Quantum Edge Detection

Jacob Emmerson, Evan Kozierok, Caden Schiffer

University of Pittsburgh

Dr. Gurudev Dutt

PHYS 1470

Abstract: Edge detection is a problem in the field of computer science relating to extracting interesting features from images. Traditional edge detection methods involve scanning through all the pixels of the image to search for changes in light levels between adjacent pixels that signify visual edges and curves in the original image. [3] These methods can be quite slow, particularly as the typical image's resolution increases. This report demonstrates the feasibility of a quantum edge detection algorithm called Quantum Hadamard Edge Detection (QHED). Although we can not easily demonstrate the speedup offered by this quantum method due to hardware limitations, we discuss the theoretical increase in speed provided by QHED and demonstrate that the results are comparable with classical (non-quantum) edge detection methods.

Objectives

The two primary objectives of this study are to demonstrate that QHED has applications for images larger than 64 pixels and discuss the theoretical speedup. An experimental speedup is not possible due to the limitations surrounding this project as well as the current limitations surrounding Quantum algorithms in practice.

Background

Edge detection was proposed as early as the mid 20th century; however, it only recently has become a very active field of research among the machine learning community as well as a commercial product. This is a common trend seen across machine learning fields as computing power has historically been one of the greatest barriers to exploring more advanced machine learning techniques. Early approaches involved convolving images with simple filters in the form of vectors; however, this was a naive approach and was very limited in its ability to detect complex shapes. More modern techniques have further developed the field of computer vision through deep learning-based algorithms, neural networks, and new approaches to convolution and kernels. However, while modern techniques have led to significant developments within the field, there remain several limitations that are actively being researched. These limitations include susceptibility to noise, edge orientation classification, and computing power for real-time analysis. [4]

As we develop more advanced techniques for edge detection as well as mitigate the error that comes with our current limitations, the potential of edge detection increases considerably. Edge detection is foundational to computer vision; it essentially forms the shapes that allows computers to ‘see’. Many modern smartphone cameras utilize edge detection to recognize faces or even landmarks; new applications allow for automated transportation or even streamlined healthcare services. Edge detection has opened an entire new area of machine learning that allows for a whole array of new data that can be trained on, evaluated, and applied in fields beyond just software engineering. [5]

Method

Quantum edge detection, as well as quantum image processing, is a relatively new field that has a theoretical exponential speedup compared to their classical counterparts. Though classical methods are plenty efficient for general image processing, they tend to struggle with higher resolution images due to the high pixel density.

Since a quantum approach is used, the image can be encoded using Quantum Probability Image Encoding (QPIE) which allows the encoding of a $2^{n/2}$ x $2^{n/2}$ image as n-qubits since n-qubits allow for 2^n states in superposition. For this report, a 256x256 image is used; $256 = 2^8$ which means the image can be represented using 16 qubits. Generalizing the encoding process, the image state for n-qubits is the following, where each c_i is the normalized color value of one pixel in the image:

$$|\mathrm{Img}\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle$$

For the edge detection algorithm, a variation of QHED is used. This variation contains an auxiliary qubit that has a Hadamard gate applied to it. The auxiliary qubit allows for simultaneous computation of even and odd pixel-pairs. Prior to computation, the image state gets initialized to $|\mathrm{Img}\rangle = (c_0, c_1, c_2, \ldots, c_{N-2}, c_{N-1})^T$ which produces an (n + 1) qubit redundant image state represented as shown:

$$|\mathrm{Img}\rangle \otimes \frac{(|0\rangle + |1\rangle)}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} c_0 \\ c_0 \\ c_1 \\ c_1 \\ c_2 \\ c_2 \\ \vdots \\ c_{N-2} \\ c_{N-2} \\ c_{N-1} \\ c_{N-1} \end{bmatrix}$$

This image state contains the redundant probability amplitudes. A decrement gate is applied to the image state to rearrange the coefficients make the gradient computation more efficient.

$$D_{2^{n+1}}|\mathrm{Img}\rangle = (c_0, c_1, c_1, c_2, c_2, c_3..., c_{N-2}, c_{N-1}, C_{N-1}, c_0)^T$$

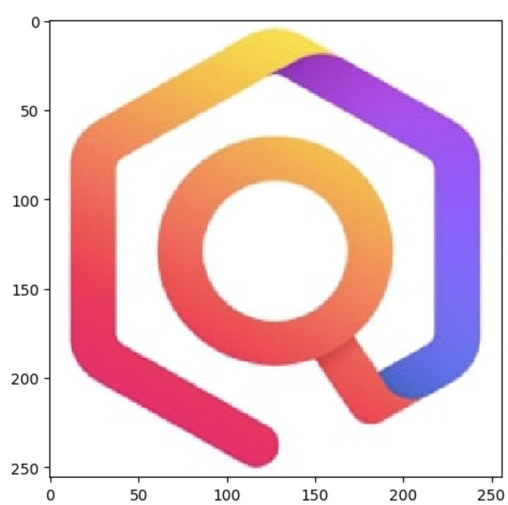
By re-applying a Hadamard gate to the auxiliary qubit, the gradients for all pixel pairs are simultaneously computed. By measuring this state conditioned on the auxiliary qubit being in the state $|1\rangle$, the resulting horizontal gradient values (c_i - c_{i+1}) are found for all pairs of adjacent pixels. This allows us to determine horizontal edges; vertical edges are obtained by transposing the original image (rotating 90 degrees) and repeating the process. Both scans are composed together to form the final results. [1]

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

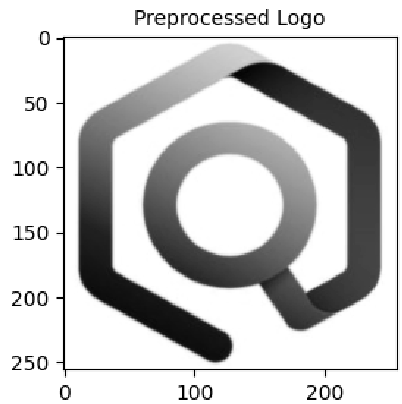
Results

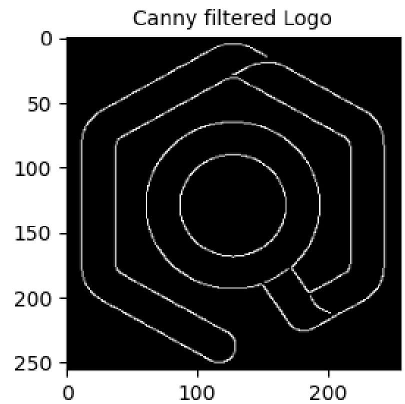
We demonstrate the results of performing QHED on a simple 256x256 logo. We compare this result to a classical algorithm (Canny Edge Detection) performed using scikit-learn.

Base Image (256 x 256, resized in document):

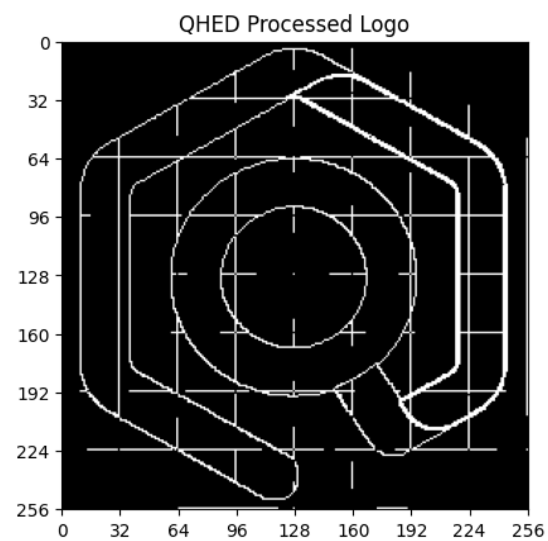


Classical Algorithm (Canny ED, Gaussian Blur Preprocess):





Quantum Algorithm (QHED with Auxiliary Qubit):



Discussion

As is apparent from the results, although our implementation of the QHED algorithm is not as clean as canny edge detection, it is certainly functional. Due to constraints with quantum hardware, we were forced to use a simulator instead of a true quantum computer and thus can't easily demonstrate a practical speedup, but it does provide a substantial theoretical speedup. The actual edge detection process is an O(1) operation simply via applying Hadamard gates and the quantum oracle, although it is a slower $O(n^2)$ operation to set up the quantum states. Measuring the exact probabilities of the resulting state is in theory $O(2^n)$, but in practice we do not need the exact probabilities and can discern a binary value (edge or not) also in $O(n^2)$, for an overall complexity of $O(n^2)$. Canny edge detection on the same $2^{n/2}$ x $2^{n/2}$ image is an $O(2^n \log 2^n) = O(n*2^n)$ operation, so QHED provides an exponential speedup. [1, 2]

Our implementation is not perfect - in addition to the measured speed of the algorithm being slow in practice, our results show noise. In particular, at the edges between chunks, we believe the algorithm “wraps around” to the other side of the chunk. If there is a color difference there, it will detect a non-existent edge, which is the cause of the “gridlines” in the image. Further improvements to the implementation would account for this issue to counteract it.

Conclusion

We have shown that edge detection is a problem that can be solved using a quantum algorithm, and that it is one that can in theory provide major benefits over the classical algorithms we often use. By utilizing the nature of the Hadamard gate and the concise representation of image states, QHED can achieve an exponential speed up compared to industry standard models like Canny Edge Detection. In practice, QHED does perform as it is advertised, although the results shown here are preliminary and not as polished as the current leading ML libraries. Further work can be done to increase the capabilities of quantum hardware and to clean up irrelevant noise in the QHED results.

References

1. “Quantum Edge Detection.” *Qiskit*, https://learn.qiskit.org/course/ch-applications/quantum-edge-detection.
2. Katiyar, Sunil Kumar, and P. V. Arun. "Comparative analysis of common edge detection techniques in context of object extraction." arXiv preprint arXiv:1405.6132 (2014). https://arxiv.org/abs/1405.6132
3. “Edge Detection.” *Mathworks*, https://www.mathworks.com/discovery/edge-detection.html.
4. Raha, Ritwik. “A Brief History of Edge Detection.” *Medium*, https://medium.com/@ritwikraha.nsec/a-brief-history-of-edge-detection-b2008f2ff3c4.
5. Gül, Rabia. “What Are The Applications of Edge Detection?” *Cameralyze*, https://www.cameralyze.co/blog/what-are-the-applications-of-edge-detection.