(k\)th speckle pattern in the signal field. Whereas \(P\) (in Fig. 7) indicates the sequence of speckle patterns that are obtained from the idler field, “\(\otimes\)” performs cross-correlation operation between the speckle patterns \(P\) and their statistical weights \(W\) in the signal field, after which, the ghost imaging result \(R\) is obtained in the form formulated in (8):

\[
R = \langle W(P) - (W)(P) \rangle,
\]

where \(\langle \cdot \rangle = \frac{1}{2^m} \sum_k \cdot \) denotes an ensemble average over \(2^m - 1\) phase realizations. Meanwhile, every pixel in \(R\) (e.g., at coordinate \((y, x)\)) could be represented as:

\[
R(y, x) = \frac{1}{2^m} \sum_{k=0}^{2^s-1} W_k \cdot \mathcal{F}_k(y, x) - \frac{1}{2^m} \sum_{k=0}^{2^s-1} W_k \cdot \sum_{k=0}^{2^s-1} \mathcal{F}_k(y, x).
\]

The remainder of this section dwells on the quantum circuit implementation of ghost imaging based on the formulations and discussions presented in (7)-(9) as well as in earlier sections of the study.

**B. GENERATION OF QUANTUM SPECKLE PATTERNS**

In the context of our proposed QGI, we define a speckle pattern in the format presented in (1), where, for \(l=1\), the color information of every pixel in the speckle pattern consists of only two states, i.e. 0 (black) or 1 (white). It is randomly generated by utilizing the Hadamard gate, i.e. \(H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\), which transforms the initial state \(|0\rangle\) to \(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\). While the whole procedure is outlined in the circuit in Fig. 8, it is noteworthy that, whereas on classical computers, the randomness of the gray value is generated with a certain periodicity (i.e. it is pseudo-random), by utilizing the quantum gates (Hadamard and CNOT gates), truly random speckle patterns can be generated on the quantum computing paradigm.

Moreover, as presented in Fig. 8, all initial states \(|0\rangle\) are transferred into the outcome of the speckle pattern, which consists of \(|s\rangle = |s_{m-1}s_{m-2}\ldots s_0\rangle = |0\rangle \otimes |y\rangle = |y_{m-1}y_{m-2}\ldots y_0\rangle = |0\rangle \otimes |x\rangle = |x_{n-1}x_{n-2}\ldots x_0\rangle = |0\rangle \otimes |\mathcal{C}_{xy}\rangle\), where \(|\mathcal{C}_{xy}\rangle\) is the final state (of the signal field). Formally, \(|\mathcal{C}_{xy}\rangle\) could be represented as:

\[
\mathcal{C} = \frac{1}{\sqrt{2}} \sum_{y=0}^{2^m-1} |y\rangle \otimes \sum_{x=0}^{2^n-1} |x\rangle \otimes \sum_{s=0}^{2^s-1} |s\rangle.
\]
the final speckle pattern comprising of states \(|y\rangle, |x\rangle, |s\rangle,\) and \(|c_{xy}\rangle\) as presented in Fig. 8.

C. QUANTUM CIRCUIT FOR GHOST IMAGING EXPERIMENT

Ghost imaging is a technique that is focused on producing the image of an object by combining information from two sources. In photonic quantum computing, implementation of ghost imaging involves the use of a source pair of entangled photons and each pair is shared between the two detectors. Relying on the discussions and outcomes in Section III-A, our QIP-based circuit model for implementing QGI could be construed with reference to (8). Consequently, we tailor our QGI in terms of quantum computing resources to compute the parameters \((\langle WP \rangle, \langle W \rangle, \langle P \rangle, \) and \(\langle WP \rangle - \langle W \rangle\langle P \rangle\) in (8).

1) CALCULATION OF \(\langle WP \rangle\) AND ITS CIRCUIT IMPLEMENTATION

In quantum information science, a circuit is an effective and pictorial description of the quantum state evolution from inputs to its outputs. The calculation of \(\langle WP \rangle\) is detailed in the remainder of this subsection, while we outline the execution of the five stages of the quantum circuit implementation of the QGI as presented in Fig. 9.

(i) Interaction between the speckle patterns and phase mask:

As delineated in the blue rectangle in Fig. 9, this unit specifies the interaction between the speckle patterns and the phase mask as required to obtain the quantum desired (or target) image (i.e. \(|I_{xy}\rangle\)). The qubit sequence \(|y_{m-1} ... y_{1} y_{0} x_{n-1} ... x_{1} x_{0} c_{xy}^{m} \rangle\) at the top of this unit (in Fig. 9) represents the quantum phase mask, wherein \(|y\rangle\) and \(|x\rangle\) are the coordinates, while \(|c_{xy}^{m}\rangle\) records the gray value of the pixel in the quantum mask image. The extra qubits, i.e. \(|y\rangle, |x\rangle, |s\rangle,\) and \(|c_{xy}^{s}\rangle\), represent a series of information about the speckle patterns, where \(|s\rangle\) indicates the number of the patterns, and \(|y\rangle\) as well as \(|x\rangle\) is the pixel coordinates of the pattern, while \(|c_{xy}^{s}\rangle\) represents the gray value of pixels in these speckle patterns. In addition, \(|c_{xy}^{i}\rangle\) records the gray value of the patterns across the mask image, i.e. the generated quantum desired image.

To realize the needed interaction between the quantum mask image and the speckle pattern, as discussed in Section III-A, a position-wise correspondence comparison between the pixels at the same position of the quantum mask image and each quantum speckle pattern is required. To do this, we have to compare the gray values of the two pixels (i.e. \(|c_{xy}^{m}\rangle\) and \(|c_{xy}^{s}\rangle\)), if both of them are in state \(|1\rangle\), the gray value of the pixel in the desired image (i.e. \(|c_{xy}^{i}\rangle\)) is transformed from its initialized state \(|0\rangle\) to \(|1\rangle\) by using a CNOT gate (via multiple control conditions). Otherwise, the gray value of the pixel in the desired image remains \(|0\rangle\) (i.e. black). In accomplishing the aforesaid comparison, it is imperative to ensure that the control conditions imposed on the \(|y\rangle\) and \(|x\rangle\) coordinates of the quantum mask image are identical with those in the quantum speckle pattern. This ensures that the operation is executed at the same position.

(ii) Calculation of the weight \(|w_{s}\rangle\) in the signal field:

The second unit of our circuit (delineated using a yellow legend) accumulates the weights of all pixels of each quantum desired image. The quantum ACC module...
presented in Section II-D is used to undertake this task. The qubit sequences \(|y⟩, |x⟩\), and \(|c^y_{sx}\rangle\) are regarded as three inputs of the ACC module, while \(|w_{sx}\rangle = \sum_{y=0}^{2^{m-1}} \sum_{x=0}^{2^{n-1}} \langle c^y_{sx}\rangle\) is the output to store the weight value of every desired quantum image. The control qubits applied on the ACC module in Fig. 9 guarantee that the pixel accumulation is confined within the same desired image, i.e. the weight \(|w_{sx}\rangle\). After the accumulation operation, the output of this part is \(|s^m w_{sx}\rangle\) which contains all the weights of each quantum desired image.

(iii) Calculation of the product of weights and speckle patterns:
The third unit of our circuit (delineated in green) is used to compute the product of weights \(|w_{sx}\rangle\) in the signal field and the sequence of speckle patterns in the idler filed \((|I⟩_d|)\) which are not interacted with the mask. The quantum MUL module presented in Section II-E is used to perform this task. The two inputs of the MUL module are the weights of the signal field \(|w_{sx}\rangle\) and \(|c^y_{sx}\rangle\) (i.e. the color information of the speckle pattern \(|I⟩_d|)\). Meanwhile, the control conditions on \(|s^m\rangle\) and \(|s⟩\) confine the operation to the same pairs in the multiplication process (i.e. each \(|w_{sx}\rangle\) corresponding to each \(|I⟩_d|\) in the sequence). The outcome is stored in \(|c^y_{sx}\rangle = |c^y_{sx}\rangle |w_{sx}\rangle\) and used as the input of the next unit.

(iv) Accumulation of the corresponding pixels in the speckle pattern sequence:
At the top of the unit highlighted in purple color (Fig. 9), a quantum ACC module is used. The inputs of this ACC module are \(|c^y_{sx}\rangle\) and \(|s⟩\), while the output is stored in \(|c^y_{sx}\rangle\). The purpose of this unit is to simultaneously accumulate the gray values of the pixels in the same position of each pattern (in the sequence). The control conditions on \(|y⟩\) and \(|x⟩\) ensure that the ACC module is restricted to the same position-wise concurrence in each pattern.

(v) Calculation of the final result \(|w_{sx}p_s⟩\):
Upon \(2^{m+n}\) accumulations of gray values from the last unit, the color information of the speckle patterns in the idler field, i.e. \(|c^y_{sx}\rangle\), has been transformed to \(|w_{sx}p_s⟩\). The main task of this unit is to change the state \(|w_{sx}p_s⟩\) to \(|w_{sx}p_s⟩\), i.e. the ensemble average over \(|s⟩\) phase realizations of \(|w_{sx}p_s⟩\). Using the DIV module proposed in Section II-F, the state \(|w_{sx}p_s⟩\) could be divided by \(|s⟩\) to obtain the outcome, i.e. \(|w_{sx}p_s⟩\).

2) CALCULATION OF \(|⟨P⟩|\) AND \(|⟨W⟩|\) AND THEIR CIRCUIT IMPLEMENTATIONS
The circuit to calculate \(|⟨P⟩|\) is shown in Fig. 10(a), where \(|yx c^y_{sx}\rangle\) is initialized as \(\sum_{y=0}^{2^{m-1}} |y⟩ \otimes \sum_{x=0}^{2^{n-1}} |x⟩ \otimes |0⟩\), while \(|c^y_{sx}yx⟩\) represents the quantum speckle patterns which is created by the circuit in Fig. 8 and generally used as shown in Fig. 9. The concurring color information of each pixel \((|c^y_{sx}\rangle\) at the same position in the 2\(^n\) speckle patterns are accumulated using ACC module and its result would be stored as state \(|p_s⟩\). Meanwhile, the control qubits combinations guarantee concordance in terms of the content of these patterns. In addition, \(|p_s⟩\) is the input of DIV module to calculate the final result, i.e. \(|⟨P⟩| = |p_s⟩/|s⟩\).

Figure 10(b) outlines the circuit to execute calculation of \(|⟨W⟩|\) as well as notes to aid thereof. The interaction between the speckle patterns and the quantum mask image, as well as the calculation of \(|w_s⟩\) utilizing \(2^m\) ACC modules are accomplished in steps similar to those in Fig. 9 delineated using blue and yellow legends. In addition, since \(|w_s⟩\) is an entangled state of \(2^m\) states corresponding to each state of \(|s⟩\), the ACC module in the last unit (highlighted in red rectangle) is used to aggregate all the \(|w_s⟩\) states. Finally, the accumulation result is divided by \(|s⟩\) to compute the final output, i.e. \(|⟨W⟩|\).

3) CREATION OF THE GHOST IMAGE BY USING \(|⟨R⟩| = |⟨WP⟩| − |⟨W⟩||P⟩\|
By concatenating the various sub-circuits presented in this section, we realize the circuit to implement the ghost imaging technique which also translates to the operation \(|⟨R⟩| = |⟨WP⟩| − |⟨W⟩||P⟩\|\) (that is \(|⟨r⟩| = |⟨wp⟩| − |⟨w⟩||p⟩\|\)) as presented in Fig. 12. It is trivial that \(|⟨w⟩|\) and \(|⟨p⟩|\) are the two inputs of the MUL module and the process for their computation has already been discussed in Section III-C.1. The SUB module executes the subtraction operation, but before it, a COM module (whose output is \(|e_0⟩|\) is set in Fig. 12. As presented earlier in Section II-C, \(|e_0⟩|\) = \(1\) indicates the subtrahend is larger than the minuend, i.e. \(|⟨w⟩⟩|p⟩\| > |⟨w⟩⟩|p⟩\|\), so the SET-0 operation is triggered. The SET-0 operation sets all input qubits to \(0\) state by using a sequence of CNOT gates (as used in [14]). In such instances, \(|e_0⟩|\) becomes a control qubit of the ADD module. It is important to clarify that the ADD module is activated only when \(|e_0⟩|\) = \(0\). After \(2^{m+n}\) comparison, SET-0, and subtraction operations, the final result \(|⟨r⟩| = |⟨wp⟩| − |⟨w⟩||p⟩\|\) is obtained. The state \(|yx ⟨r⟩⟩\) is akin to the quantum representation of ghost image.

D. QUANTUM CIRCUIT COMPLEXITY ANALYSIS
In quantum computation, complexity theory has been studied to analyze transformations in terms of unitary operations that oversee the evaluation of the circuit. Therefore, the complexity of quantum algorithms are usually computed in terms of quantum basic gates. However, circuit complexity depends largely on the strategy employed for circuit decomposition and the designated basic operations. Consequently, in this study, we will confine our discussion on complexity evaluation to the number of CNOT gates, since it is considered as the relatively most “expensive” elementary gate and it can be easily utilized to simulate the other complicated gates. For simplicity, the complexity evaluation for our proposed QGI operations rely on the approach in Fig. 11, where seemingly complicated circuits (on the left) are decomposed into circuit networks composed entirely of basic or elementary quantum gates (on the right), i.e. NOT, CNOT, and Toffoli gates.
Furthermore, as indicated in Fig. 11(c), an \( n \)-controlled NOT gate can be decomposed into \( 2(n - 1) \) Toffoli gates and 1 CNOT gate. Meanwhile, each Toffoli gate can be further simulated using 6 CNOT gates.

Consequently, for our QGI application, a module-wise computation of complexity is adopted since subsequent combination of the complexity of the modules would add up to the complexity of the QGI circuit network. In that regard, as seen in Fig. 1, the ADD module consists of \( 2n - 1 \) “carry” units, \( n \) “sum” units, and 1 CNOT gate. Further, the “carry” unit could be decomposed into 2 Toffoli gates and 1 CNOT gate, while the “sum” unit could be executed using 2 CNOT gates as presented in Fig. 1(a) and (b). Therefore, to facilitate the ADD module, we need \( 28n - 12 \) basic CNOT gates. Similarly, the implementation of COM module in Fig. 2 requires \( 24n^2 + 6n \) CNOT gates, where \( 2n \) denotes the length of the bit strings for the inputs.

As argued in Section II, by referring to the ADD and COM modules as elementary arithmetic operations, the ACC, MUL, and DIV modules are realized as advanced quantum arithmetic operations for the implementation of QGI. To implement the ACC module (in Fig. 3), \( 2^{2n} \) ADD operations (each includes \( 4l - 2 \) Toffoli gates and \( 4l \) CNOT gates) are employed and each is equipped with \( 2n \) control conditions. Therefore, the computational complexity of this ACC module can be simulated as \( O(l \cdot 2^{2n}) \), where \( l + 2n \) is the length of the bit strings for the inputs. Similarly, the computational complexities of MUL (in Fig. 4) and DIV (in Fig. 6) modules can be simulated as \( O(mn) \) and \( O(m^2n) \), respectively, where \( m + n \) is the length of the bit strings for the inputs.

Finally, since the realization of our QGI entails combining the three (i.e. ACC, MUL, and DIV) modules in Figs. 9, 10, and 12, then the combined complexity of the QGI operation is calculated as \( O(\sigma \cdot 2^\sigma) \), \( \sigma = m + n + t \), where \( m \) and \( n \) are the number of qubits to encode the speckle pattern image and \( t \) is number of the qubits to encode the number of speckle patterns. In comparison with the complexity of quantum image preparation in [34], the complexity to facilitate the procedure can be considered “cheaper” and, as such, acceptable.
IV. EXPERIMENTS AND ANALYSIS OF QGI

Based on the discussions in earlier sections, two sets of simulation-based experiments are executed to demonstrate both the utility and reliability of our proposed QGI protocol. In the first set of experiments, four 128 × 128 sized mask sets (each a quantum image): (i) 13 uppercase English alphabets, (ii) all 26 A to Z uppercase English alphabets, (iii) all 26 A to Z uppercase and additional 13 lowercase English alphabets, and (iv) all 26 A to Z uppercase and all 26 A to Z lowercase alphabets, are prepared. The images in the rightmost column of Fig. 13 are their original versions, i.e. the mask image of the QGI protocol.

Technically, the QGI protocol executes three tasks: choose the mask image, generate the speckle patterns, and determine the number of such speckle patterns. Since approaches guiding choice of mask image and that for generation of speckle patterns have been discussed in Section III-B, for the remainder of our experimental implementation, we focus on determining the number of speckle patterns, which is known to impact the quality of the ghost image. Therefore, in this simplified execution of the QGI protocol, for each image, 8 experimental results (i.e. resulting images) are obtained according to the different number of speckle patterns generated. To further assess this trend, larger, i.e. 256 × 256, versions of the same mask images are used and similar results are presented in Fig. 13(b).

Considering the method employed to encode the speckle patterns and target image in this study, as well as future implementation in quantum computing systems [13], the Normalized Cross Correlation (NCC) metric is used to evaluate the quality of the obtained images. Figure 14(a) presents the NCC values of four (256 × 256) images with varying number of alphabets (letters). The outcomes suggest that with increase in number of speckle patterns, the quality of the resulting images is improved (i.e. the NCC values increases). In addition, these results indicate that the NCC value of the image for mask set (iv) is greater than that for mask set (iii) and so on (i.e. decreasing with the number of letters in the mask set). Fig. 14(b) presents a summary of the relationship between the NCC values and resolution of speckle image (for mask set (iv)). Conversely, the NCC value is found to be inversely proportional to the resolution of the speckle image. The curves show the variation in the resolution and NCC values for different number of speckle patterns. As seen in this plot, the NCC values decreased with increase in the resolution. This trend applies to all instances (e.g., 500, 1000, and 2000) of number of speckle patterns. Furthermore, Fig. 14(c) shows the relationship between the number of alphabets (i.e. the mask sets) and the adopted quality measure (i.e. the NCC values), in this case for 1000 speckle patterns. As seen in the two curves in this figure, the NCC values increase with decrease in the resolution of the image (from 128 resolution (in black) to 256 resolution (in green)). Looking at the curves point-wise, it can be seen that the lower resolution images (i.e. 128 resolution) had better NCC values. This could be interpreted to suggest that the smaller resolution images produce better fidelity in terms of resulting ghost image irrespective of the number of speckle patterns. Finally, Fig. 14(d) shows the 3-dimensional relationship between the number of alphabets, NCC values, and number of speckle patterns for both resolutions (128 and 256) reported in the QGI experiments. As seen from the two curves, the outcomes for the lower resolution (128) images outperforms the higher...
resolution in terms of NCC values as number of speckle patterns increases.

The results reported in this section support earlier claims that many of the properties of quantum computing that are exploited (notably, parallelism and superposition) provide our proposed QGI protocol with the potency to accelerate sorting large size of images and mask image with high demands for speckle patterns. Furthermore, exploiting the randomness associated with the use of quantum gates to generate speckle patterns, high quality ghost images can be realized.

V. CONCLUSIONS AND DISCUSSIONS

A. DISCUSSION ABOUT QGI-BASED QUANTUM IMAGE ENCRYPTION

Encouraged by the framework and outcomes realized from the use of our proposed QGI protocol, we envision the promising applications in quantum image encryption. In this section, we surmise on such implementation which is contingent on the inherent property of ghost imaging that neither of the two beams in the signal and idler fields carries the information from the object. It behooves that the proposed protocol may offer an outstanding performance for the quantum image encryption. Inspired by similar use in classical computing, our explorative extension of the proposed QGI protocol is primarily aimed at protecting the integrity of multimedia (in this case quantum images) and enhancing its security during transmission.

Employing quantum computing resources to scramble contents of an image into incomprehensible formats have been demonstrated to enhance security during transmission [10], [17], [19], [37]. Hence, we contend that our proposed QGI protocol could be employed for similar purposes. Specifically, we adduce that by utilizing speckle patterns, weights, cross-correlation, etc as presented in earlier parts of this study, a quantum image can be encrypted prior to transmission and recovered at the receiver’s end of the transmission line. The speculative sketch of this implementation of quantum image encryption is presented in the steps below.

(i) The encryption of a quantum image based on QGI protocol:
The circuit network presented earlier in Fig. 8 forms the core of the speckle patterns generation step. Further, the interactions between the speckle patterns and the quantum mask image are accomplished by utilizing the blue region of the circuit in Fig. 9, and by implementing the yellow region therein, the series of corresponding weights are obtained. In this case, the quantum image being encrypted for transmission can be similarly decomposed into two parts, i.e. speckle patterns and weights, and, as such separately, they are indistinguishable via computer or human inspection.

(ii) The decryption of a quantum image based on QGI protocol:
When the receiver obtains the two sets of information (i.e. speckle patterns and corresponding weights), the cross-correlation operation is utilized to “decrypt” the quantum images. Again, referring to Fig. 9, the last green, purple, and red regions demonstrate the execution of the cross-correlation operation whose inputs are the speckle patterns and weights that were obtained during the encryption procedure. After this decryption process, the real information that the sender intends to transmit could be recovered.

The speckle patterns, weights, and the image being encrypted are all encoded as quantum states. As enumerated in the sequel, such states have some properties that could be exploited to enhance the security of the encryption protocols.

(i) On intercepting the encrypted information, i.e. the weights or speckle patterns, an unauthorized user would resort to quantum measurement operations to extract the specific information. However, as is inherent to quantum systems, such destructive measurements lead to loss of quantum coherence and, with it, a collapse of the quantum superposition state to a basis state, a process that is irreversible. This itself could serve as a signal that a channel is unsafe. Moreover, the “collapsed” state that is so realized is insufficient to meaningfully reconstruct the transmitted message.

(ii) Furthermore, the fact that the recovery of an “encrypted” image requires a large number of cross-correlation operations between the large number of speckle patterns and the corresponding weights, provides additional guarantee regarding the security of encrypted images from illicit use.

(iii) The corresponding weights (regarded as “quantum information key” in the encryption and decryption procedures) could be interpreted as superposed quantum
states. Therefore, further scrambling this key could help to enhance the security of the encryption algorithm. Additionally, the no-cloning theorem (NCT) [20] imposes the restriction that quantum states cannot be copied, a property that further safeguards access to the transmitted message by unauthorized users.

B. CONCLUSIONS
In this study, we proposed a method to implement the ghost imaging experiment using quantum circuits. To achieve this, a corteg of circuit networks to execute the accumulation, multiplication, and division operations were proposed. The contributions from this study mainly include: (1) utilizing the quantum superposition, a quantum register to enhance the capacity to store quantum speckle patterns was broached. In addition, by employing the quantum gates (i.e., Hadamard and CNOT gates), a truly random speckle pattern sequence was generated, from which the ghost imaging quality was improved via adjustments to the sampling rate of the speckle pattern. (2) Utilizing quantum parallelism, some universal quantum arithmetic modules (such as ACC, MUL, and DIV modules) with low computational complexity were designed to facilitate execution of ghost imaging. Hopefully, such modules would find additional useful applications in other protocols.

Our future work will focus on the following aspects. First, as introduced in the study, the quantum mask image consists of binary levels for every pixel, i.e. state $|0\rangle$ or $|1\rangle$, which is a strong astriction for the ghost imaging technique as well as its applications. Therefore, extensions to include modifications to its color information becomes an important focus for future studies. Second, it seems practical to extend our proposed QGI protocol to implement a quantum compressive sensing algorithm whose application will improve present approaches for single-pixel imaging via compressive sampling. Since ghost imaging usually requires a large number of speckle patterns, such an algorithm may improve efficiency when a high quality image is obtained in the proposed ghost imaging protocol. Finally, while evaluating the computational complexity of the QGI protocol, an advanced strategy should be developed to reduce the number of quantum gates. This will make the protocol more practicable.

REFERENCES
[1] N. Abura’ed, F. S. Khan, and H. Bhaskar, “Advances in the quantum theoretical approach to image processing applications,” ACM Comput. Surv., vol. 49, no. 4, pp. 1–49, Feb. 2017.
[2] L. Basano and P. Ottonello, “A conceptual experiment on single-beam coincidence detection with pseudothermal light,” Opt. Express, vol. 15, no. 19, pp. 12386–12394, Sep. 2007.
[3] R. S. Bennink, S. J. Bentley, and R. W. Boyd, “two-photon coincidence imaging with a classical source,” Phys. Rev. Lett., vol. 89, p. 113601, Aug. 2002.
[4] R. W. Boyd and P. J. Reynolds, “Introduction to the special issue on quantum imaging,” Quantum Inf. Process., vol. 11, no. 4, pp. 887–889, Aug. 2012.
[5] D. Deutsch, “Quantum theory, the church-turing principle and the universal quantum computer,” Proc. Roy. Soc. A, Math., Phys. Eng. Sci., vol. 400, no. 1818, pp. 97–117, Jul. 1985.
[6] B. I. Erkmen and J. H. Shapiro, “Ghost imaging: From quantum to classical to computational,” Adv. Opt. Photon., vol. 2, no. 4, pp. 405–450, Aug. 2010.
[7] R. P. Feynman, “Simulating physics with computers,” Int. J. Theor. Phys., vol. 21, nos. 6–7, pp. 467–488, Jun. 1982.
[8] A. Gatti, M. Bondani, L. A. Lungiato, M. G. A. Paris, and C. Fabre, “Comment on ‘can two-photon correlation of chaotic light be considered as correlation of intensity fluctuations?’” Phys. Rev. Lett., vol. 98, no. 039301, Jan. 2007.
[9] M. R. Hush, “Machine learning for quantum physics,” Science, vol. 355, no. 6325, p. 580, Feb. 2017.
[10] S. Heidari, M. Vafaei, M. Houshmand, and N. Tabatabae-Mashadi, “A dual quantum image scrambling method,” Quantum Inf. Process., vol. 18, p. 9, Nov. 2019.
[11] A. Iliyasu, “Towards realising secure and efficient image and video processing applications on quantum computers,” Entropy, vol. 15, no. 12, pp. 2874–2974, Jul. 2013.
[12] A. M. Iliyasu, “Roadmap to talking quantum movies: A contingent inquiry,” IEEE Access, vol. 7, pp. 23864–23913, 2019.
[13] A. M. Iliyasu, K. A. Abuhaseel, and F. Yan, “A quantum-based image fidelity metric,” in Proc. Sci. Inf. Conf., Jul. 2015, pp. 664–671.
[14] A. M. Iliyasu, P. Q. Le, F. Dong, and K. Hirota, “Watermarking and authentication of quantum images based on restricted geometric transformations,” Inf. Sci., vol. 186, no. 1, pp. 126–149, Mar. 2012.
[15] K. Kikuchi, “Quantum-classical communication protocol for secure image transmission by unauthorized users,” IEEE Trans. Inf. Theory, vol. 58, no. 3, pp. 1476–1489, Mar. 2012.
[16] H.-S. Li, P. Fan, H.-Y. Xia, H. Peng, and S. Song, “Quantum implementation circuits of quantum signal representation and type conversion,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 1, pp. 341–354, Jan. 2019, doi: 10.1109/tcsi.2018.2853655.
[17] H.-S. Li, X. Chen, S. Song, Z. Liao, and J. Fang, “A block-based quantum image scrambling for GNEQR,” IEEE Access, vol. 7, pp. 138233–138243, 2019.
[18] P. Li and X. Liu, “Bilinear interpolation method for quantum images based on quantum Fourier transform,” Int. J. Quantum Inf., vol. 16, no. 4, Jun. 2018, Art. no. 1850031.
[19] X. Liu, D. Xiao, W. Huang, and C. Liu, “Quantum block image encryption based on Arnold transform and sine chaotification model,” IEEE Access, vol. 7, pp. 57188–57199, 2019.
[20] M. A. Nielsen and I. Chuang, Quantum Computation and Quantum Information, Cambridge, U.K.: Cambridge Univ. Press, 2000.
[21] D. S. Oliveira and R. V. Ramos, “Quantum bit string comparator: Circuits and applications,” Quantum Comput. Comput., vol. 7, no. 1, pp. 17–26, Jul. 2007.
[22] T. B. Pitman, Y. H. Shih, D. V. Strekalov, and A. V. Sergienko, “Optical imaging by means of two-photon quantum entanglement,” Phys. Rev. A, Gen. Phys., vol. 52, no. 5, pp. R3429–R3432, Jul. 2002.
[23] G. Scarcelli, V. Berardi, and Y. H. Shih, “Scarcelli, berardi, and shih reply,” Phys. Rev. Lett., vol. 98, Jan. 2007, Art. no. 039302.
[24] J. H. Shapiro and R. W. Boyd, “Response to ‘the physics of ghost imaging- nonlocal interference or local intensity fluctuation correlation?’” Quantum Inf. Process., vol. 11, no. 4, pp. 1003–1011, Mar. 2012.
[25] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in Proc. 35th Annu. Symp. Found. Comput. Sci., Nov. 1994, pp. 124–134.
[26] R. Schneider et al., “Quantum imaging with incoherently scattered light from a free-electron laser,” Nature Phys., vol. 14, no. 2, pp. 126–129, Feb. 2018.
[27] Y. Shih, “The physics of ghost imaging-nonlocal interference or local intensity fluctuation correlation?” Quantum Inf. Process., vol. 11, no. 4, pp. 995–1001, Apr. 2012.
[28] A. Trabesinger, “Quantum computing: Towards reality,” Nature, vol. 543, p. S1, Mar. 2017.
[29] H. Takenaka, A. Carrasco-Casadó, M. Fujiwara, M. Kitamura, M. Sasaki, and M. Toyoshima, “Satellite-to-ground quantum-limited communication based on quantum transmission circuits of quantum signal representation and type conversion,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 1, pp. 341–354, Jan. 2019, doi: 10.1109/tcsi.2018.2853655.
[31] S. E. Venegas-Andraca, “Introductory words: Special issue on quantum image processing published by Quantum Information Processing,” Quantum Inf. Process., vol. 14, no. 5, pp. 1535–1537, May 2015.

[32] V. Vedral, A. Barenco, and A. Ekert, “Quantum networks for elementary arithmetic operations,” Phys. Rev. A, Gen. Phys., vol. 54, no. 1, pp. 147–153, Jul. 2002.

[33] F. Yan, A. M. Iliyasu, Y. Guo, and H. Yang, “Flexible representation and manipulation of audio signals on quantum computers,” Theor. Comput. Sci., vol. 752, pp. 71–85, Dec. 2018.

[34] F. Yan, A. M. Iliyasu, and S. E. Venegas-Andraca, “A survey of quantum image representations,” Quantum Inf. Process., vol. 15, no. 1, pp. 1–35, Jan. 2016.

[35] F. Yan, A. M. Iliyasu, and P. Q. Le, “Quantum image processing: A review of advances in its security technologies,” Int. J. Quantum Inf., vol. 15, no. 3, Apr. 2017, Art. no. 1730001.

[36] R. Zhou, W. Hu, P. Fan, and L. Hou, “Quantum realization of the bilinear interpolation method for NEQR,” Sci. Rep., vol. 7, May 2017, Art. no. 2511.

[37] R.-G. Zhou, Y.-N. Zhang, R. Xu, C. Qian, and I. Hou, “Asymmetric bidirectional controlled teleportation by using nine-qubit entangled state in noisy environment,” IEEE Access, vol. 7, pp. 75247–75264, 2019.

FEI YAN received the Ph.D. degree in engineering from the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Japan. He is currently an Associate Professor with the School of Computer Science and Technology, Changchun University of Science and Technology, China. He has worked as a Postdoctoral Fellow with the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology. He has published the first English book, Quantum Image Processing (Springer), in that field and more than 50 articles in the fields of quantum information processing, computational intelligence, and medical image analysis. He also served as the Chairman for the sessions on quantum information processing or image processing of several international conferences, including IEEE DSCI 2019, IEEE CIST 2018, and IEEE WISP 2015. He has been serving as an Associate Editor for the Journal of Advanced Computational Intelligence and Intelligent Informatics and a Guest Editor for the Journal of Medical Imaging and Health Informatics and the International Journal of Information Science and Technology.

KEHAN CHEN received the bachelor’s degree in science from Jilin University and the master’s and Ph.D. degrees in engineering from the Changchun University of Science and Technology, China. He is currently an Assistant Professor with the School of Computer Science and Technology, Changchun University of Science and Technology. His research interests include ghost imaging, quantum image processing, and quantum circuit implementation. He has published more than ten academic articles and obtained three national invention patents in the aforementioned fields.

KAORU HIROTA received the Dr.Eng. degree in electrical engineering from the Tokyo Institute of Technology, Japan. He is currently an Emeritus Professor with the Tokyo Institute of Technology and a 1000 Talents Professor with the Beijing Institute of Technology, China. His research interests include fuzzy systems, intelligent robot, computational intelligence, industrial informatics, and image understanding. He was the President of the International Fuzzy Systems Association (IFSA). He is also a Fellow of the International Fuzzy Systems Association (IFSA) and the President of the Japan Society for Fuzzy Theory and Systems (SOFT). He was recognized and awarded with the Chinese Friendship Medal, in 2019. He was a recipient of the Banki Donat Medal, the Henri Coanda Medal, the Grigore MOISIL Award, the SOFT Best Paper Award, and the Acoustical Society of Japan Best Paper Award. He has organized numerous international conferences, workshops, and symposia, as the Founding Chair, the General Chair, and the Program Chair. He is also the Editor-in-Chief of the Journal of Advanced Computational Intelligence and Intelligent Informatics.

ABDULLAH M. ILYASU (AKA ABDUL M. ELIAS) (Member, IEEE) received the M.E. and Dr.Eng. degrees in computational intelligence and intelligent systems engineering from the Tokyo Institute of Technology (Tokyo Tech.), Japan. He is currently a Research Faculty with the School of Computing, Tokyo Tech. He is also the Principal Investigator and the Team Leader of the Advanced Computational Intelligence and Intelligent Systems Engineering (ACIISE) Research Group, College of Engineering, Prince Sattam Bin Abdulaziz University (PSAU), Saudi Arabia. He is also a Visiting Professor with the School of Computer Science and Technology, Changchun University of Science and Technology, China. In addition to being a pioneer of the emerging quantum image processing (QIP) subdiscipline, he has to his credit more than 100 publications traversing the areas of computational intelligence, quantum cybernetics, quantum image processing, quantum machine learning, cyber and information security, hybrid intelligent systems, the Internet of Things, 4IR, health informatics, and electronics systems reliability. He has been the Managing Editor of Fuji Technology Press in Japan. He is a member of editorial board of the Journal of Advanced Computational Intelligence and Intelligent Informatics (JACII), Quantum Reports Journal, and the Journal of Medical Imaging and Health Informatics (JMIHI). He is also an Associate Editor of many other journals, including IEEE Access and Information Sciences.