Hindawi
Modelling and Simulation in Engineering
Volume 2022, Article ID 8128754, 11 pages
https://doi.org/10.1155/2022/8128754

Research Article

Integration of a Quantum Voting Scheme into Grayscale Images Using the Novel Enhanced Quantum Representation and Qiskit Framework

Alexandru-Gabriel Tudorache, Vasile Manta, and Simona Caraiman

Department of Computer Science and Engineering, "Gheorghe Asachi" Technical University of Iasi, D. Mangeron 27A, 700050 Iasi, Romania

Correspondence should be addressed to Alexandru-Gabriel Tudorache; alexandru-gabriel.tudorache@academic.tuiasi.ro

Received 24 September 2021; Revised 19 March 2022; Accepted 30 March 2022; Published 19 April 2022

Academic Editor: Angelos Markopoulos

This paper illustrates the way a proposed quantum voting scheme can be designed in combination with a steganography technique called Least Significant Bit (LSB), by modifying a small number of pixels in multiple grayscale images. It combines the voting scheme with the novel enhanced quantum representation (NEQR) of an image, where the LSBs of these pixels represent the vote for each entity that takes part in the voting process. A server is also used, not only to count but also to guarantee the integrity of the votes (which is done inherently, by its design and quantum properties). The superdense coding circuit is part of the design, allowing each voter to use one qubit in order to transmit two classical bits (the vote value). The selected platform for testing this scheme is IBM Quantum Experience, together with the open-source framework called Qiskit (written in Python). This framework allows users to create various quantum circuits, using a wide selection of quantum gates, and then to simulate them, either on a simulator or on a real quantum device. The quantum circuits and the measurement results are also presented in this paper.

1. Introduction

Secure voting using electronic systems has always been seen as an important problem by researchers, with multiple schemes proposed over the years. Numerous voting campaigns across the world, where multiple actors are involved with often conflicting interests, have always raised the issue of creating secure environments that guarantee various properties of the votes (correctness, integrity, and anonymity). Quantum computing, with its well-known properties such as entanglement and superposition, can be used to create certain connections between qubits that represent the votes (hidden in images and configurable by the voters) and qubits belonging to the server, therefore coming in aid of this dilemma. To be more precise, quantum computing allows us to entangle a qubit (or more) from the voting entity with one (or more) qubits from the server.

This article's innovative nature is the idea of bridging the gap between the fields of quantum voting schemes and steganography, as well as quantum image representations. It combines the quantum properties with the superdense coding circuit to show how qubits can be used to design a voting scheme that can be implemented on real systems, as well as simulators, also presenting another use for one of the most popular quantum algorithms. The main advantage of the scheme presented in this paper is that it uses one of the already proposed quantum image representation techniques; the algorithm extends the circuit designed for the novel

enhanced quantum representation to the field of steganography, thus presenting a potential application in the area of secure information transmission.

Some of the most important properties of quantum voting systems as well as various ideas that can be used in designing these systems are presented in paper [1]. The critical conditions required by a voting scheme are verifiability, security (one valid vote per person), and privacy. Among the security requirements, we mention the nonreusability (1 voter–1 vote), fairness/correctness (no data regarding the votes of others can be leaked before the vote ends), and eligibility. The voting systems can be grouped into two categories: the first implies that each voter has a separate container (ballot) for the vote, which must be sent to the server/master after the voter expresses his opinion (this idea can be referred to as DB (distributed ballot)); in the second one, the same container must reach all voters (TB (travelling ballot)). The concept of privacy, used in the field of quantum voting, is tied to the nature of the ballots, which, if handled in a secret manner, can lead to an architecture called anonymous voting. The scenarios for two voters and a generalized number of voters are analyzed, for both the TB and the DB concepts—the authors prove that the privacy is indeed protected. In the case of TB, entangled quantum states are selected to act as the ballot (set up by a master entity, referred to as *authority*). The entire demonstration for each discussed property can be consulted in paper [1].

This article is structured as follows: the second section presents some of the most important articles in the fields of quantum voting schemes and quantum steganography. The third section presents the proposed quantum image representations techniques, the key aspects of the quantum framework used to implement the operations (IBM Quantum Experience and Qiskit) and the proposed algorithm design. The fourth section presents the implementation details of this protocol for a sample grayscale image and the obtained results, after configuring and simulating the desired circuits. The last two sections describe some ideas to continue the research and point out the main contributions.

2. Related Work

Multiple ideas have been proposed in research papers from the quantum topic, and some of them are presented here, serving as inspiration for the current work. The articles have been grouped in two classes: those that approach the voting systems, using the quantum properties and proposing quantum circuits for this purpose, and articles that contain ideas for hiding data in images (quantum steganography).

2.1. Quantum Voting Systems. In article [2], the authors present a voting system where the ballot is in an unknown state. The state of the ballot cannot be easily modified, as the no-cloning theorem ensures the fact that no quantum state can be cloned. Anonymity is guaranteed, as the voters cannot choose a random value for the ballot (and then proceed with the algorithm). One important part of the presented algorithm, the encoding of the vote, is dependent on the ballot: since it is in a random state, each voter has to apply his quantum gates according not only to his vote value but also to the ballot's state. A *cut-and-choose* protocol is described, where if certain pieces of the vote are found to be invalid, the voting process is cancelled. Figures 1, 2 and 3 from paper [2] illustrate the state of the quantum ballot: blank, randomized blank, and a filled voting ballot, encoding a certain message, thus offering a better understanding of the protocol. The authors of article [2] believe that quantum schemes such as the one described will soon be able to break a large part of the actual cryptographic algorithms.

A voting scheme based on a multiparticle entangled state is presented in paper [3], where the security of the proposed system is analyzed. This scheme consists of voters, an administrator, who counts the votes, and a scrutinizer. The voting scheme is described for the participation of 3 voters (one of each type), one of whom (the scrutinizer) creates a number of qubit pairs in the Greenberger-Horne-Zeilinger (GHZ) maximally entangled state, which can be written as follows:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|000000\rangle \pm |111111\rangle\right). \quad (1)$$

The administrator receives the first particle, while the voter receives particles 2 and 3. According to a convention table that specifies the connection between the value of the vote and certain quantum gates, the voters perform multiple operations, which result in a different state configuration (the obtained quantum states are still entangled). The secret key is represented by a function chosen by the voter from the table. The paper also presents a detection process, ran by the administrator, that is part of the voting scheme (an example is given in Table 2 of article [3], which shows Alice's and Bob's quantum states in the detection process); an analysis for a *man-in-the middle* attack is also described, demonstrating the security of the protocol in this scenario. An extension idea is suggested for this scheme, based on quantum key distribution, and the above concepts can be further developed by carrying out various types of analysis for different attacks, involving parties that work together to sabotage the vote or try to interfere by performing illegal voting operations.

Paper [4] presents different ideas for quantum voting, based on entangled states, which by design ensure the anonymity of the votes; the proposed quantum protocol achieves a complexity reduction of order N, where N represents the number of voters. The paper describes a method called *comparative ballot*, where each vote represents the expression of a binary option; it illustrates a protocol for anonymous survey and a particular case of this survey, called *anonymous binary-valued ballot*—this extends the idea of a binary vote to a list of questions, each requiring a binary answer from the voters. The authors discovered an inverse relation between the privacy of the vote and the number of options for each voter, as a general property for systems of this kind. Two-valued and multivalued ballots are presented in paper [5], where a special operation is described, with the

purpose of stopping multiple votes from the same voter, as well as forbidding the cast of negative votes (thus stopping malicious voters from cheating). The protocol uses entangled states of continuous variables and also relies on the honesty of the tallyman, as the authors recommend that the operations performed by this entity should be done in a transparent (public) manner. Figure 3 from paper [5] also shows the experimental setup, for the representation and interaction with the voting ballot, using different modulators, a beam splitter, and an optical parametric oscillator.

2.2. Quantum Steganography. Steganography is the subfield of information processing that specializes in hiding various types of information inside messages or files; quantum steganography uses the special properties of the quantum universe, such as superposition and entanglement in achieving this purpose. Multiple ideas have been introduced in the recent years in this research area, some of which are highlighted below.

The authors of paper [6] describe a new watermarking protocol, combining the NEQR representation of an image (see paper [7]), the Gray code transform, and the LSB technique. A similar idea that also integrates the Arnold scrambling is illustrated in article [8], which yields a lower time complexity. Two keys are used in embedding and recovering the secret message in paper [9], where the bit-plane scrambling technique is also added, as a preliminary operation on the original image, before applying the Arnold transform. The exploiting modification direction (EMD) algorithm is the base concept on which the authors of article [10] build on, using color images and generating the key by using two of the (R, G, B) color channels. In paper [11], in addition to EMD, the authors hide the message using the bit-plane technique, and together with dynamically sharing between subgroups, they obtain a better embedding rate. The inverted pattern method can also be used (see paper [12]), where the state of the quantum key determines if the pixel from the message should be inverted. The key is saved in the LSBs of a parameterized number of pixels. A method of detection is presented in article [13], where the authors combine the NEQR representation with the LSB technique. The idea is to split the pixels into multiple blocks, which are then classified in 3 groups; evaluating each group reveals the potential presence of a hidden information. Multiple metrics that indicate the correct transmission are explained in paper [14], such as geometric coherence and ½-affinity coherence, evaluated for a version of the BB84 quantum protocol. The application of LSBq for multiwavelength quantum images in steganography is described in article [15], where the authors suggest using the modulo method and the Hilbert scrambling technique to embed the message. For recent developments and various techniques in the field of steganography, we also refer the reader to papers [16–33].

3. Materials and Methods

3.1. Image Representation Techniques. There have been a considerate number of proposals regarding quantum image representations, each with its advantages and restrictions. Among the most important ones we mention the flexible representation of quantum images (FRQI, introduced in paper [34]), where the quantum state of the system is written as a combination of pixel color and position, taking into account the angle vector for the color component. In the novel enhanced quantum representation (NEQR, presented in [7], used for grayscale images), the information about the pixels' color is stored in the multiqubit computational basis of the superposition state. Novel quantum representation of color digital images (NCQI, see [35]) is the generalized version of NEQR, applied to color images (based on the RGB color model). Another idea to process images is the new quantum representation model of color digital images (QRCI, proposed in paper [36]), which allows researchers to define the quantum state of the image using bit-planes.

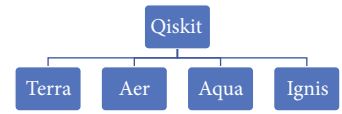

Figure 1: The Qiskit framework and its modules.

For the implementation part of this paper, we chose the NEQR representation, allowing us to have good control of the color information; we also have the ability to recover the original image in a finite number of quantum measurements.

3.2. IBM's Quantum Framework. The framework and computing platform used to test the proposed solution is IBM Quantum Experience [37], along with Qiskit, IBM's open-source framework, written in Python. IBM Quantum Experience is an online platform that allows users to design and simulate quantum circuits, by offering a wide selection of quantum gates to be added from a visual interface. Qiskit is intended for programmers who want to write their code and make their circuits more adaptable (or parameterized) for each algorithm they choose to test. In both scenarios, the researchers can opt to simulate the circuit on the local machine (limited in terms of qubits by the memory of the device) or on a real quantum device, accessible in the cloud; at this time, there are real machines that use 1, 5, and 15 qubits, available to the general public. If the real device is selected, the circuit is added to the execution queue for that respective device, and after the experiment is run, the results are available for analysis. There are four main parts in Qiskit, each with its own purpose—Qiskit Terra, Aer, Aqua, and Ignis. Terra is the name for the collection of tools used at the circuit level; Aer is the simulator part of Qiskit; Aqua can be seen as a library of algorithms, and Ignis helps researchers analyze the impact of noise in quantum systems. Figure 1 shows the mentioned parts of the Qiskit framework.

3.3. Algorithm Design. This paper proposes the following ideas for a quantum voting system. We assume that multiple entities want to take part in a voting process, where each

Table 1: Applied gate in correspondence with the vote value, used in the superdense coding circuit, where one qubit belongs to the voter and the other one to the server.

Vote value (binary)	Applied quantum gate
00	I
01	X
10	Z
11	ZX

voter can choose only one option from a maximum of 4 options (to which we have associated the values 0, 1, 2, and 3—we refer to this as the vote value). The proposed system is made up of a master entity, the server, responsible for the integrity and correctness of the votes; it shares 1 pair of qubits with each of the voters required for the vote value (one for the server and one for the voter), as well as one more qubit, shared between the server and each voter, that is set to state $|1\rangle$, after he has expressed his opinion. This qubit is set by each voter, using 2 additional qubits, both belonging to the voter. For example, if we take into consideration 3 voters (Alice, Bob, and Charlie), this means that the server has 3 qubits, each of them entangled with each voter's value qubit, and 3 more, to which we refer to as validation qubits, that are shared by the sever and each voter; they are initially set in the $|0\rangle$ state, and each voter then sets his corresponding validation qubit to $|1\rangle$, after he has set his value qubit (so 6 qubits in total for the server).

The server selects and sends the same grayscale image to each voter; before sending the image, the server resets (to 0) the LSBs of the voting pixels. In case a person decides not to vote, he simply does not set to $|1\rangle$ the vote validation qubit, which is equivalent to leaving it unmodified, to its initial state, $|0\rangle$. The voting pixels can be randomly chosen from the edges of the image or specifically selected from a certain region. For the ease of explaining and implementing the described algorithm, this paper analyzes the scenario where the voting pixels are chosen in the opposite corners of the image, namely, the upper-left and lower-right corners. After each voter receives the image, he modifies the LSB of the voting pixels and then represents them in their NEQR form. The value of the vote is encoded in the LSBs using the following convention:

$$\text{vote}_{\text{value}} = \text{LSB}_{\text{pixel}_{00}} \text{LSB}_{\text{pixel}_{(n-1)(m-1)\,(2)}}, \tag{2}$$

as binary notation, for an image of $n * m$ pixels. In decimal, this is equivalent to

$$\text{vote}_{\text{value}} = 2 * \text{LSB}_{\text{pixel}_{00}} + \text{LSB}_{\text{pixel}_{(n-1)(m-1)\,(10)}}. \tag{3}$$

The qubits for the LSBs are then used to set the state of the shared validation qubit to $|1\rangle$, when their state corresponds to the desired vote value (a Toffoli gate is used). Each voter also entangles his value qubit with a qubit on the server, respecting the quantum superdense coding circuit: he applies a certain gate depending on the value that he wants to send. Only one qubit from the voter is necessary to transmit 2 classical bits of information to the server. Table 1 shows the available options and the gates applied.

Once the voting window ends, the server analyzes the qubits for each voter, by first measuring the validation qubit, to make sure that the voter has expressed his vote and then by measuring the state of the corresponding value qubit for that voter, thus finding out his option. If the state of the validation qubit is $|0\rangle$ for a voter, then that particular vote is simply not counted. The server can then use the classical information processing tools to save and count all the votes. The entire voting system is schematically presented in Figure 2.

4. Implementation Details and Results

This section illustrates how the described procedure can be implemented using IBM's quantum framework, Qiskit, as well as the simulation results for the proposed circuits. The first part of the implementation focuses on the NEQR representation of an image (only the target pixels in our case) and on designing the circuit that allows the voter to set to $|1\rangle$ the state of the validation qubit, available to the server. This qubit informs the server that the voter has configured his image according to the vote value so that the server knows that the voter actually expressed his opinion at the end of the time voting window. The second part presents the interpretation of the superdense coding circuit, used in our case to transmit the vote value using a single qubit for each voter.

The server first selects a grayscale image (a pixel is represented in our case using 8 bits, in the interval 0–255) and sets its LSB to 0, for the two mentioned target pixels, as can be seen in Figure 3 (the original image can be found at source [38]; this is a downscaled version, used for demonstration purposes).

We analyzed the case of 3 voters, Alice, Bob, and Charlie, each of them free to choose one option out of 4, represented by values 0, 1, 2, and 3 in decimal (*00*, *01*, *10*, and *11* in binary). The first bit in the binary representation of the vote value is embedded in the upper-left pixel's LSB, while the second one is hidden in the lower-right pixel's LSB. Once the grayscale image is received by the voters, they modify it, according to their votes, and then represent those important pixels in their NEQR form. Assuming that Alice and Charlie both vote with "2" (10_2) and Bob votes with "1" (01_2), their images are presented in Figure 4.

Alice changes the LSB of the upper-left pixel to "1" (so her first pixel has gray level 137 instead of 136), while leaving the lower-right pixel's LSB set to "0" (keeping the gray value at 74). Similarly, Bob leaves the upper-left pixel in his image unmodified, at gray value 136, its LSB already being set to "0," and changes the lower-right pixel's LSB to "1" (the gray value becomes 75).

Below, we illustrated the NEQR representation of Alice's vote pixels, in Figure 5 (for the upper-left pixel) and Figure 6 (for the lower-right pixel), along with the measurement results obtained on the local simulator, using 1024 *shots*

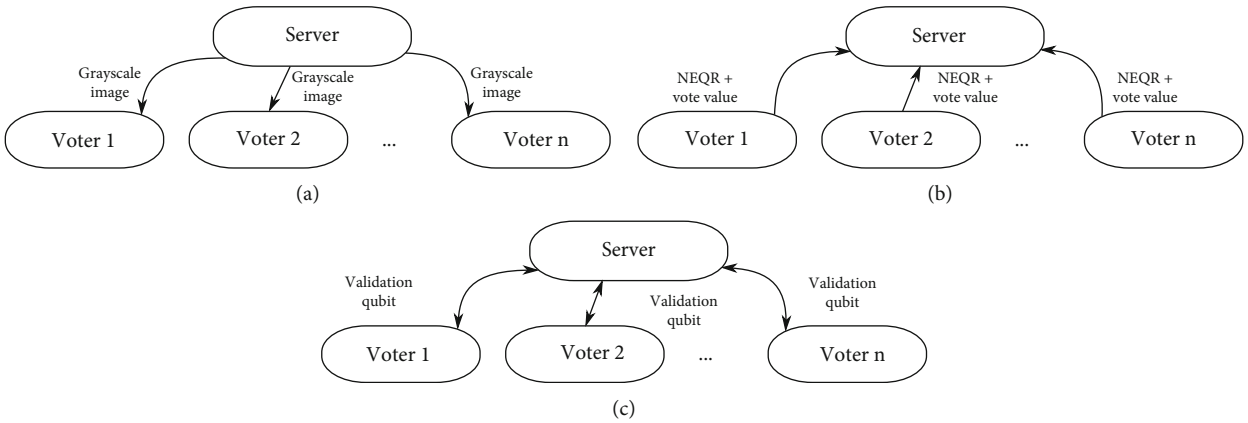

Figure 2: The proposed voting scheme. (a) Step 1: the voters first receive a grayscale image from the server. (b) Step 2: the voters create the NEQR representation of the pixels in the corners of the image. For each voter, the qubits representing the LSB denote the operation gate that needs to be applied to the voter's value circuit (the superdense coding circuit) so that the server can retrieve the actual vote. (c) Step 3: once each voter finishes setting the value qubit, he can set the validation qubit to $|1\rangle$; this indicates to the server that the voter expressed his option and can now add it, after the measurement, to the final count.

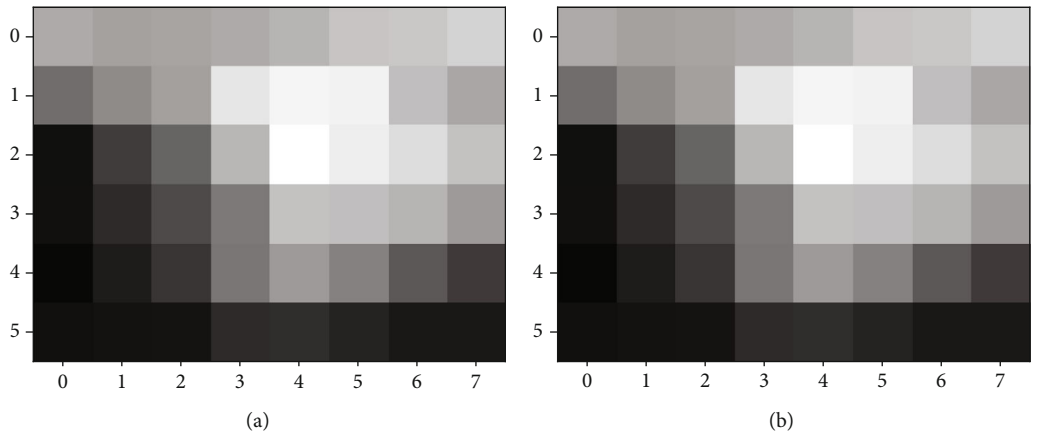

Figure 3: (a) The original 8×6 grayscale image selected by the server and (b) its modified version—the LSBs of the upper-left and lower-right pixels are set to 0. The differences between the two are hardly noticeable.

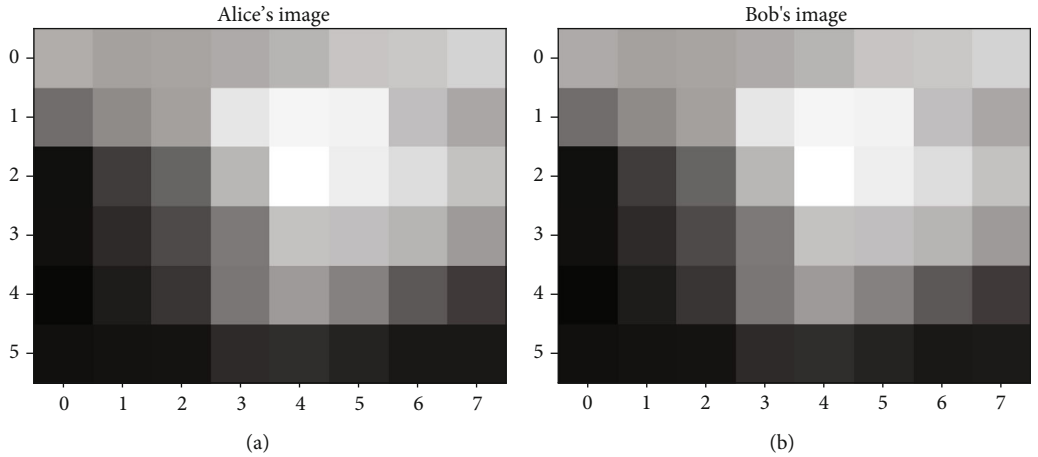

Figure 4: (a) Alice's (same as Charlie's) and (b) Bob's images, with their votes embedded in the upper-left and lower-right pixels' LSB.

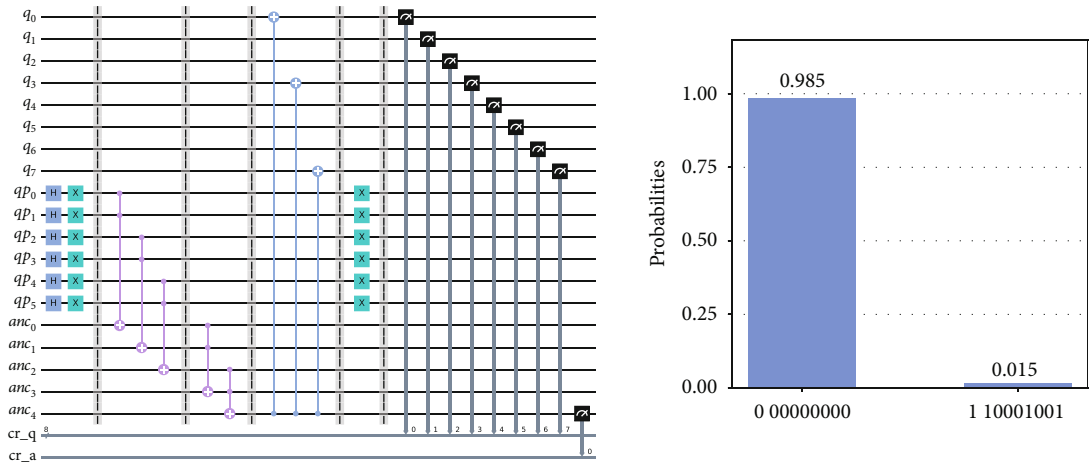

FIGURE 5: The NEQR representation of gray value 137 (the pixel in the upper-left corner of Alice's image, where the last bit was set to "1," changing the value from 136 to 137). The binary representation of 137 is 10001001_2, where the first bit (to the left) represents the MSB, corresponding to q_7, and the last one (to the right)—the LSB, corresponding to q_0; the classical binary representation of the pixel gray value should be seen in relation to qubits $q_7\, q_6\, q_5\, q_4\, q_3\, q_2\, q_1\, q_0$. The LSB of the pixel is represented by qubit q_0, and its state is set to $|1\rangle$ (the first bit from the binary representation of 10_2, Alice's vote). The auxiliary qubit anc_4 is set to state $|1\rangle$ only after the position qubits are in the desired state; here, for the first pixel, the desired states are all initially $|0\rangle$, since the target pixel is at coordinates (0, 0), so a round of NOT gates on all position qubits is required. The last three CNOT gates configure the states for the corresponding gray value qubits, q_7, q_3, and q_0, thus ensuring the state matches level 137. The purpose of the final layer of NOT gates is to allow another potential NEQR representation of a new gray value for a different pixel after this one (they must be set to cancel the NOT gates in the beginning). On the right, in the histogram that illustrates the result of the quantum measurement, we can see that the gray level qubits are only set to their desired states when the anc_4 qubit is in state $|1\rangle$.

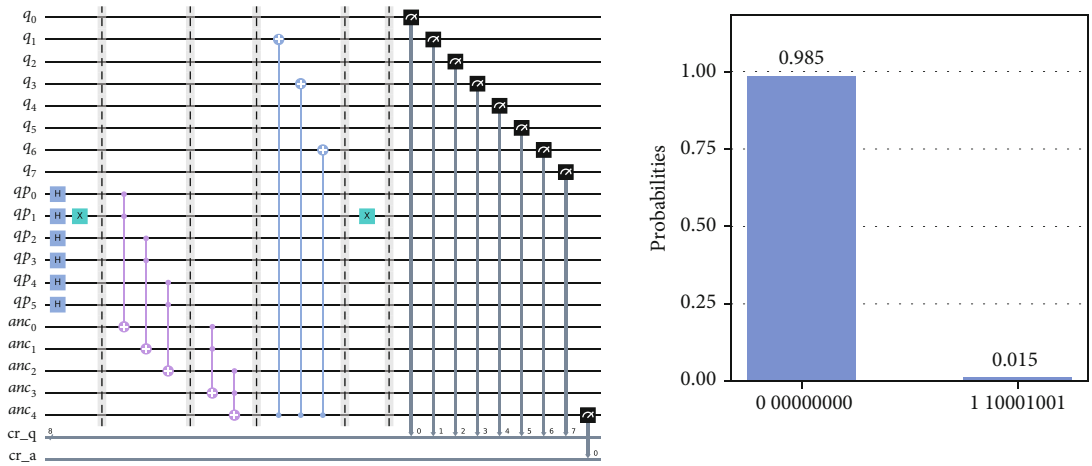

FIGURE 6: The NEQR representation of gray value 74 (the pixel in the lower-right corner of Alice's image). The LSB is represented by qubit q_0, and its state is set to $|0\rangle$ (the second bit from the binary representation of 10_2, Alice's vote in binary). The voter needs to add a NOT gate in the validation circuit for this qubit (q_0). In a similar manner to the representation in Figure 5, the $q_7\, q_6\, q_5\, q_4\, q_3\, q_2\, q_1\, q_0$ representation must match the binary value of 74, which is 00101111_2, when the position qubits are in the expected states. Here, the pixel's coordinates are (5, 7)—values 101_2 ($qp_2\, qp_1\, qp_0$) and 111_2 ($qp_5\, qp_4\, qp_3$) in binary; therefore, only one NOT gate is needed for the position qubits before the connection between anc_4 and the gray level qubits, more exactly for qp_1.

(program runs). This implementation requires 8 qubits for the gray level, 6 qubits for the position component (qp_5 - qp_3 for the columns, qp_2 - qp_0 for the rows), and 5 auxiliary qubits; the last auxiliary one, anc_4, is set to state $|1\rangle$ when all the position qubits match the state of the correct row and column. This means that 19 qubits are required in total for the NEQR representation. The circuit also contains 2 classical registers, *cr_q*, on 8 bits, and *cr_a*, on 1 bit, used to save the collapsed state, after measuring the gray level and the anc_4 qubit, respectively.

For the circuits in Figures 5 and 6 (the NEQR representation), we can formally write the action of the qubits as follows (all qubits are initially in the $|0\rangle$ state):

(1) The auxiliary qubits are deduced from the position qubits, using CCNOT gates:

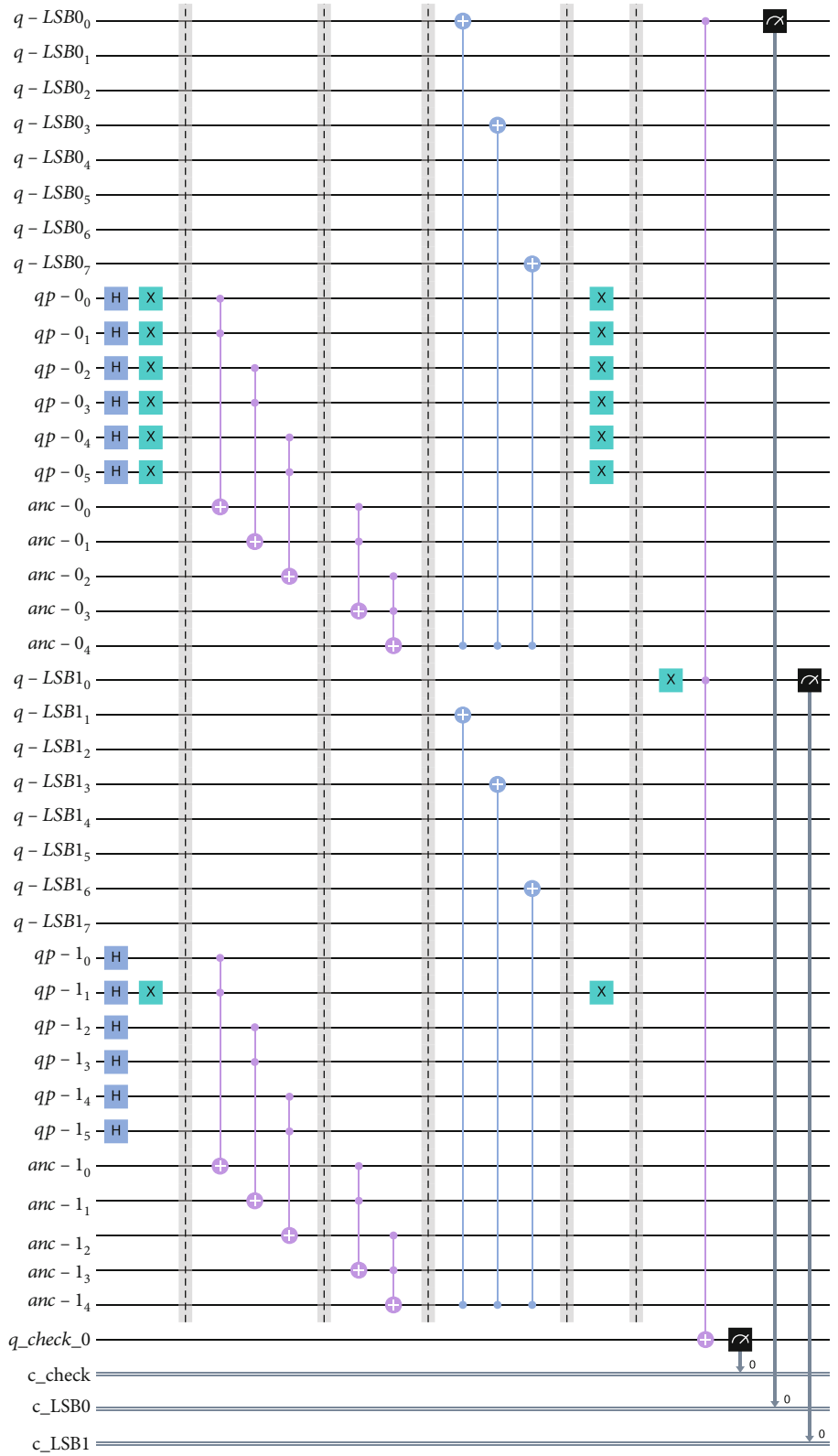

Figure 7: The NEQR representation of the two pixels describing Alice's vote: q-LSB0 (quantum register with 8 qubits) for the pixel in the upper-left corner (value 137) and q-LSB1 (quantum register with 8 qubits) for the pixel in the lower-right corner (value 74). Qubit q_check, which belongs to the server, is set to state $|1\rangle$ when the control qubits of the last Toffoli gate (after the second barrier) are both in state $|1\rangle$ (so the voter has to configure his part of the circuit according to his vote and add, if necessary, NOT gates for q-LSB0$_0$ and q-LSB1$_0$). In this particular situation, the voter added a NOT gate for q-LSB1$_0$. The representation for each gray value is explained in detail in Figures 5 and 6.

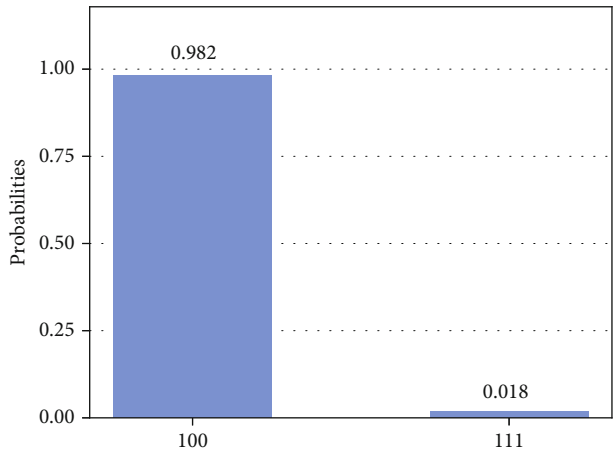

Figure 8: The simulation results for the circuit in Figure 7, where the measured qubits are q-LSB1$_0$, q-LSB0$_0$, and q_check (bottom to top)—there is a 0.018 probability of q_check being in state $|1\rangle$ (also conditioned by the $|1\rangle$ state of the other two measured qubits). Here, q-LSB1$_0$ is always in state $|1\rangle$, since the NEQR representation of gray value 74 leaves the LSB qubit in state $|0\rangle$ with 100% probability, and adding a NOT gate inverts its state (without modifying the probability).

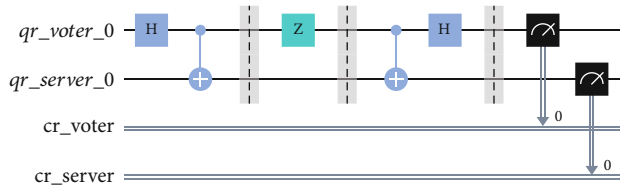

Figure 9: The superdense coding circuit, represented for Alice's vote value of 10_2. According to Table 1, the gate used to encode this value in a single qubit is the Z gate.

$$\begin{aligned}
\text{anc}_0 &= \text{anc}_0 \oplus (qp_0 \wedge qp_1),\\
\text{anc}_1 &= \text{anc}_1 \oplus (qp_2 \wedge qp_3),\\
\text{anc}_2 &= \text{anc}_2 \oplus (qp_4 \wedge qp_5),\\
\text{anc}_3 &= \text{anc}_3 \oplus (\text{anc}_0 \wedge \text{anc}_1),\\
\text{anc}_4 &= \text{anc}_4 \oplus (\text{anc}_2 \wedge \text{anc}_3).
\end{aligned} \tag{4}$$

(2) The gray value qubits (q_7 - q_0) are set using CNOT gates where necessary, using the anc$_4$ qubit; for value 137 in Figure 5, the state of qubits q_1, q_2, q_4, q_5, and q_6 is left unchanged, while qubits q_0, q_3, and q_7 are modified according to the following relations:

$$\begin{aligned}
q_0 &= q_0 \oplus \text{anc}_4,\\
q_3 &= q_3 \oplus \text{anc}_4,\\
q_7 &= q_7 \oplus \text{anc}_4.
\end{aligned} \tag{5}$$

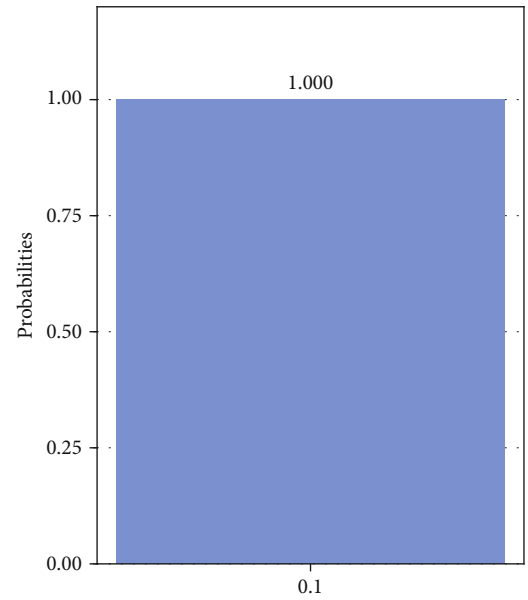

Figure 10: The measurement result for the circuit in Figure 9. The outcome for the vote value is always *10* (the result *01* can be explained by taking into account the fact that in Qiskit, the measurement results are read bottom-top, in our case from the lower classical bit, *cr_server*, to the upper one, *cr_voter*).

The two LSBs from the vote pixels are further used in another circuit, one that controls the server's validation qubit for that voter. This is achieved by combining the mentioned circuits, adding a NOT gate for each LSB if required (the voter can add any number of gates he desires for his qubits), and then adding a Toffoli gate, as shown in Figure 7. The final qubit, q_check, belongs to the server.

In addition to the equations describing the position, auxiliary, and gray value qubits, which still apply to the NEQR representations in Figure 7, we can also write equation (6) for q_check (its initial state in the circuit above is $|0\rangle$). Alice's vote is 10_2, and these 2 bits correspond to q-LSB0$_0$ and q-LSB1$_0$; for this vote value, a NOT gate is required for the q-LSB1$_0$ qubit (the LSB of value 74 is 0):

$$q_{\text{check}} = q_{\text{check}} \oplus \left(q - \text{LSB0}_0 \wedge \overline{q - \text{LSB1}_0}\right). \tag{6}$$

The circuit presented in Figure 7, containing 39 qubits, was simulated on the local machine, with an Intel i9-9900K processor at a base speed of 3.60 GHz, using approximately 16.56 GB of RAM, for 1024 shots, and the results are shown in Figure 8. The CPU ran at 100% at 4.68 GHz for the simulation time of 22.259 seconds; the Qiskit version used for this simulation was 0.25.2. For this circuit configuration, the probability of finding the q_check qubit in state $|1\rangle$ is close to the theoretical one ($1/64 = 0.015625$), as there is only one scenario where the qubits from both images are configured as desired.

The second part of the algorithm involves using the superdense coding circuit, which allows each voter to use a single qubit to send two bits of classical information (the

Table 2: Main results for the presented simulations.

Circuit	Number of qubits	Simulation time	RAM memory	Operating system	Processor speed (during simulation)	Qiskit version	Number of shots (program runs)
Alice's vote circuit (Figure 7)	39	22.259 seconds	16.56 GB	Windows 10	4.68 GHz	0.25.2	1024
Superdense coding circuit (Figure 9)	2	0.349 seconds	Unnoticeable for 2 qubits				

voting value in binary) to the server. For Alice's vote value, 10_2, the circuit is presented in Figure 9.

By simulating the circuit in Figure 9 and measuring the results, we obtain the expected result of 10_2 (1 for the *cr_voter* and 0 for *cr_server*) with probability 100% (Figure 10). The simulation took place on the same local machine described above (with the same configuration), and it required only 0.349 seconds (the RAM change is unnoticeable since we are only simulating 2 qubits).

Once all voters have first set the state for their value qubit (by adding the corresponding gate for their vote value) and then set the state of their validation qubit to $|1\rangle$, the voting system requires the server to wait for the dedicated time window to close. At the end of this time interval, the server first validates each vote by making sure that the state of the validation qubit is $|1\rangle$ and then obtains the vote value from the superdense coding circuit and counts it. The counting process can be done in a classical manner, after which the server can post the results on a public website (assuming it can be done so securely, otherwise a different system can be used to share the vote results). In the presented situation, the server would nominate candidate "2" as the election winner with 2 votes (from Alice and Charlie), while candidate "1" would have a single vote from Bob, and there would be no votes for the other 2 candidates.

The simulation conditions, describing the machine used to create the above circuits (for Alice's vote value circuit, in Figure 7, and the superdense coding circuit, from Figure 9), together with the simulation results, can also be summarized in Table 2.

5. Discussion

In this section of the paper, we present some of the ideas that can be developed in order to continue the research described above. These ideas are grouped as follows:

(a) The voting system can rely on different types of images (RGB images) to be shared by the server, as well as various types of quantum representations techniques for these images (FRQI, NCQI, and QRCI)

(b) A different algorithm can be used to embed the vote, while still maintaining a connection to the chosen image representation technique. This paper selects the LSB of certain pixels; another idea is to use the average value of the gray pixel intensity from dedicated regions of the image. The image alteration techniques can also be done not only in the time domain, but also in the frequency domain (using the quantum Fourier transform)

(c) The voting system can be extended to allow more candidates and more options for the voters. If more candidates decide to take part in the election, this requires designing an extension of the superdense coding circuit (more qubits are therefore needed for the server). The voters can also express their view using an option from the following group for each candidate (approve, reject, and abstain), which translates to a larger part of the image being used to express the vote (a certain region for each candidate)

The potential implications of designing such a voting system, together with the mentioned alternatives and extension ideas, are far-reaching: by using different types of images, or even other types of data, the size of the information given by the users could be very well increased; its efficiency could also be improved, allowing for a better embedding system, being harder to notice by the human eye. The same can be said about using a different domain representation (such as the frequency one). Implementing the discussed algorithm (or close variations) on a future, but already existing, quantum infrastructure, would reduce the costs required for building secure, dedicated devices that would be used only for elections. The benefits would extend even further, not only from a financial point of view but also from a sociodemographic perspective—if a system such as this would be deployed, then all countries would have access to a more transparent and democratic voting system; this is crucial for assuring that the will of the people is not tampered with and can therefore help a civilization advance in the right direction and at a greater speed. Its usage could also be intertwined with the cryptographical area, perhaps helping in generating private and/or public keys for the already existing crypto-systems or be selected in the design of new quantum protocols. The scheme described in this article can be extended to other algorithms (new or existing), where the vote values can be interpreted as required by those protocols. Choosing to use color images adds a new layer of complexity, given by their quantum representations, but also a better integration with what the human eye is already accustomed to. Color images, by having more components in their representation (such as RGB), allow for more granular modifications; for example, integrating a larger number could be done by making small changes in each component, with a dedicated algorithm required for recovering it. The result of such an action would be an image that looks closer

to the original one and much less prone to be analyzed by an unknowing third party.

6. Conclusions

The proposed paper shows a way in which the properties of the quantum information processing universe are used when designing a voting scheme. The article presents the details of a voting system, which allows its participants to select a candidate, by choosing an index associated with it. Multiple voters express their option and after the vote ends, a server counts and validates each vote, these entities having shared a couple of qubits. It can be seen as a two-part system: the first part describes the quantum representation of images (NEQR), together with the application of a steganography technique (LSB), which are used to tell the server that each voter expressed his opinion (actually voted); in the second part, the superdense coding circuit is selected to transmit the vote value. The paper also illustrates the mentioned concepts, by creating and simulating the corresponding circuits, implemented with the help of the Qiskit quantum framework.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the "Gheorghe Asachi" Technical University of Iasi.

References

[1] M. Hillery, M. Ziman, V. Bužek, and M. Bieliková, "Towards quantum-based privacy and voting," *Physics Letters A*, vol. 349, no. 1-4, pp. 75–81, 2006.

[2] T. Okamoto, K. Suzuki, and Y. Tokunaga, "Quantum voting scheme based on conjugate coding," *NTT Technical Review*, vol. 6, no. 1, pp. 1–8, 2008.

[3] P. Xue and X. Zhang, "A simple quantum voting scheme with multi-qubit entanglement," *Scientific Reports*, vol. 7, pp. 1–4, 2017.

[4] J. A. Vaccaro, J. Spring, and A. Chefles, "Quantum protocols for anonymous voting and surveying," *Physical Review A*, vol. 75, pp. 1–8, 2007.

[5] L. Jiang, G. He, D. Nie, J. Xiong, and G. Zeng, "Quantum anonymous voting for continuous variables," *Physical Review A*, vol. 85, article 042309, pp. 1–6, 2012.

[6] W. W. Hu, R.-G. Zhou, J. Luo, and B. Y. Liu, "LSBs-based quantum color images watermarking algorithm in edge region," *Quantum Information Processing*, vol. 18, pp. 1–27, 2019.

[7] Y. Zhang, K. Lu, Y. Gao, and M. Wang, "NEQR: a novel enhanced quantum representation of digital images," *Quantum Information Processing*, vol. 12, pp. 2833–2860, 2013.

[8] R.-G. Zhou, W. Hu, and P. Fan, "Quantum watermarking scheme through Arnold scrambling and LSB steganography," *Quantum Information Processing*, vol. 16, pp. 1–21, 2017.

[9] R.-G. Zhou, J. Luo, X. Liu, C. Zhu, L. Wei, and X. Zhang, "A novel quantum image steganography scheme based on LSB," *International Journal of Theoretical Physics*, vol. 57, no. 6, pp. 1848–1863, 2018.

[10] W.-W. Hu, R.-G. Zhou, X.-A. Liu, J. Luo, and G.-F. Luo, "Quantum image steganography algorithm based on modified exploiting modification direction embedding," *Quantum Information Processing*, vol. 19, pp. 1–28, 2020.

[11] Z. Qu, H. Sun, and M. Zheng, "An efficient quantum image steganography protocol based on improved EMD algorithm," *Quantum Information Processing*, vol. 20, pp. 1–29, 2021.

[12] G. Luo, R.-G. Zhou, and W. W. Hu, "Efficient quantum steganography scheme using inverted pattern approach," *Quantum Information Processing*, vol. 18, pp. 1–24, 2019.

[13] J. Luo, R.-G. Zhou, W.-W. Hu, G.-F. Luo, and G. Z. Liu, "Detection of steganography in quantum grayscale images," *Quantum Information Processing*, vol. 19, pp. 1–17, 2020.

[14] Z. Qu, Y. Huang, and M. Zheng, "A novel coherence-based quantum steganalysis protocol," *Quantum Information Processing*, vol. 19, pp. 1–19, 2020.

[15] E. Şahin and İ. Yilmaz, "A novel quantum steganography algorithm based on LSBq for multi-wavelength quantum images," *Quantum Information Processing*, vol. 17, pp. 1–24, 2018.

[16] R. Chetia, S. M. B. Boruah, and P. P. Sahu, "Quantum image edge detection using improved Sobel mask based on NEQR," *Quantum Information Processing*, vol. 20, pp. 1–25, 2021.

[17] S. S. Yadahalli, S. Rege, and R. Sonkusare, "Implementation and analysis of image steganography using least significant bit and discrete wavelet transform technique," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1325–1330, Coimbatore, India, 2020.

[18] V. Kalaichelvi, P. Meenakshi, P. Vimala Devi, H. Manikandan, P. Venkateswari, and S. Swaminathan, "A stable image steganography: a novel approach based on modified RSA algorithm and 2–4 least significant bit (LSB) technique," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 7235–7243, 2020.

[19] A. Jain, "A secured steganography technique for hiding multiple images in an image using least significant bit algorithm and Arnold transformation," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 38, pp. 373–380, 2020.

[20] A. Alabaichi, M. A. A. K. Al-Dabbas, and A. Salih, "Image steganography using least significant bit and secret map techniques," *International Journal of Electrical & Computer Engineering*, vol. 10, no. 1, pp. 935–946, 2020.

[21] H. A. Atee, A. K. Yasari, R. Ahmad, and N. Mohd Noor, "Text in image steganography based on a dynamic non-sequential least significant bit technique in grayscale and RGB images," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 5, pp. 1615–1622, 2019.

[22] P. Artiemjew and A. Kislak-Malinowska, "Using *r*-indiscernibility relations to hide the presence of information for the least significant bit steganography technique," *Communications in Computer and Information Science*, vol. 1078, pp. 209–220, 2019.

[23] B. Abd-El-Atty, A. A. Abd El-Latif, and S. E. Venegas-Andraca, "An encryption protocol for NEQR images based

on one-particle quantum walks on a circle," *Quantum Information Processing*, vol. 18, pp. 1–26, 2019.

[24] P. Fan, R.-G. Zhou, W. Hu, and N. Jing, "Quantum image edge extraction based on classical Sobel operator for NEQR," *Quantum Information Processing*, vol. 18, pp. 1–23, 2019.

[25] A. Gupta and S. Ahuja, "An improved image steganography technique using block division least significant bit approach," in *2018 international conference on advances in computing, communication control and networking, ICACCCN*, pp. 335–339, Greater Noida, India, 2018.

[26] I. G. Wiryawan and I. G. A. Gunadi, "Steganography based on least significant bit method was designed for digital image with lossless compression technique," in *2018 International Conference on Signals and Systems, ICSigSys 2018*, pp. 98–102, Bali, Indonesia, 2018.

[27] N. Mounika, K. Kalaivani, and A. V. Phamila, "Enhanced data hiding using least significant bit masking technique for image steganography," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 9, no. 6, pp. 90–96, 2017.

[28] R.-G. Zhou, W. Hu, P. Fan, and H. Ian, "Quantum realization of the bilinear interpolation method for NEQR," *Scientific Reports*, vol. 7, no. 1, pp. 1–17, 2017.

[29] G. Sugandhi and C. P. Subha, "Efficient steganography using least significant bit and encryption technique," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1–6, Coimbatore, India, 2016.

[30] J. Sang, S. Wang, and X. Niu, "Quantum realization of the nearest-neighbor interpolation method for FRQI and NEQR," *Quantum Information Processing*, vol. 15, no. 1, pp. 37–64, 2016.

[31] G. Swain and S. K. Lenka, "LSB array based image steganography technique by exploring the four least significant bits," *International Conference on Computing and Communication Systems*, 2012, pp. 479–488, Berlin, Heidelberg, 2012.

[32] G. Swain and S. K. Lenka, "A robust image steganography technique using dynamic embedding with two least significant bits," *Advanced Materials Research*, vol. 403, pp. 835–841, 2011.

[33] M. Asad, J. Gilani, and A. Khalid, "An enhanced least significant bit modification technique for audio steganography," in *International Conference on Computer Networks and Information Technology*, pp. 143–147, Abbottabad, Pakistan, 2011.

[34] P. Q. Le, F. Dong, and K. Hirota, "A flexible representation of quantum images for polynomial preparation, image compression, and processing *operations*," *Quantum Information Processing*, vol. 10, no. 1, pp. 63–84, 2011.

[35] J. Sang, S. Wang, and Q. Li, "A novel quantum representation of color digital images," *Quantum Information Processing*, vol. 16, pp. 1–14, 2017.

[36] L. Wang, Q. Ran, J. Ma, S. Yu, and L. Tan, "QRCI: a new quantum representation model of color digital images," *Optics Communication*, vol. 438, pp. 147–158, 2019.

[37] IBM Quantum, 2021, https://quantum-computing.ibm.com.

[38] H. W. Smith, 2021, https://www.freeimages.com/photographer/battelking-57024.