MI
MOTION IMAGING JOURNAL

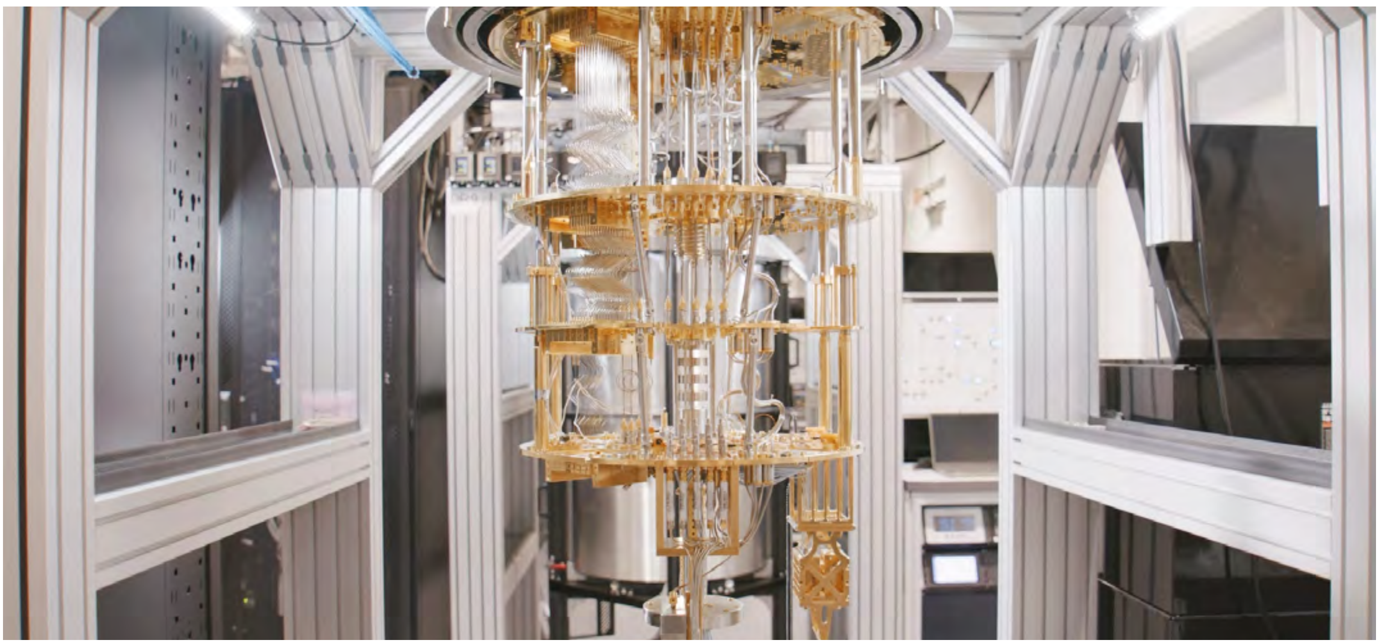
SMPTE

TECHNICAL PAPER

KEYWORDS QUANTUM // VIDEO PROCESSING // NISQ // NEQR // EFFICIENCY

Como leitor, gosto sempre de ser desafiado! E o artigo desta edição é um grande desafio. Nele são apresentados os fundamentos de superposição e emaranhamento da computação quantica, e explica como estas propriedades permitem avaliar múltiplas possibilidades em paralelo. O texto aborda o estágio atual dos computadores (*Noisy Intermediate-Scale Quantum* -NISQ), acessíveis via nuvem, e explora representações quânticas de imagens, como o modelo NEQR, que viabiliza novas abordagens para compressão, detecção de bordas, escalonamento e até processamento em vídeo. Mesmo que ainda seja uma tecnologia em evolução, o potencial de aceleração é claro. Vale a leitura para quem quer antecipar o próximo salto no broadcast e mídia digital.

Tom Jones Moreira
(tvdigitalbr@gmail.com)



Video Processing on Quantum Computers

By Thomas Edwards

Image above: A dilution refrigerator housing a superconducting-qubit quantum chip developed and manufactured at the AWS Center for Quantum Computing in Pasadena, CA. Credit: AWS

Quantum algorithms can provide advantages over classical algorithms by using superposition and entanglement. Superposition allows quantum algorithms to evaluate many possibilities in parallel, unlike classical computers, which evaluate possibilities sequentially. Entanglement allows changes in one qubit to immediately affect the other qubits with which it is entangled.

Abstract
Quantum computing is a multidisciplinary field comprising aspects of computer science, physics, and mathematics that utilize quantum mechanics to solve complex problems faster than on classical computers. Quantum computers are available today on the cloud. However, they are considered "Noisy Intermediate-Scale Quantum" (NISQ) computers with a small number of quantum bits (qubits) and limited performance due to short coherence time and noisy gates. However, quantum computers are constantly improving, and it is possible that they could accelerate video processing workflows in the future. This article will give a short overview of quantum computing basics, some methods for representing images in qubits, and describe some research on potential video applications of quantum computing.

Quantum computers promise to provide a revolutionary way of processing data using a set of quantum algorithms that provide substantial speed-up over "classical" computing. At the current time, quantum computers exist, which are considered "Noisy Intermediate-Scale Quantum" (NISQ) computers, which have a limited computation capacity due to having a small number of quantum bits with short coherence time (how long a qubit can maintain its quantum state) and noisy processing. However, quantum computers are improving with time and can now even be accessed over the Internet in the cloud via mechanisms such as Amazon Braket.[1] This paper gives a brief overview of quantum computer processing and then addresses how quantum computing might be applied to image and video processing in the future.

What is "Quantum"?

In the early 1900s, some strange things were found in the physics of very small sizes and energies. In 1900, Max Planck discovered that the spectrum of light emitted by heated objects (known as "blackbody radiation") made sense only if light energy was emitted or absorbed in discrete packets he called "quanta" instead of in continuous values. In 1905, Albert Einstein proposed that light must also consist of quantized particles (which we now call "photons") through his analysis of the photoelectric effect. This went against hundreds of years of work by scientists like Christiaan Huygens and Thomas Young, who showed the wave nature of light through mechanisms such as diffraction.

J. J. Thomson discovered in 1897 that cathode rays were not continuous, but instead "corpuscles," or particles. And as the model of the atom emerged to have negatively charged electrons around a dense positively charged core, it became clear that if electrons were particles orbiting the nucleus of an atom, they would radiate away all their energy due to the circular acceleration and then collapse onto the nucleus, but that wasn't happening. In 1924, Louis de Broglie proposed that electrons in an atom were not in actual "orbits" but instead were constrained to be standing waves around the nucleus.

Two formulations combining the wave and particle natures of photons and electrons were soon developed. In 1925, Werner Heisenberg published "matrix mechanics," and in 1926, Erwin Schrödinger published "wave mechanics." These theories reproduced the observed absorption spectrum of hydrogen, and the wave nature of electrons was clearly shown in 1927 when Clinton Davisson and Lester Germer of Bell Labs showed a diffraction pattern of an electron beam.

So light, thought of as waves, was now also particles, and electrons, thought of as particles, were now also waves. Moreover, a range of other physical phenomena at the small scale also existed simultaneously in waves and particles, all well described by the quantum mechanics equations.

Quantum mechanics states that systems have "wave functions" over space and time, which can have a real and an imaginary component. The square of the wave function represents the probability of finding a particle in a particular state and location in space. And the very act of trying to measure the location of a quantum causes the wave function of probability to "collapse" into a specific real value.

Quantum Computing 101

For a great introduction to quantum computing, the author recommends "Quantum Country," an interactive online book by Andy Matuschak and Michael Nielsen (at https://quantum.country/).

There are two main divisions of quantum computers. Gate-based quantum computers use a sequence of quantum logic gates to perform computations, while "Adiabatic" quantum computers use continuous quantum state evolution to

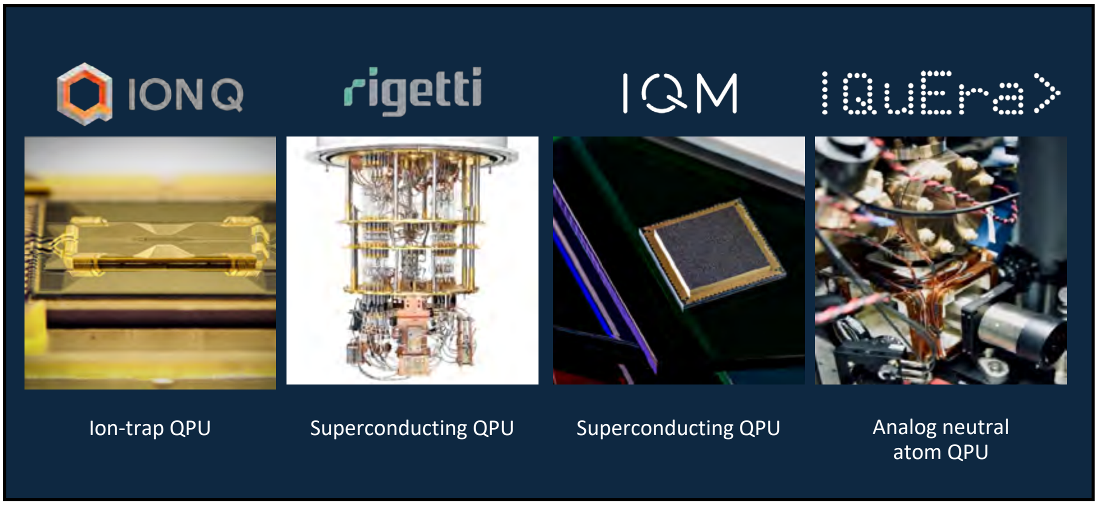


FIGURE 1: Some types of quantum computers on Amazon Braket.

solve optimization problems. This paper concentrates on gate-based quantum computers, as they allow more generalized computations.

Quantum bits, or qubits, are the data elements of quantum computers. A range of quantum systems, such as the spin of an electron, electron orbital energy levels, or photon polarization, can implement them. These systems are implemented on many substrates, such as superconducting loops, simulants of electron orbitals in atoms, electromagnetically trapped ions, optically trapped neutral atoms, or defects in solid crystals. **Figure 1** shows a sample of quantum computer types available on the Amazon Braket service on AWS.

The Greek letter psi (ψ) typically denotes the quantum wave function. A quantum state can be represented as a linear combination of basis states in a vector. "Ket" notation makes it a bit simpler to talk about quantum states without writing out an entire vector. A qubit can take on a linear combination of the two basis states reflecting a real "0" or "1" bit, $|0\rangle$ and $|1\rangle$, as seen below.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The quantum state of a qubit can then be expressed as a linear combination, known as a "superposition," of these basis states:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Where α and β are complex numbers called probability amplitudes. The square of the absolute value of these probability amplitudes (i.e., $|\alpha|^2$ and $|\beta|^2$) gives the probability of the system being measured in the corresponding basis state.

The coefficients α and β must satisfy the normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1$$

TABLE 1. Three Example Quantum Logic Gates.

Gate Name	Symbol	Matrix Representation
Hadamard	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X (aka NOT)	$\oplus$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Controlled NOT (aka CNOT)	$\bullet$ / $\oplus$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

This ensures that the total probability of finding the system in any state is 1.

The concept of superposition allows quantum systems to exist in states that combine multiple classical states simultaneously. Furthermore, the quantum state of multiple qubits can also be linked together through "entanglement." The qubits can be acted upon by quantum logic gates, which are essentially matrixes that multiply the input quantum state vector, and the result is a new output state vector.

There are many different types of quantum logic gates, but three highly used gates are shown in **Table 1**. These are the "Hadamard" gate (which operates on a single qubit and thus has a 2 × 2 matrix representation to operate on the two coefficients in a single qubit state), the Pauli-X or "NOT" gate, and the "Controlled Not" gate (which operates on 2 qubits, thus has a 4 × 4 matrix representation to operate on the four coefficients of the 2 qubits).

Figure 2 shows a simple quantum circuit with two qubits (represented by the two horizontal paths). The paths moving from left to right represent how quantum gates are applied to the qubits over time. The two qubits are initialized at the beginning (e.g., on the left) in the $|0\rangle$ state (i.e., for the quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with α=1 and β=0). The top qubit is then operated on by the Hadamard gate, which with input state $|0\rangle$ has an output state of an equal superposition of $|0\rangle$

and $|1\rangle$, that is $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. When α and β are squared to find the probability of a $|0\rangle$ or $|1\rangle$ measurement of the qubit after the Hadamard gate, the probability is 1/2 (50%) for $|0\rangle$ and 1/2 (50%) for $|1\rangle$.

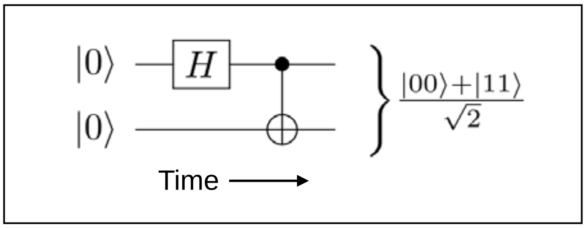


FIGURE 2. Quantum circuit to produce example entangled state (Bell State).

The next gate applied is the CNOT. The graphical representation is that the filled-in circle is the control input qubit, and the open circle with a cross is the qubit that will be acted upon. If the control input is $|1\rangle$, the other qubit will be changed from the $|0\rangle$ to $|1\rangle$ state, or from the $|1\rangle$ to the $|0\rangle$ state, like the classical NOT gate. However, that is a simplification, as the input qubits could be a superposition of the $|0\rangle$ and $|1\rangle$ states, so the proper way to evaluate the gate output is to multiply the input qubits state vectors by the gate matrix.

The output state of the entire circuit is shown on the right of **Fig. 2**. It implies a 50% probability of measuring the qubits in the $|00\rangle$ state (a shorthand to indicate that both qubits are in the $|0\rangle$ state) and a 50% probability of measuring the qubits in a $|11\rangle$ state. There is a 0% probability of measuring the qubits in the $|01\rangle$ or the $|10\rangle$ state. The two qubits are "entangled" such that if you measure one qubit, you know that you will measure the other qubit with the same state (this particular entangled superposition is known as a "Bell State").

Multiple identical runs and output qubit measurements are required to properly characterize the output of quantum computers to estimate the probabilities of different outcomes. This could mean hundreds or thousands of runs (also known as "shots") to ensure measurements provide a good sample of qubit state probabilities.

Using the Amazon Braket SDK,[2] the Bell State circuit program is defined as:

```
import boto3
from braket.aws import AwsDevice
from braket.devices import LocalSimulator
from braket.circuits import Circuit

# create the circuit

bell = Circuit().h(0).cnot(0, 1)
```

Then, it can be run on a quantum computer simulator:

```
# instantiate the local simulator
local_sim = LocalSimulator()
# run the circuit
result = local_sim.run(bell, shots=1000).result()
counts = result.measurement_counts
print(counts)
```

With output showing about the same number of $|00\rangle$ and $|11\rangle$ output states counted over the 1000 shots:

```
Counter({'11': 503, '00': 497})
```

The Bell State circuit can also be run on an actual quantum computer. In this case, we use an IonQ Aria 1 ion trap quantum computer with 25 qubits, although we only use 2 qubits in this circuit. The output measurements (**Fig. 3**) over the 1000 shots shows a few "wrong answers" due to decoherence and noise, but most of the outputs measure $|00\rangle$ and $|11\rangle$.

```
1000 shots taken on machine arn:aws:braket:us-east-1::device/qpu/ionq/Aria-1.
Measurement counts: Counter({'11': 488, '00': 485, '01': 18, '10': 9})
```

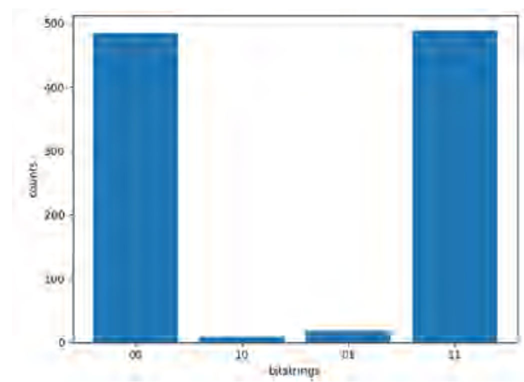


FIGURE 3: Measurement statistics from Aria quantum computer of Bell State Circuit.

Quantum Algorithms and Applications

Quantum algorithms can provide advantages over classical algorithms by using superposition and entanglement. Superposition allows quantum algorithms to evaluate many possibilities in parallel, unlike classical computers, which evaluate possibilities sequentially. Entanglement allows changes in one qubit to immediately affect the other qubits with which it is entangled.

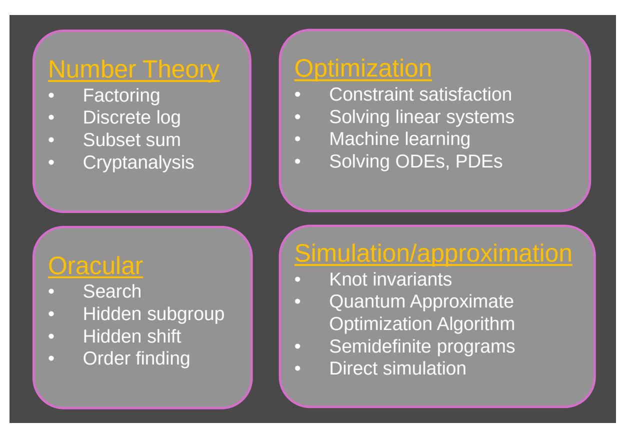


FIGURE 4. Quantum algorithms under investigation.

Quantum algorithms promise substantial speed-ups over classical systems, making it possible to solve problems that would be impractical or impossible for classical computers. **Figure 4** shows areas under investigation for quantum algorithms. Some particularly famous algorithms are Shor's Algorithm, Grover's Algorithm, and the quantum Fourier transform.

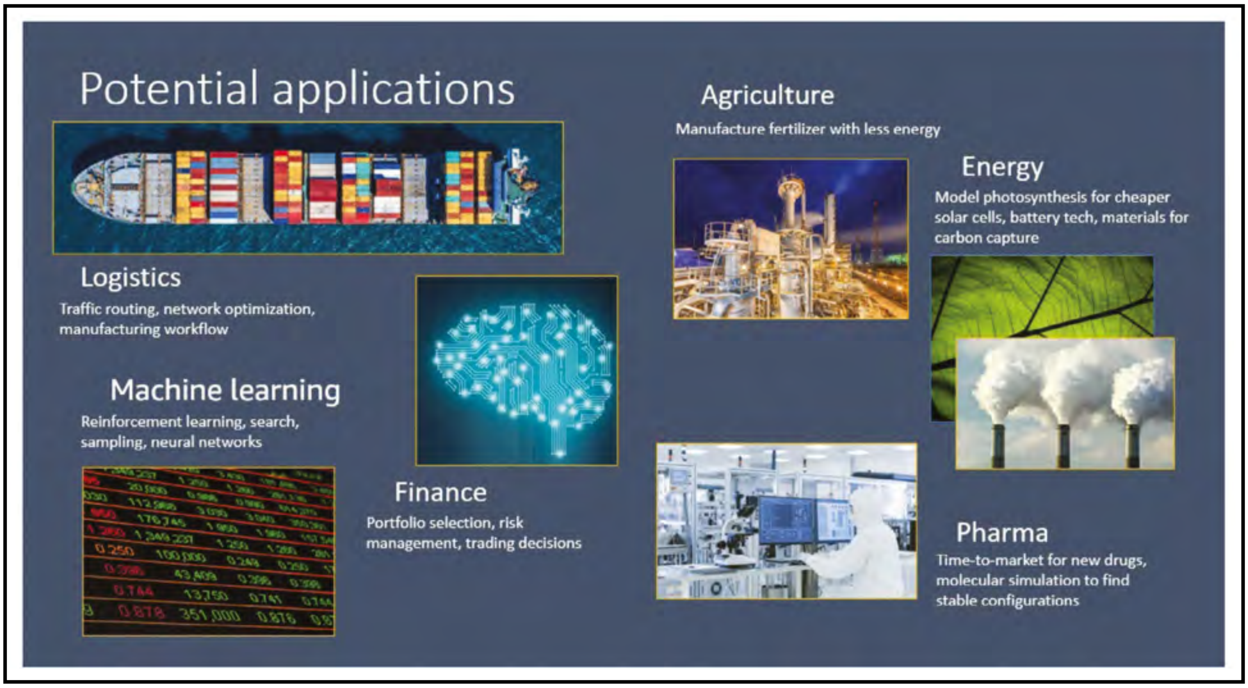


FIGURE 5. Potential applications for quantum computing.

Peter Shor developed a quantum algorithm that allows factorization in polynomial time rather than exponential time achieved using classical algorithms.[3] One potential practical application for Shor's algorithm is to break RSA cryptosystems.[4] While no existing quantum computer can break any practically sized RSA system, the potential for future quantum computers to do so has led to the design of new "quantum-safe" cryptosystems not dependent on factorization.

Grover's algorithm is a quantum algorithm for unstructured search,[5] such as finding a particular element in an unordered list of data. On a classical computer, an exhaustive search would take $O(n)$ operations. Grover's algorithm achieves a quadratic speedup; it can search the n possibilities in only $O(\sqrt{n})$ operations.

The quantum Fourier transform (QFT) is important for image processing,[6] a quantum algorithm that efficiently performs the Fourier transform on a quantum computer. Shor's factoring algorithm uses the QFT internally. QFT also provides an exponential speed-up over its classical equivalent.

Figure 5 shows some areas of potential applications of quantum computing.

Quantum Image Representation

The first step in quantum image processing is determining how to represent images on a quantum computer. Several representations have been proposed, such as the Qubit Lattice,[7] Real Ket,[8] entangled image,[9] and Flexible Representation of Quantum Image (FRQI).[10] Since the Novel Enhanced Quantum Representation (NEQR) method[11] is easily understood, it is presented next.

The pixel data of an image with 2^q-bit color information with color bits ($C^0 \ldots C^{q-1}$) at pixel location (Y,X) is defined as:

$$f(Y,X) = C^0_{YX} C^1_{YX} \ldots C^{q-2}_{YX} C^{q-1}_{YX},\ C^k_{YX} \in [0,1],\quad f(Y,X) \in [0, 2^q - 1]$$

The NEQR quantum representation of a 2^n x 2^n pixel image is:

$$|I\rangle = \frac{1}{2^n} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |f(Y,X)\rangle |YX\rangle$$

In plain terms, enough qubits are required to represent the bits of color in each pixel and the bits of the X and Y locations in the image. For example, a 24-bit color pixel in a 1080p image needs 24 qubits for the color and 22 qubits for the Y and X coordinates (since 2^{11}=2048, 11 is enough bits to handle either 1920 columns or 1080 lines) for a total of 46 qubits.

Figure 6 shows a 2 × 2 pixel NEQR image with 2-bit grayscale, along with the quantum representation of the image on 4 qubits. The tensor product operator ⊗ combines the two quantum states, and in Ket notation $|01\rangle \otimes |01\rangle$ becomes $|0101\rangle$.

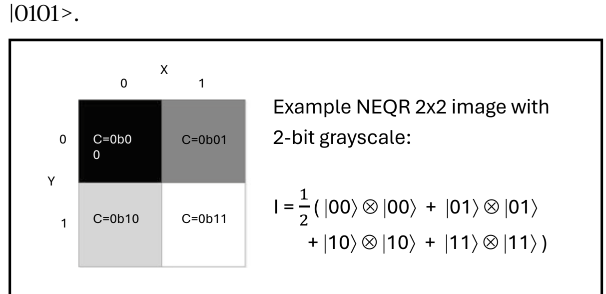


FIGURE 6. NEQR representation of a sample image.

The quantum circuit in **Fig. 7** shows how to load these pixels into the image (The circuit diagram was generated using Quirk, which can be accessed live at http://algassert.com/quirk). First (i.e., on the left), all qubits are initialized to state |0⟩. Then Y and X value qubits have the Hadamard gate applied to put them in an equal superposition of state |0⟩ and state |1⟩. Thus, the (Y, X) qubits have a superposition of all possible pixel locations. The color qubits are entangled with the (Y, X) coordinates using Mixed-Polarity Multiple-Control Toffoli (MPMCT) gates. These multi-qubit gates are similar to the CNOT gates in that they have control input qubits that affect the output qubit, although the filled circle inputs must be in state |1⟩ and the open circle inputs must be in state |0⟩ to change the state of the qubit with the crossed circle. Since pixel (0,0) has both bits of color equal to zero, the color qubits do not need to be changed from their initialized state |0⟩. The other pixels have their (Y, X) qubits entangled to form the proper state |1⟩ 's of the color qubits.

Once the qubits are loaded with the image, processing can begin. For example, using the circuit in **Fig. 8** one could invert the grayscale color of all the image pixels by having the color qubits go through quantum NOT gates and then measure all the output qubits (the measurement icon looks like a dial on a multimeter). A grayscale inversion can be done in one time unit on the quantum computer with a number of quantum NOT gates equal to the number of pixel color bits. Of course, reading out the pixel image probabilities precisely takes a number of shots depending on the complexity of the image.

To move this representation from images to video, additional qubits could be added to represent the video's frame number.

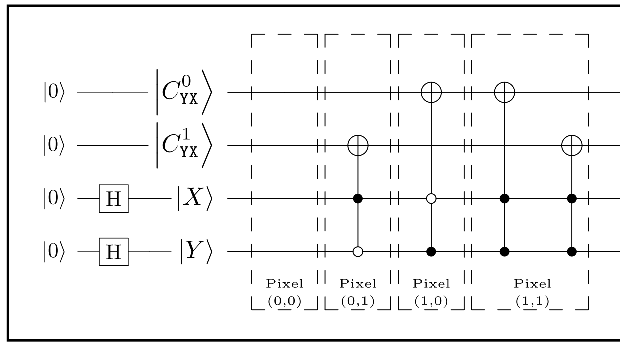


FIGURE 7. Quantum circuit to load pixels into an image.

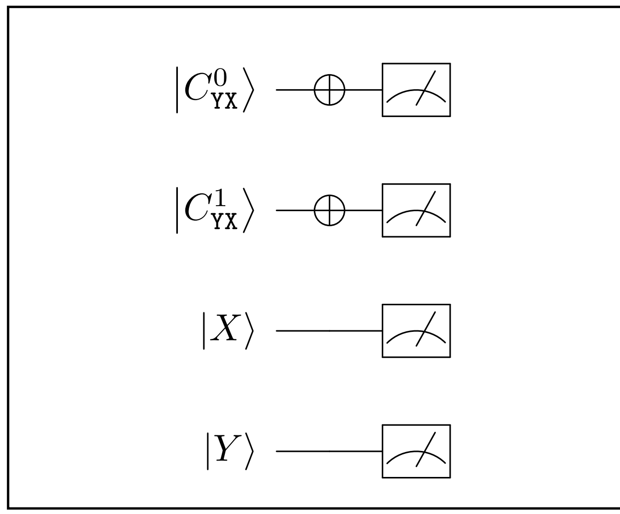


FIGURE 8. Example quantum circuit for grayscale inversion with measurement.

The Potential for Quantum Video Processing

Several quantum image and video processing algorithms have been proposed, for example:

- Motion detection[12]
- Edge detection[13]
- Using the quantum discrete cosine transform for image compression[14]
- Quantum-accelerated fractal image compression[15]
- Image up- and down-scaling[16]
- Image watermarking[17]

For a detailed description of many quantum image and video processing algorithms, refer to the excellent reviews in Refs. 18 and 19.

Image processing using neural networks on classical computers has become very important, and there are several mechanisms to potentially implement neural networks on quantum computers.[20] Also, recent advances in generative AI could be implemented on quantum computers.[21]

Conclusion

Quantum computing is still in its early stages of development. However, every year there are advances in the number of qubits available in quantum computers, and improvements in gate noise and coherence both due to improvements in the physical substrates the quantum systems are built on, and through techniques where several physical qubits are used to create less error-prone "logical qubits" using quantum error-correction coding.[22]

The author is personally reminded of the situation of neural network technology when he was a student in that field in the early 1990s. While we understood most of the algorithms for training neural networks used today, we did not have the computational power to do anything beyond "toy problems," like playing tic-tac-toe or deciding in which of two interlocking spiral-shaped regions a given coordinate lies. Due to remarkable increases in computational power from GPUs, AI accelerator chips like AWS Trainium and AWS Inferentia, and the ability to assemble massive amounts of on-demand cloud computing, neural networks can achieve amazing feats today.

Some years from now, when large non-noisy quantum computers are developed, we can utilize the algorithms presented above to do amazingly fast video processing. Until then, we can connect to the cloud and try out some "toy problems" on existing real quantum computers to prepare for that day.

References

1. Amazon Web Services, "Cloud Quantum Computing Service—Amazon Braket," [Online]. Available: https://aws.amazon.com/braket/
2. Amazon Braket, Braket Python SDK. GitHub. [Online]. Available: https://github.com/amazon-braket/amazon-braket-sdk-python [Accessed: 10 Feb. 2025]
3. P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," *Proc. 35th Annu. Symp. Found. Comput. Sci.* IEEE Computer. Soc. Press. pp. 124-134, 1994. doi:10.1109/sfcs.1994.365700. ISBN 0818665807. S2CID 15291489.
4. V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The Impact of Quantum Computing on Present Cryptography, 2018." [Online]. Available: *arXiv preprint arXiv:1804.00200*. https://arxiv.org/abs/1804.00200
5. L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," *Proc. 28th Annu. ACM Symp. Theory Comput.*, pp. 212-219, May 1996.
6. D. Coppersmith, "An Approximate Fourier Transform Useful in Quantum Factoring," IBM Research Report RC19642, IBM T. J. Watson Research Center: Yorktown Heights, NY, Nov. 1994.
7. S. E. Venegas-Andraca, and S. Bose, "Storing, Processing, and Retrieving an Image using Quantum Mechanics," *Proc. SPIE Quantum Inf. Comput.*, Vol. 5105, pp. 137-147, 2003.
8. J. I. Latorre, "Image Compression and Entanglement." 2005. [Online]. Available: *arXiv preprint quant-ph/0510031*. https://arxiv.org/abs/quant-ph/0510031
9. S. E. Venegas-Andraca, and J. L. Ball, "Processing Images in Entangled Quantum Systems," *Quantum Inf. Process.*, 9(1): 1-11, Jan. 2010.
10. P. Q. Le, F. Dong, and K. Hirota, "A Flexible Representation of Quantum Images for Polynomial Preparation, Image Compression, and Processing Operations," *Quantum Inf. Process.,* 10(1): 63-84, Jan. 2011.
11. Y. Zhang, K. Lu, Y. Gao, and M. Wang, "NEQR: A Novel Enhanced Quantum Representation of Digital Images," *Quantum Inf. Process.*, 12(8): 2833-2860, Aug. 2013.
12. S. Wang, "Frames Motion Detection of Quantum Video," In *Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proc. Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 145-151, Nov. 21-23, 2016, Kaohsiung, Taiwan, Springer International Publishing.
13. X. W. Yao, H. Wang, Z. Liao, M. C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo, and W. Zheng, "Quantum Image Processing and Its Application to Edge Detection: Theory and Experiment," *Physical Review X*, 7(3): 031041, Sep. 2017.
14. C. Y. Pang, Z. W. Zhou, and G. C. Guo, "Quantum Discrete Cosine Transform for Image Compression," 2006. [Online]. Available: *arXiv preprint quant-ph/0601043*. https://arxiv.org/abs/quant-ph/0601043v2
15. S. Du, Y. Yan, Y. and Y. Ma, 2014. "Quantum-Accelerated Fractal Image Compression: An Interdisciplinary Approach," *IEEE Signal Process. Lett.*, 22 (4): 499-503, Apr. 2014.
16. R. G. Zhou, X. Liu, and J. Luo, "Quantum Circuit Realization of the Bilinear Interpolation Method for GQIR," *Int. J. Theor. Phys., 56:2966–2980*, Oct. 2017.
17. P. Li, Y. Zhao, H. Xiao, and M. Cao, "An Improved Quantum Watermarking Scheme Using Small-Scale Quantum Circuits and Color Scrambling," *Quantum Inf. Process.*, 16: 1–34, Jul. 2017.
18. S. Caraiman and V. Manta, "Image Processing Using Quantum Computing," *Proc. 16th Int. Conf. Syst. Theory Control Comput*. (ICSTCC), pp. 1-6, 2012.
19. Z. Wang, M. Xu, and Y. Zhang, "Review of Quantum Image Processing," *Archives of Computational Methods in Engineering*, *29(2)*: 737-761. April 2022.
20. S. K. Jeswal, and S. Chakraverty, "Recent Developments and Applications in Quantum Neural Network: A Review," *Arch. Comput. Methods Eng., 26*: 793-807, Jul. 2019.
21. J. Tian, X. Sun, Y. Du, S. Zhao, Q. Liu, K. Zhang, W. Yi, W. Huang, C. Wang, X. Wu, and M. H. Hsieh, "Recent Advances for Quantum Neural Networks in Generative Learning," *IEEE Trans. Pattern Anal. Mach. Intell., 45(10)*: 12321-12340, Oct. 2023.
22. S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum Error Correction for Beginners," *Reports on Progress in Physics, 76(7)*: 076001, Jul. 2013.

About the Author

Thomas Edwards is Senior Manager, Solutions Architecture for Media & Entertainment at Amazon Web Services. He is a SMPTE Fellow and recipient of the SMPTE Workflow Systems Medal.

SMPTE