Quantum Edge Detection - QHED Algorithm

Bernardo Leite
Electrical and Computer Eng.
University of Coimbra
Coimbra, Portugal
uc2017268958@student.uc.pt

Ângelo Huang
Electrical and Computer Eng.
University of Coimbra
Coimbra, Portugal
uc2018292729@student.uc.pt

***Abstract*—Edge Detection is a crucial step in any image feature extraction process. Modern classical image processing algorithms make considerable use of edge detection to extract the structure of the objects and features shown in an image. Being a new field, quantum image processing is fascinating and, in some circumstances, allows for exponential speedups over traditional picture processing. The use of quantum computing in edge detection has the potential to improve the performance of image processing algorithms, making them faster and more accurate. Quantum Hadamard Edge Detection is a technique used to detect edges in digital images using quantum computing. The technique utilizes the Hadamard transform, a linear transformation commonly used in image processing, in a quantum computing context. The Hadamard transform is applied to the image using quantum gates, which allows for the detection of edges with high precision and speed. In this paper, we start by looking at how an image can be represented in the quantum realm, then we analyse the Quantum Hadamard Edge Detection. Finally, some drawbacks and concerns, as well as future hopes for the field are presented.**

***Index Terms*—Quantum Edge Detection, Quantum Computing**

I. Introduction

Quantum computing has long held out the promise of increasing processing efficiency in a variety of sectors. The quantum algorithm has been shown to successfully solve the NP problem of computers and get around the bottleneck caused by conventional computing techniques since the 1990s when Shor and Grover algorithms were proposed. Traditional images play a significant role as information carriers in multimedia, and the visual system is the primary source of information for 70% of people. However, due to the substantial amount of digital image data and storage space, an efficient processing method is urgently required. The use of geometric transformation, pattern recognition, image classification, quantum morphology, quantum edge extraction, image matching, and quantum watermarking are the main applications of the new technology known as quantum image processing. The image edge detection algorithm is crucial to the study of image processing and is crucial for image target detection, mosaic creation, and segmentation. Classic image edge extraction algorithms have been continuously proposed in recent years. The calculation of a $2^n \times 2^n$ image cannot be completed using the well-known classical edge extraction algorithms due to their complexity $O(2^n)$. The breakthrough in image edge detection technology in the new era is achieved by quantum image processing, which accelerates image computation using quantum mechanics' physical properties.

II. Quantum Image Representation

A critical component of quantum image processing and the processing of quantum information is quantum image representation. With the Quantum Probability Image Encoding (QPIE), we explore the quantum image representation. The pixel values of a classical image are stored using the QPIE representation using the probability amplitudes of a quantum state.

Yao et al. proposed the QPIE notation [1] based on the Real Ket notation, and is especially interesting in the field of feature extraction, a key process in machine learning and artificial intelligence applications, that relies a lot on Edge Detection [7]. When the pixels of an image, I, are transformed into a single-dimension vector using the flexible representation of quantum image (FRQI) notation, the resulting vector can be mapped onto a pure quantum state that contains the image data. As a result, if the described vector $\vec{i}$ has the dimensions $rows \times columns = N$and $\vec{i} = vec(I)$, then the quantum state that maps the image data is

$$|i\rangle = \sum_{k=0}^{2^n-1} c_k |k\rangle, n = [\log_2(N)] \tag{1}$$

where n is the number of necessary qubits and the coefficient c_k encodes the intensity of the pixel. The intensities must be normalised so that the sum of the squares of all the probability amplitudes equals 1, as the pixel value must be represented as the probability amplitudes of a quantum state. So we have

$$c_k = \frac{I_{yx}}{\sqrt{\sum I_{yx}^2}}, k < N \tag{2}$$

for every c_k corresponding to I_{yx}, where y denotes the pixel's row and x its column. A few rotation and CNOT gates [7] can be used to create such a state very effectively, and it has been demonstrated that QPIE uses fewer resources than the other quantum image representation techniques [1].

III. Quantum Edge Detection Algorithm

Typically, classical edge detection algorithms rely mostly on the computation of image gradients identifying locations in the image for dark-to-light intensity transitions. Ergo, the worst case time complexity for most of them is $O(2^n)$. This indicates that the gradients must be calculated for each individual pixel.

In order to study the edge detection, we first need to find adequate ways of detect them in the quantum realm, so in this section various edge detections edge detection algorithms of images in the quantum realm will be briefly described.

A. QSobel

QSobel, a novel quantum image edge extraction algorithm that was proposed as an extension of the classical Sobel edge detection method. The Sobel edge detection method is a popular algorithm for detecting edges in images, based on the gradient of the image intensity. QSobel algorithm uses a quantum version of the Sobel filter, which is implemented using quantum gates. The quantum Sobel filter is applied to the image, and then a quantum measurement is performed to extract the edge information. QSobel algorithm was designed by Yi Zhang, Kai Lu and YingHui Gao [4], which combines the quantum image model of FRQI and Sobel edge extraction algorithm. Through designing the quantum circuit of QSobel, it is reported that QSobel can extract edges in the computational complexity of $O(n^2)$ for a FRQI quantum image with a size of $2^n \times 2^n$. One of the main advantages of QSobel is that it uses a smaller number of quantum gates compared to other quantum edge detection algorithms, which makes it more feasible for implementation on current and near-term quantum devices.

B. Edge detection of the quantum image based on the Kirsch operator

The Kirsch operator is a type of edge detection filter that is commonly used in image processing. It consists of a set of 8 filters, each of which is used to compute the gradient of the image intensity in a specific direction. The Kirsch operator is considered to be more robust to noise and more accurate than other edge detection methods like the Sobel operator.

In recent years, researchers have proposed a quantum version of the Kirsch operator for edge detection in quantum images. The basic idea is to use quantum gates to perform the Kirsch operator on the quantum image, and then perform quantum measurements to extract the edge information. The use of the Kirsch operator in a quantum context has the potential to improve the speed and accuracy of edge detection compared to classical methods.

Xu Pengao et al. [5], employed a novel enhanced quantum representation (NEQR) as the image representation model for processing quantum image, which generates results of edge extraction using the Kirsch operator. The implementation of a quantum version of the Kirsch operator for edge detection in quantum images would involve the following steps:

- Step 1: Eight quantum images are obtained by coordinate transformation. A set of quantum images $|I_{xy}\rangle$ can be obtained by cyclic shifting of the original image.

$$\left\{\begin{matrix} |I_{x-1y-1}\rangle, & |I_{xy-1}\rangle, & |I_{x+1y-1}\rangle, & |I_{x-1y}\rangle, & |I_{xy}\rangle \\ |I_{x+1y}\rangle, & |I_{x-1y+1}\rangle, & |I_{xy+1}\rangle, & |I_{x+1y+1}\rangle & \end{matrix}\right\}$$
(3)

- Step 2: The gradient of the pixel is calculated by Kirsch operator.

$$\begin{aligned}
|M_0\rangle &= qADD(|I_{x-1y-1}\rangle, |I_{x-1y}\rangle)\\
|M_1\rangle &= qADD(|M_0\rangle, |I_{x-1y+1}\rangle)\\
|M_2\rangle &= qADD(|M_1\rangle, |M_1\rangle)\\
|M_3\rangle &= qADD(|M_2\rangle, |M_2\rangle)\\
|M_4\rangle &= qADD(|M_3\rangle, |M_1\rangle)\\
|N_0\rangle &= qADD(|I_{xy-1}\rangle, I_{xy+1}\rangle)\\
|N_1\rangle &= qADD(|I_{x+1y-1}\rangle, I_{x+1y}\rangle)\\
|N_2\rangle &= qADD(|I_{x+1y+1}\rangle, N_0\rangle)\\
|N_3\rangle &= qADD(|N_1\rangle, |N_2\rangle)\\
|N_4\rangle &= qADD(|N_3\rangle, |N_3\rangle)\\
|N_5\rangle &= qADD(|N_4\rangle, |N_3\rangle)\\
|K_0\rangle &= qSUB(|M_4\rangle, |N_5\rangle)
\end{aligned}$$
(4)

By analogy, K_{MAX} is finally obtained by template operation. According to the gradient operation, the processed image is:

$$|I_{\hat{k}}\rangle = \frac{1}{2^n} \sum_{X=0}^{2^{2n}-1} \sum_{Y=0}^{2^{2n}-1} |f_{k_{max}}(Y,X)\rangle |YX\rangle \tag{5}$$

- Step 3: The gradient is classified by the threshold operation U_T, set a threshold T:

$$U_T(\sum_{YX=0}^{2^{2n}-1} |I_{YX}\rangle|0\rangle) = \sum_{I_{YX}\geq T} |I_{YX}\rangle|1\rangle + \sum_{I_{iX}\leq T} |I_{iYX}\rangle|0\rangle \tag{6}$$

Finally, the expected output quantum state:

$$|T\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} |T_{YX}\rangle|YX\rangle, T_{YX} \in \{0,1\} \tag{7}$$

When $T_{YX} = 1$ means edge point, otherwise not.

C. Quantum SUSAN edge detection

Based on the classical SUSAN operator and the quantum genetic algorithm (QGA), a new edge detection algorithm was designed [6]. To get the qubit information in the form of a vector product, the unitary matrix operation is first applied to the quantum image expressed by the NEQR model. The pixel gradient value is then calculated. The double chains quantum genetic algorithm (DCQGA) is used to optimise the classifier's various parameters before the quantum SUSAN circuit classifies the gradient values to produce the best classification result. In order to reconstruct the classical edge results, the quantum edge detection results are decoded and stored in the new NEQR images.

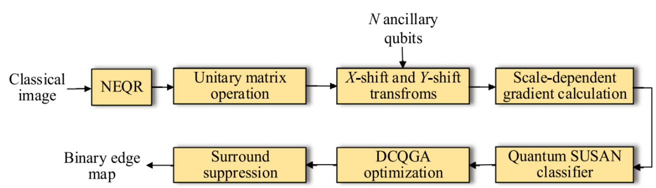

Fig. 1. QSED algorithm. [6]

The block diagram of the quantum SUSAN edge detection (QSED) algorithm is displayed in Fig.13. The algorithm mainly includes quantum image representation, quantum image shift, quantum pixel gradient estimation, SUSAN edge gradient classification, DCQGA optimization, edge point estimation, and quantum measurement. The modified QSED principle differs from the classical SUSAN principle in two aspects. The first aspect is that the improved QSED scheme deals with quantum gradient instead of original grayscale and the second aspect is that the size and the shape of the masks are different. The following describes the precise edge detection process:

1) The quantum image $|Q\rangle$ is prepared with NEQR model. To store an image of size $2^n \times 2^n$, $2n+q$ qubits are required. The neighborhood pixel values are stored in extra q qubits.
2) A unitary matrix operation is performed on the quantum image $|Q\rangle$ obtained in the previous step, and the results are transmitted to the X-shift and the Y-shift transforms circuit.
3) The pixel gradient value is obtained by the quantum operations.
4) The gradient value is transmitted to the quantum SUSAN classifier, and the edge points and non-edge points can be roughly determined by comparing the gradient value of the edge point with that around it.
5) The DCQGA is introduced to further optimize the classification process. The surround suppression operation can suppress the texture edge points with the SUSAN classifier.

IV. Implementation of QHED Algorithm

Yao et al. proposed The Quantum Hadamard Edge Detection (QHED) algorithm [1], which aims to bring edge detection, a crucial procedure in conventional image processing, to the world of quantum computing. Edge detection algorithms locate the areas in the image where there are abrupt transitions by using computations of the colour intensity gradient. According to [7], the worst-case scenario has a time complexity of $O(2^n)$, which means that each pixel must be processed separately in order to produce the gradients of the image. In comparison to these conventional algorithms, quantum computing has the potential to offer exponential speedup. This is where QHED enters the picture, as one of the efficient quantum alternative. In the following, is described a simplified QHED algorithm.

Given an N-pixel image, we first encode our picture by the QPIE representation described in Section II, and we get an encoded state of $|i\rangle = \sum_{k=0}^{2^{n-1}} c_k|k\rangle$. Let $I_{2^{n-1}}$ be an identity matrix with size 2^{n-1}. Then, we apply a Hadamard gate to the last qubit, which has the following unitary matrix:

$$I_{2^{n-1}} \otimes H_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix} \tag{8}$$

After applying the Hadamard gate, we obtain the following vector:

$$(I_{2^{n-1}} \otimes H_0) \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{N-2} \\ c_{N-1} \end{bmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{bmatrix} c_0 + c_1 \\ c_0 - c_1 \\ c_2 + c_3 \\ c_2 - c_3 \\ \vdots \\ c_{N-2} + c_{N-1} \\ c_{N-2} - c_{N-1} \end{bmatrix} \tag{9}$$

Note that the entries $(c_k - c_{k+1})$ is the difference between two consecutive pixels. Also, all entries of the form $(c_k - c_{k+1})$ are located at the even entries. Therefore, after measurement of all qubits, we only need to look at cases where the last qubit is 1. If the difference is large, then we admit that there is an edge between the two consecutive pixels. The algorithm can be analysed in detail on the Qiskit Notebook [7]. In short, it relies on the generalisation of the Hadamard gate, using it to perform the edge detection. The outcomes from the implementation of the suggested algorithm will be presented in this section. The Jupiter notebook was used to produce the results shown below. The original image is an 8×8 pixel matrix with pixels that can only have a colour intensity of 0 or 1. This restriction was made to make the algorithm's implementation easier.

The edges of the image were successfully identified by the QHED algorithm, as can be seen.

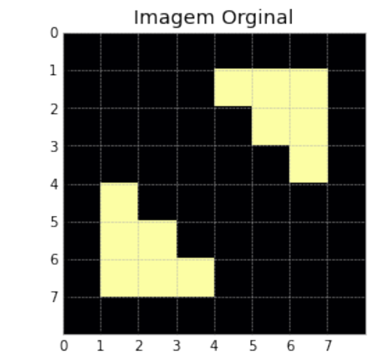

Fig. 2. Original image (8×8 matrix).

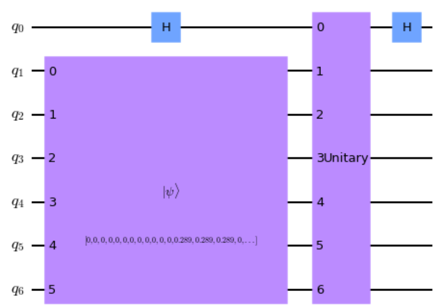

Fig. 3. Circuit for horizontal scan of the image.

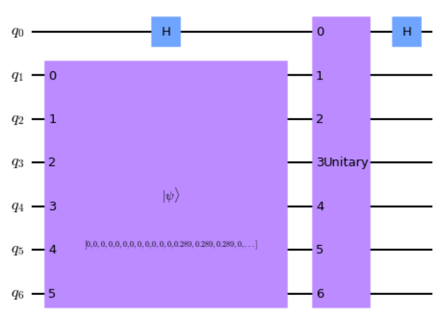

Fig. 4. Circuit for vertical scan of the image.

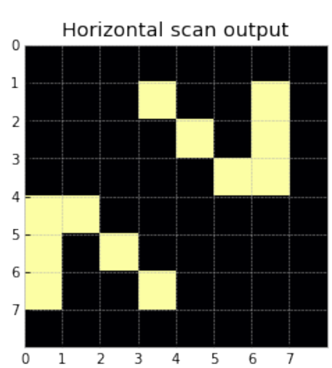

Fig. 5. Result of the horizontal edge detection scan.

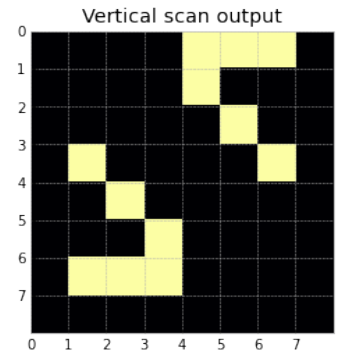

Fig. 6. Result of the vertical edge detection scan.

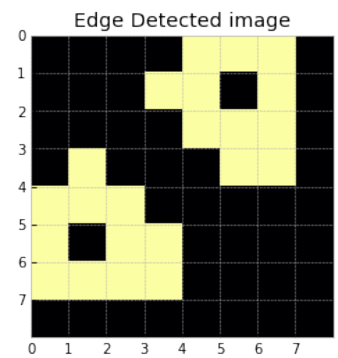

Fig. 7. Final result of the QHED algorithm.

This algorithm was also run on a real quantum computer with inherent noise and error characteristics specific to the hardware. The least busy backend for this run was ibmq_belem from IBM Quantum Lab. Since, running on actual hardware deals with encountering errors due to noise, we only limit this example to run on (2+1)-qubits. The results from the implementation on real hardware of the algorithm will be shown.

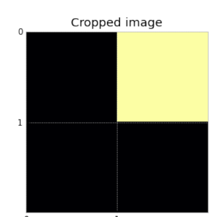

Fig. 8. Cropped image (2×2 matrix).

V. Conclusion

In summary, Quantum Edge Detection is an active area of research that aims to use quantum computing to enhance the performance of classical edge detection methods. Additionally, the field of quantum edge detection is relatively new and there is still a lot of room for improvement and research. The potential benefits of quantum edge detection, such as

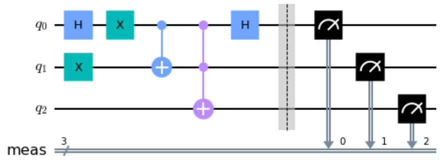

Fig. 9. Circuit for horizontal scan of the image.

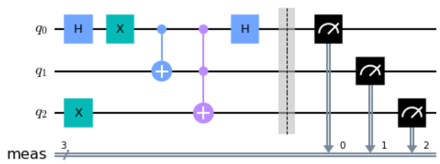

Fig. 10. Circuit for vertical scan of the image.

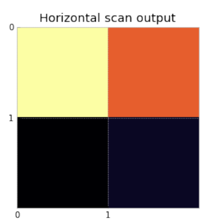

Fig. 11. Result of the horizontal edge detection scan on real hardware.

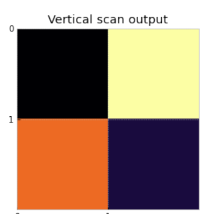

Fig. 12. Result of the vertical edge detection scan on real hardware.

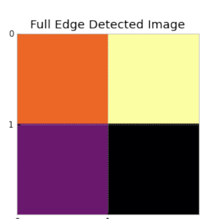

Fig. 13. Final result of the QHED algorithm on real hardware.

improved speed and accuracy, make it an interesting area of research for both academia and industry. Ultimately, QHED must also wait for the development and maturing of quantum hardware, in order for it to become clear if it brings any computational advantages or not. It was demonstrated in this paper how some basic and elementary concepts and operations can be applied to the quantum world to enhance existing edge detection methods. Overall, it seems perfectly reasonable to have optimism and to anticipate future advancements in the use of edge detection techniques.

References

[1] Yao, Xi-Wei, et al. ”Quantum image processing and its application to edge detection: theory and experiment.” Physical Review X 7.3 (2017).

[2] Ruan, Yue, Xiling Xue, and Yuanxia Shen. ”Quantum Image Processing: Opportunities and Challenges.” Mathematical Problems in Engineering (2021).

[3] Yan, Fei, Abdullah M. Iliyasu, and Salvador E. Venegas-Andraca. ”A survey of quantum image representations.” Quantum Information Processing 15.1 (2016).

[4] Zhang, Yi, Kai Lu, and YingHui Gao. ”QSobel: a novel quantum image edge extraction algorithm.” Science China Information Sciences 58.1 (2015): 1-13.

[5] Pengao Xu, Zhenxing He, Tianhui Qiu, and Hongyang Ma, ”Quantum image processing algorithm using edge extraction based on Kirsch operator,” Opt. Express 28, 12508-12517 (2020).

[6] Wu C, Huang F, Dai J, Zhou N, ”Quantum SUSAN edge detection based on double chains quantum genetic algorithm.” Physica A: Stat Mech its Appl (2022).

[7] M. S. ANIS, H. Abraham, AduOffei, R. Agarwal, G. Agliardi, M. Aharoni, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, M. Amy, S. Anagolum, E. Arbel, A. Asfaw, A. Athalye, A. Avkhadiev, C. Azaustre, P. BHOLE, A. Banerjee, S. Banerjee, W. Bang, A. Bansal, P. Barkoutsos, A. Barnawal, G. Barron, G. S. Barron, L. Bello, Y. Ben-Haim, M. C. Bennett, D. Bevenius, D. Bhatnagar, A. Bhobe, P. Bianchini, L. S. Bishop, C. Blank, S. Bolos, S. Bopardikar, S. Bosch, S. Brandhofer, Brandon, S. Bravyi, N. Bronn, Bryce-Fuller, D. Bucher, A. Burov, F. Cabrera, P. Calpin, L. Capelluto, J. Carballo, G. Carrascal, A. Carriker, I. Carvalho, A. Chen, C.-F. Chen, E. Chen, J. C. Chen, R. Chen, F. Chevallier, K. Chinda, R. Cholarajan, J. M. Chow, S. Churchill, CisterMoke, C. Claus, C. Clauss, C. Clothier, R. Cocking, R. Cocuzzo, J. Connor, F. Correa, A. J. Cross, A. W. Cross, S. Cross, J. CruzBenito, C. Culver, A. D. Corcoles-Gonzales, N. D, S. Dague, T. E. ´ Dandachi, A. N. Dangwal, J. Daniel, M. Daniels, M. Dartiailh, A. R. Davila, F. Debouni, A. Dekusar, A. Deshmukh, M. Deshpande, D. Ding, J. Doi, E. M. Dow, E. Drechsler, E. Dumitrescu, K. Dumon, I. Duran, K. EL-Safty, E. Eastman, G. Eberle, A. Ebrahimi, P. Eendebak, D. Egger, ElePT, Emilio, A. Espiricueta, M. Everitt, D. Facoetti, Farida, P. M. Fernandez, S. Ferracin, D. Ferrari, A. H. Ferrera, R. Fouilland, ´ A. Frisch, A. Fuhrer, B. Fuller, M. GEORGE, J. Gacon, B. G. Gago, C. Gambella, J. M. Gambetta, A. Gammanpila, L. Garcia, T. Garg, S. Garion, J. Garrison, T. Gates, L. Gil, A. Gilliam, A. Giridharan, J. Gomez-Mosquera, Gonzalo, S. de la Puente Gonzalez, J. Gorzinski, ´ I. Gould, D. Greenberg, D. Grinko, W. Guan, J. A. Gunnels, H. Gupta, N. Gupta, J. M. Gunther, M. Haglund, I. Haide, I. Hamamura, O. C. ¨ Hamido, F. Harkins, K. Hartman, A. Hasan, V. Havlicek, J. Hellmers, Ł. Herok, S. Hillmich, H. Horii, C. Howington, S. Hu, W. Hu, J. Huang, R. Huisman, H. Imai, T. Imamichi, K. Ishizaki, Ishwor, R. Iten, T. Itoko, A. Ivrii, A. Javadi, A. Javadi-Abhari, W. Javed, Q. Jianhua, M. Jivrajani, K. Johns, S. Johnstun, Jonathan-Shoemaker, JosDenmark, JoshDumo, J. Judge, T. Kachmann, A. Kale, N. Kanazawa, J. Kane, Kang-Bae, A. Kapila, A. Karazeev, P. Kassebaum, J. Kelso, S. Kelso, V. Khanderao, S. King, Y. Kobayashi, Kovi11Day, A. Kovyrshin, R. Krishnakumar, V. Krishnan, K. Krsulich, P. Kumkar, G. Kus, R. LaRose, E. Lacal, R. Lambert, H. Landa, J. Lapeyre, J. Latone, S. Lawrence, C. Lee, G. Li, J. Lishman, D. Liu, P. Liu, Y. Maeng, S. Maheshkar, K. Majmudar, A. Malyshev, M. E. Mandouh, J. Manela, Manjula, J. Marecek, M. Marques, K. Marwaha, D. Maslov, P. Maszota, D. Mathews, A. Matsuo, F. Mazhandu, D. McClure, M. McElaney, C. McGarry, D. McKay, D. McPherson, S. Meesala, D. Meirom, C.

Mendell, T. Metcalfe, M. Mevissen, A. Meyer, A. Mezzacapo, R. Midha, D. Miller, Z. Minev, A. Mitchell, N. Moll, A. Montanez, G. Monteiro, M. D. Mooring, R. Morales, N. Moran, D. Morcuende, S. Mostafa, M. Motta, R. Moyard, P. Murali, J. Muggenburg, T. NEMOZ, D. Nadlinger, K. Nakan- ¨ ishi, G. Nannicini, P. Nation, E. Navarro, Y. Naveh, S. W. Neagle, P. Neuweiler, A. Ngoueya, J. Nicander, Nick-Singstock, P. Niroula, H. Norlen, NuoWenLei, L. J. O'Riordan, O. Ogunbayo, P. Ollitrault, T. Onodera, R. Otaolea, S. Oud, D. Padilha, H. Paik, S. Pal, Y. Pang, A. Panigrahi, V. R. Pascuzzi, S. Perriello, E. Peterson, A. Phan, F. Piro, M. Pistoia, C. Piveteau, J. Plewa, P. Pocreau, A. Pozas-Kerstjens, R. Pracht, M. Prokop, V. Prutyanov, S. Puri, D. Puzzuoli, J. Perez, ´ Quant02, Quintiii, I. R, R. I. Rahman, A. Raja, R. Rajeev, N. Ramagiri, A. Rao, R. Raymond, O. Reardon-Smith, R. M.-C. Redondo, M. Reuter, J. Rice, M. Riedemann, Rietesh, D. Risinger, M. L. Rocca, D. M. Rodr´ıguez, RohithKarur, B. Rosand, M. Rossmannek, M. Ryu, T. SAPV, N. R. C. Sa, A. Saha, A. Ash-Saki, S. Sanand, M. Sandberg, H. Sandesara, R. Sapra, H. Sargsyan, A. Sarkar, N. Sathaye, B. Schmitt, C. Schnabel, Z. Schoenfeld, T. L. Scholten, E. Schoute, M. Schulterbrandt, J. Schwarm, J. Seaward, Sergi, I. F. Sertage, K. Setia, F. Shah, N. Shammah, R. Sharma, Y. Shi, J. Shoemaker, A. Silva, A. Simonetto, D. Singh, P. Singh, P. Singkanipa, Y. Siraichi, Siri, J. Sistos, I. Sitdikov, S. Sivarajah, M. B. Sletfjerding, J. A. Smolin, M. Soeken, I. O. Sokolov, I. Sokolov, V. P. Soloviev, SooluThomas, Starfish, D. Steenken, M. Stypulkoski, A. Suau, S. Sun, K. J. Sung, M. Suwama, O. Słowik, H. Takahashi, T. Takawale, I. Tavernelli, C. Taylor, P. Taylour, S. Thomas, K. Tian, M. Tillet, M. Tod, M. Tomasik, C. Tornow, E. de la Torre, J. L. S. Toural, K. Trabing, M. Treinish, D. Trenev, TrishaPe, F. Truger, G. Tsilimigkounakis, D. Tulsi, W. Turner, Y. Vaknin, C. R. Valcarce, F. Varchon, A. Vartak, A. C. Vazquez, P. Vijaywargiya, V. Villar, B. Vishnu, D. Vogt-Lee, C. Vuillot, J. Weaver, J. Weidenfeller, R. Wieczorek, J. A. Wildstrom, J. Wilson, E. Winston, WinterSoldier, J. J. Woehr, S. Woerner, R. Woo, C. J. Wood, R. Wood, S. Wood, J. Wootton, M. Wright, L. Xing, J. YU, B. Yang, D. Yeralin, R. Yonekura, D. Yonge-Mallo, R. Yoshida, R. Young, J. Yu, L. Yu, C. Zachow, L. Zdanski, H. Zhang, C. Zoufal, aeddins ibm, alexzhang13, b63, bartek bartlomiej, bcamorrison, brandhsn, charmerDark, deeplokhande, dekel.meirom, dime10, dlasecki, ehchen, fanizzamarco, fs1132429, gadial, galeinston, georgezhou20, georgios ts, gruu, hhorii, hykavitha, itoko, jessica angel7, jezerjojo14, jliu45, jscott2, klinvill, krutik2966, ma5x, michelle4654, msuwama, ntgiwsvp, ordmoj, sagar pahwa, pritamsinha2304, ryancocuzzo, saswati qiskit, septembrr, sethmerkel, shaashwat, sternparky, strickroman, tigerjack, tsura crisaldo, vadebayo49, welien, willhbang, wmurphy collabstar, yang.luh, and M. Cepulkovskis, "Qiskit: An open-source framework for ˘ quantum computing," 2021.