NATIONAL RESEARCH UNIVERSITY

QUANTUM HADAMARD EDGE DETECTION ALGORITHM

Galochklin Boris

Nizhniy Novgorod, 2021

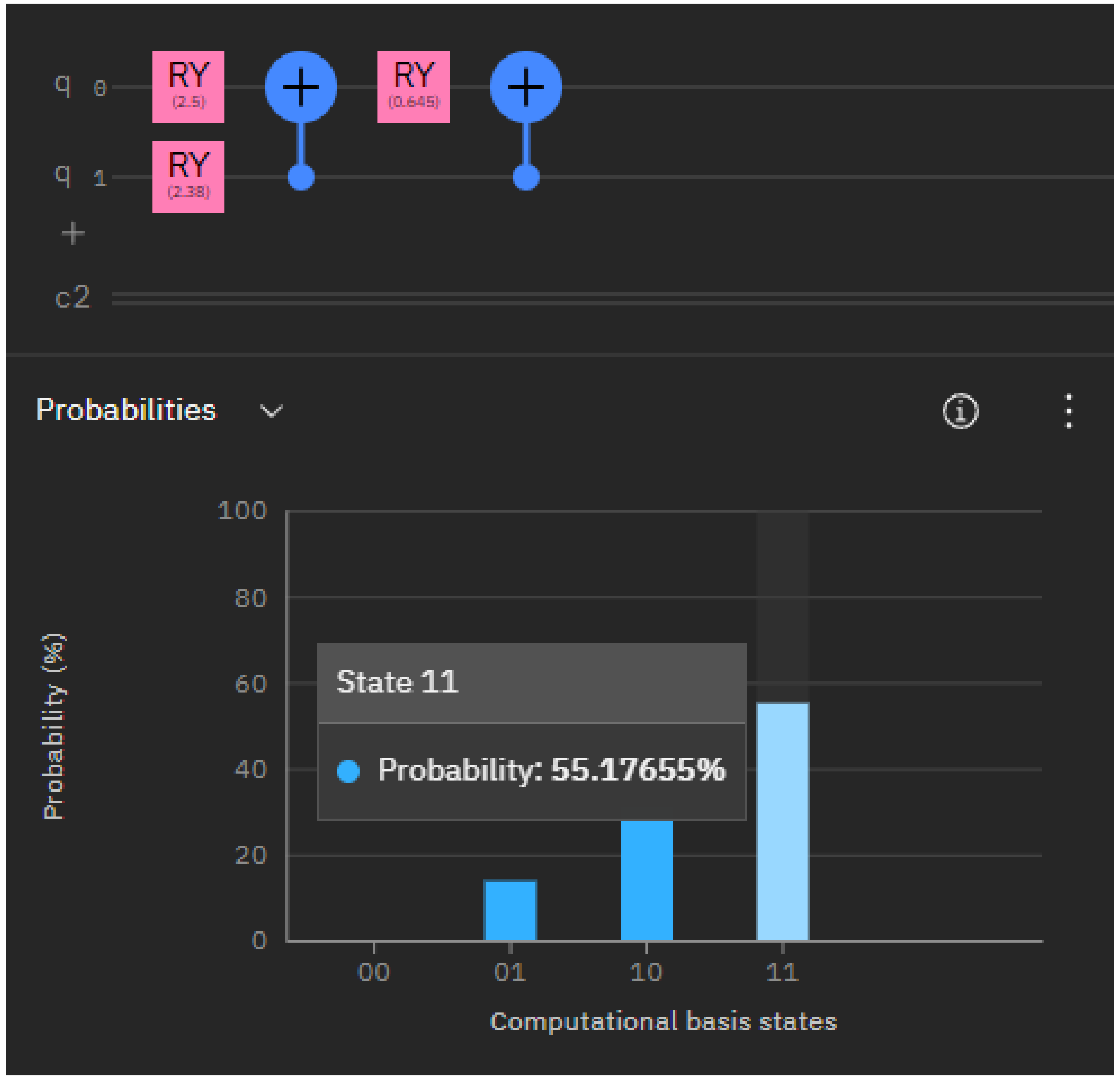
QUANTUM PROBABILITY IMAGE ENCODING

Image encoding conclusions

- Do not require many qubits for big images processing, N qubits for 2^N image sizes
- Takes some time for pre-processing, one circuit – one image.
- Can be done by framework using `initialize(amplitude)` method.

```
img = np.array([0,128,192,255])
rms = np.sqrt(np.sum(np.sum(img**2)))
i2norm = np.array([x / rms for x in img])
i2norm**2
```

```
array([0.        , 0.13852697, 0.31168568, 0.54978736])
```



QUANTUM HADAMARD EDGE DETECTION

Hadamard gate

The Hadamard gate H has the following operation on the state of qubit,

$$|0\rangle \rightarrow \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}$$

$$|1\rangle \rightarrow \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$$

The QHED algorithm generalizes this action of H-gate and uses it for edge detection of an image.

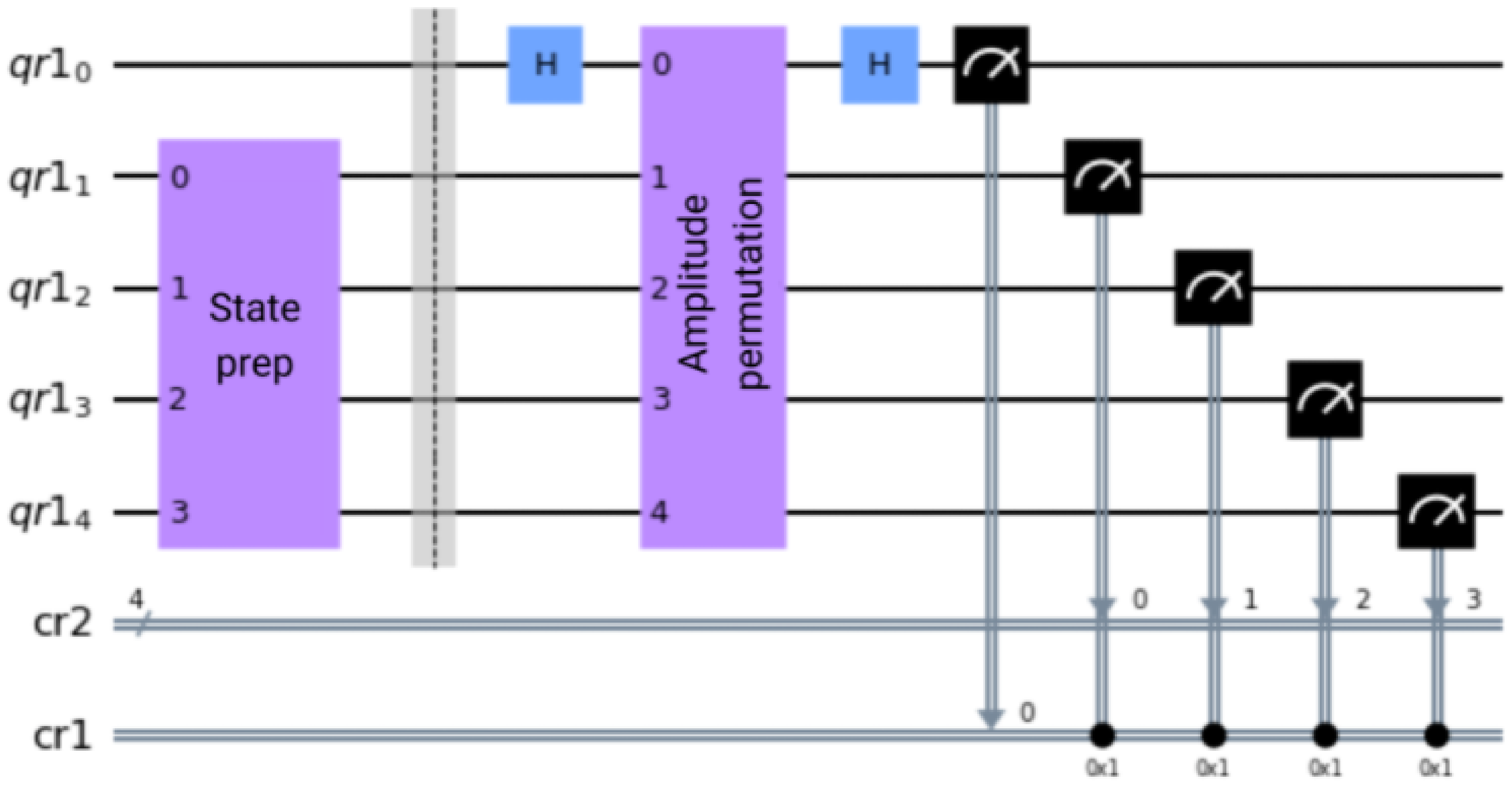
quantum register, we can represent the resultant unitary like,

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

QUANTUM HADAMARD EDGE DETECTION

Circuit

The QHED quantum circuit for the above image can be generalized as:



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

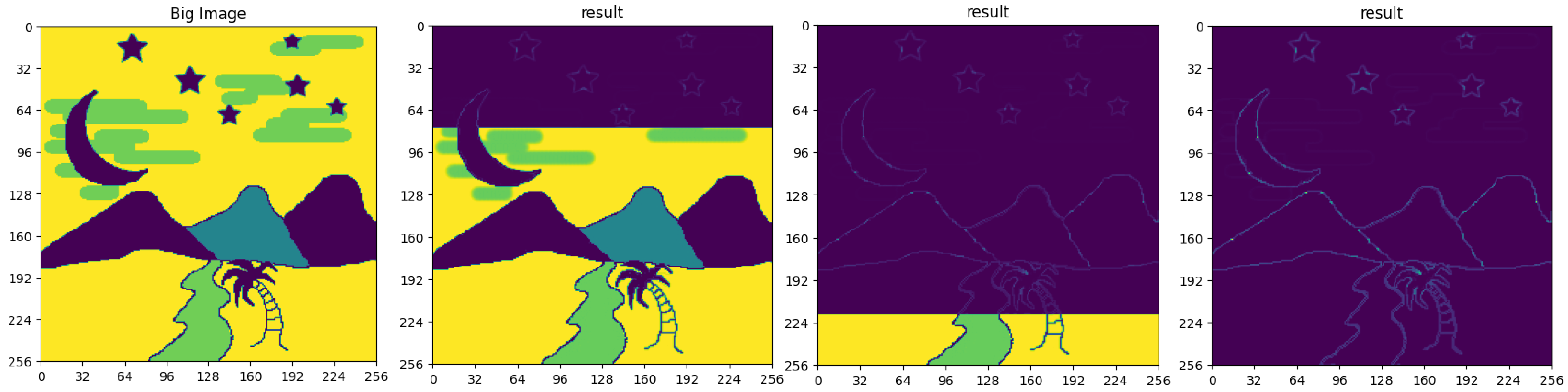
QUANTUM HADAMARD EDGE DETECTION

Time and space complexity

- Time complexity of image encoding is still O(n^2)
- Depth of circuit O(poly(n))
- Algorithm complexity without including state-preparation and amplitude permutation O(1)
- Number of measurements O(n^2)

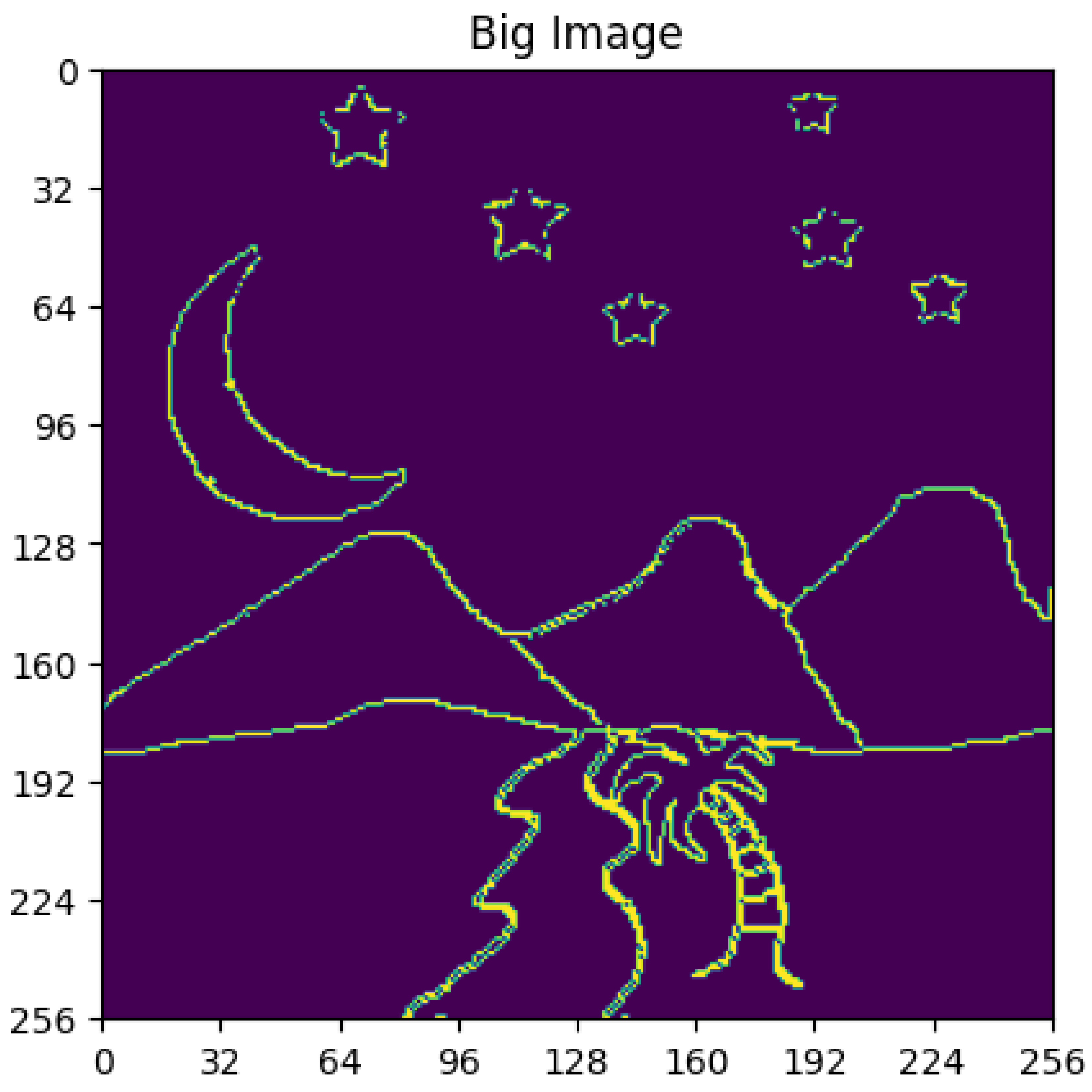
EXPERIMENTS RESULTS

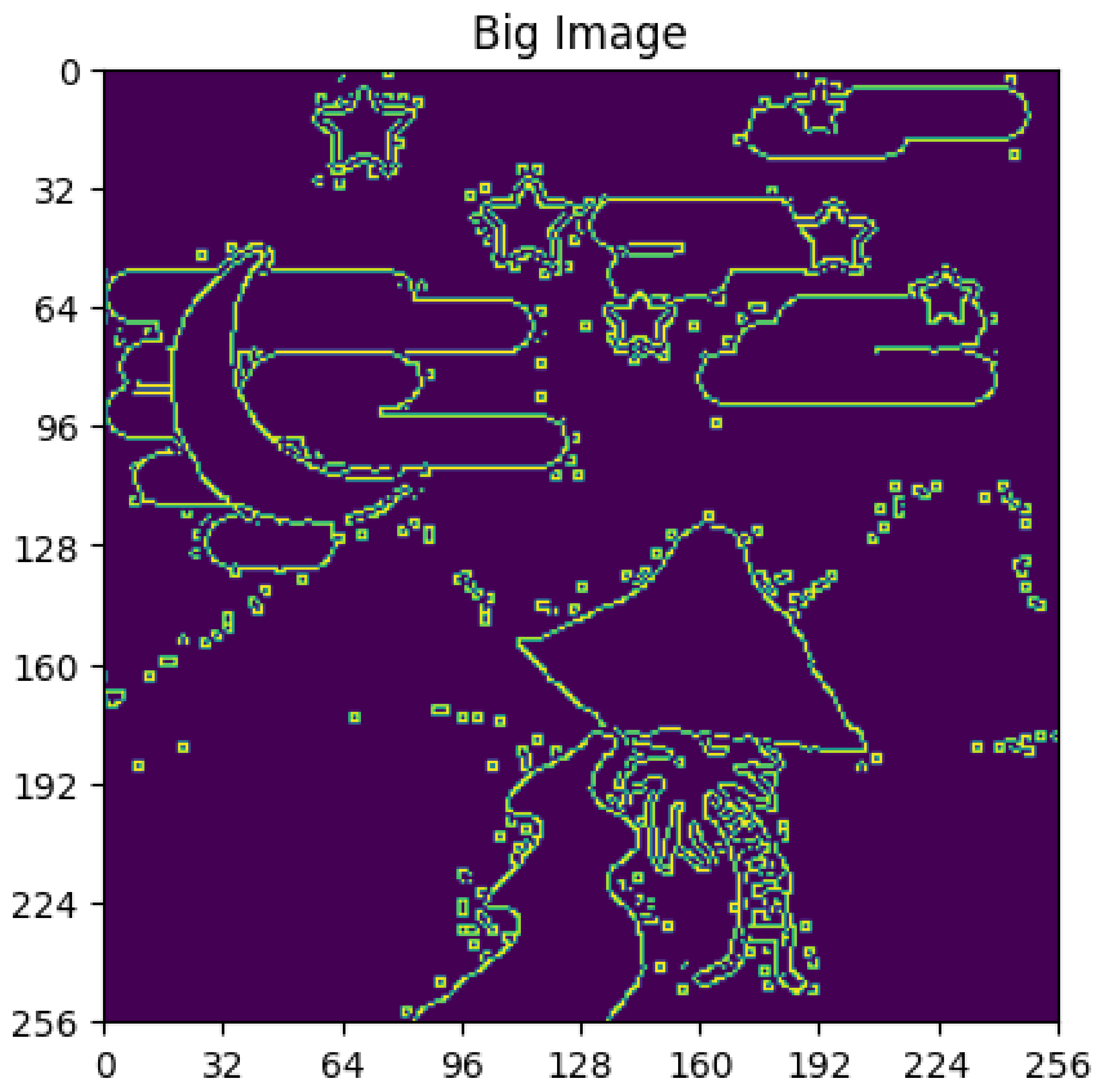
Image 256x256, simulator



EXPERIMENTS RESULTS

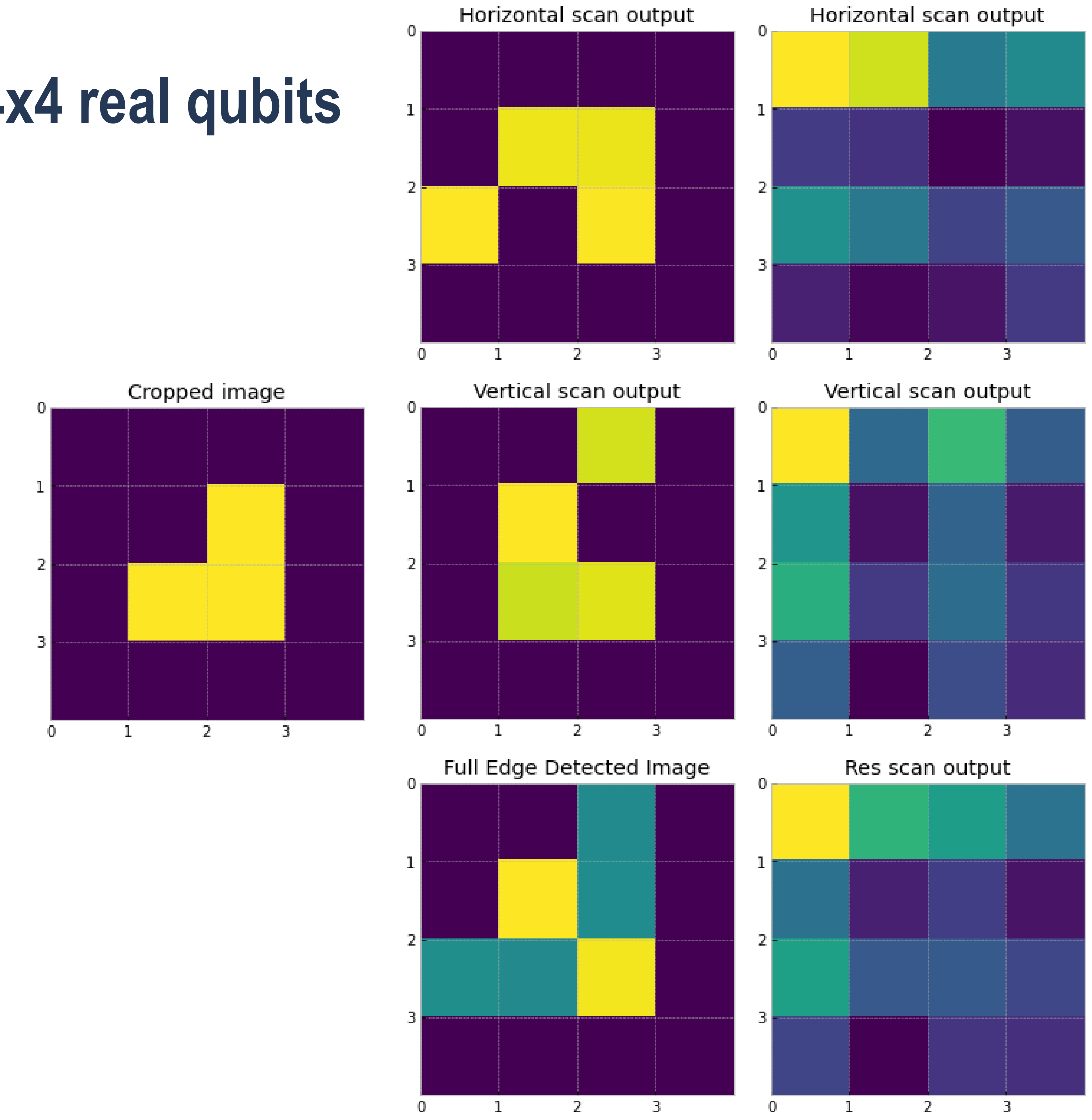
Image 256x256, simulator vs cv2.Canny

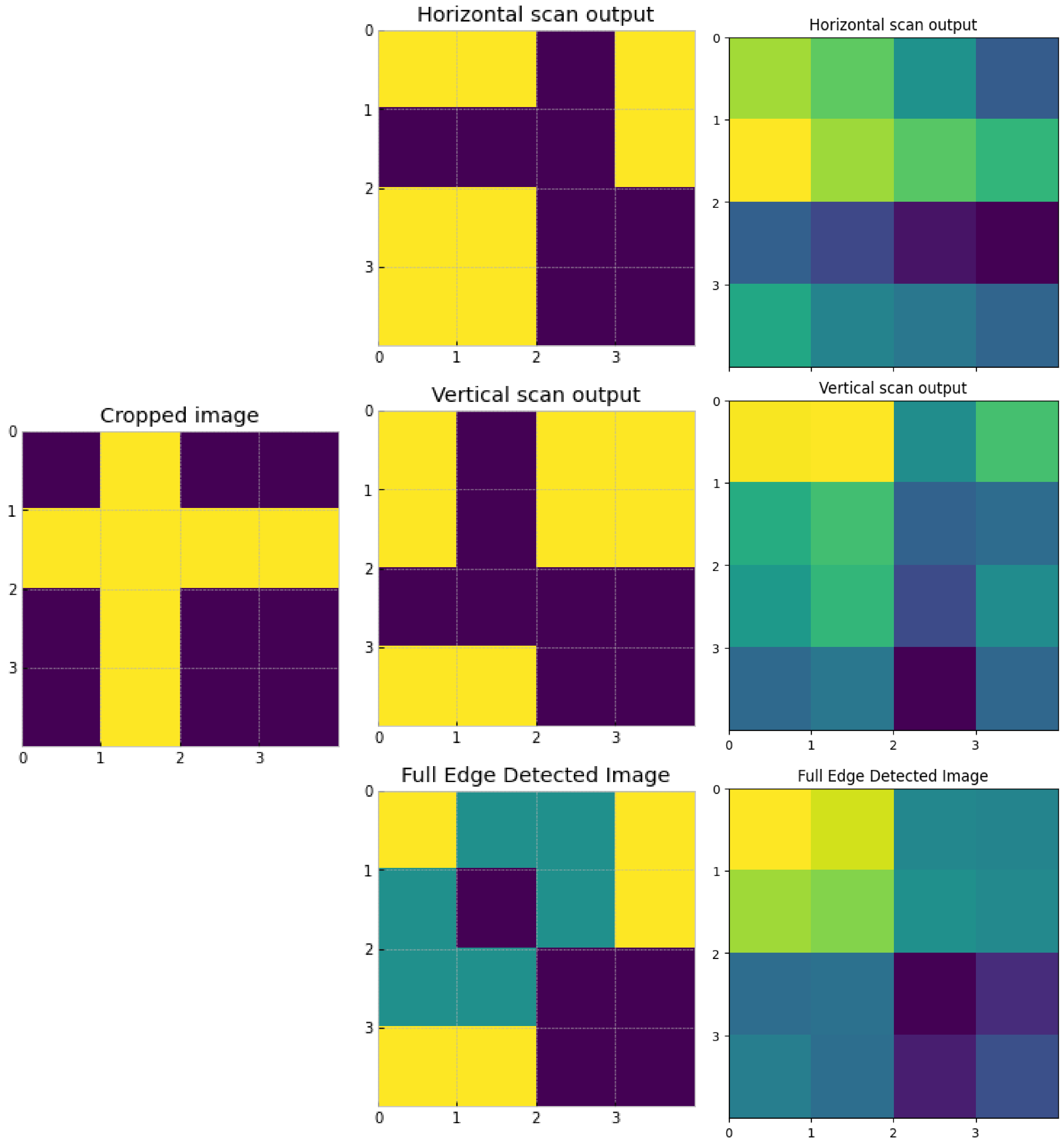




EXPERIMENTS RESULTS

Image 4x4 real qubits





EXPERIMENTS RESULTS

Conclusions

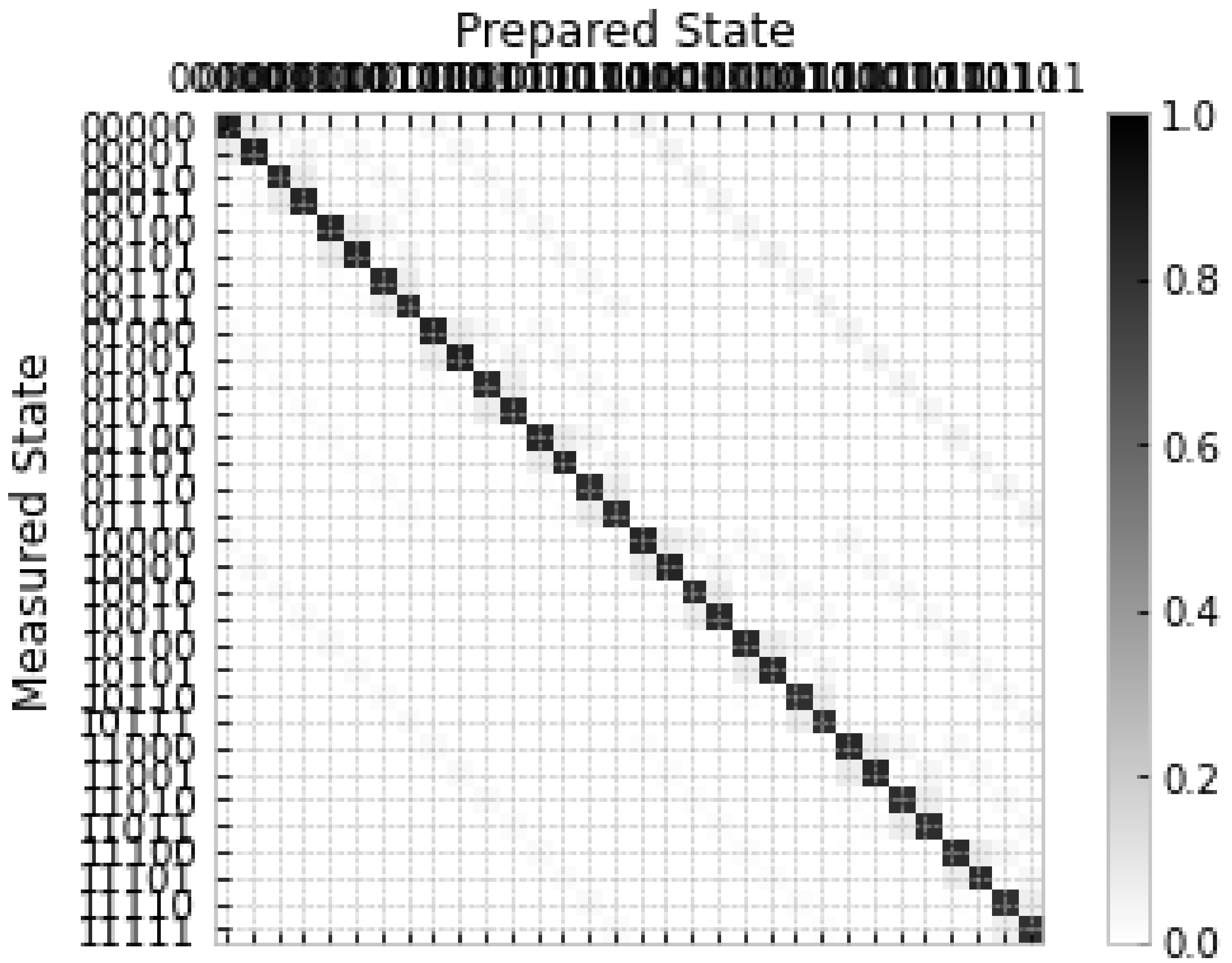
- Results from real HW are noisy, mostly because of huge depth of circuit ~3k.
- Only 5 qubits available, average execution time ~15min because of long queues.
- Image processing task is independent for each chunk of image, can by submitted as a batch of circuits, 2 circuits per chunk.

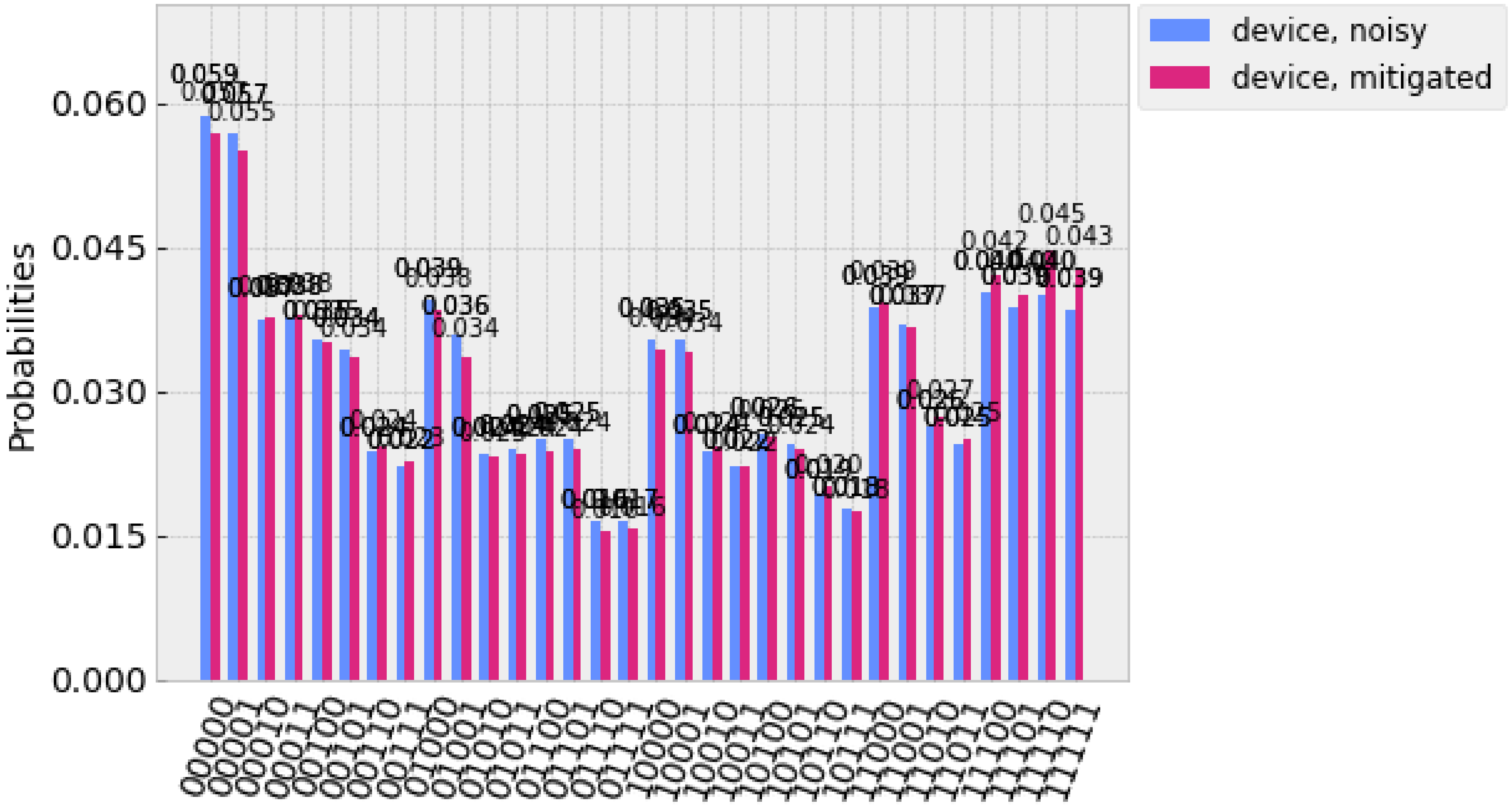
NOISE MITIGATION

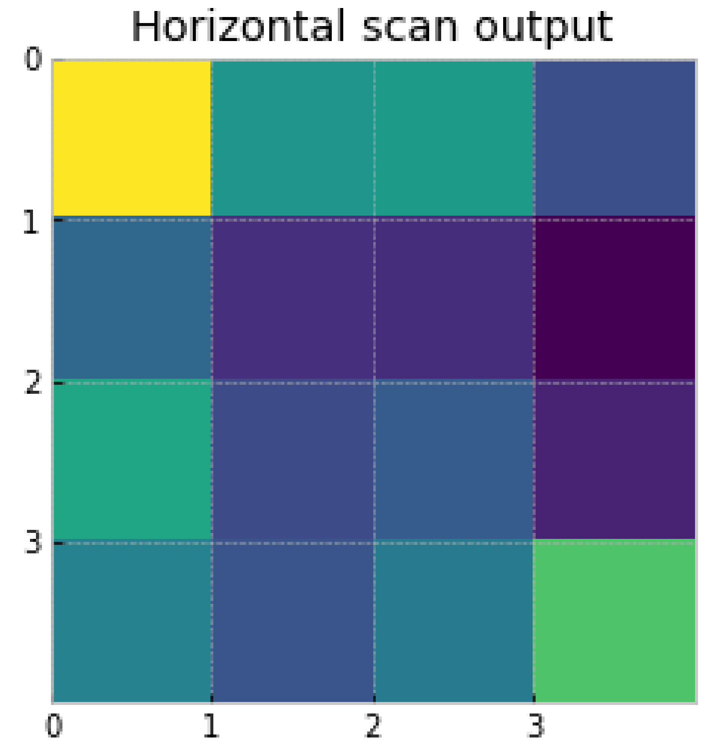
Qiskit approach

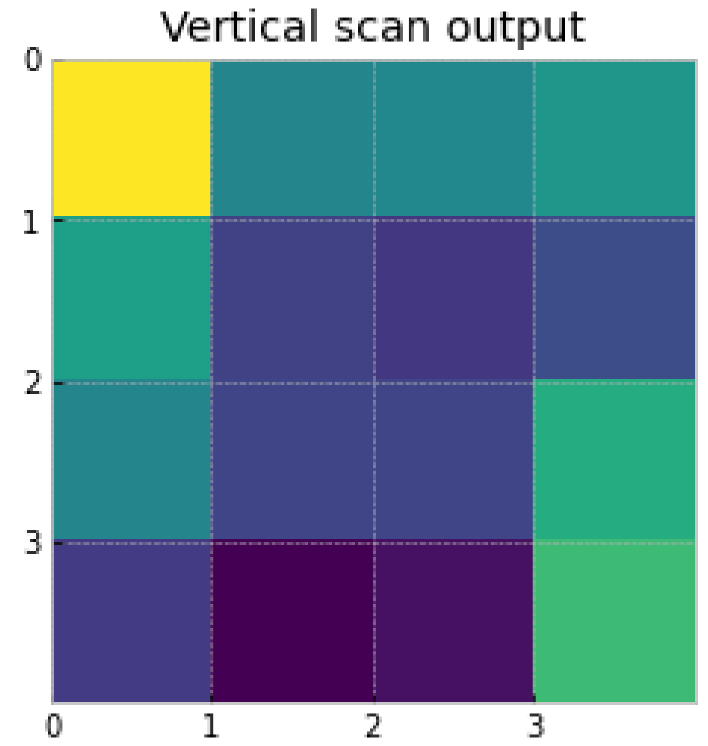
- Create empty circuits for all binary combinations of qubits. For 5 qubits – 32 circuits. X-gate for binary 1, and nothing for 0.
- Execute all circuits and gather noise for each. Can be executed at once.
- Apply gathered diff to your code results

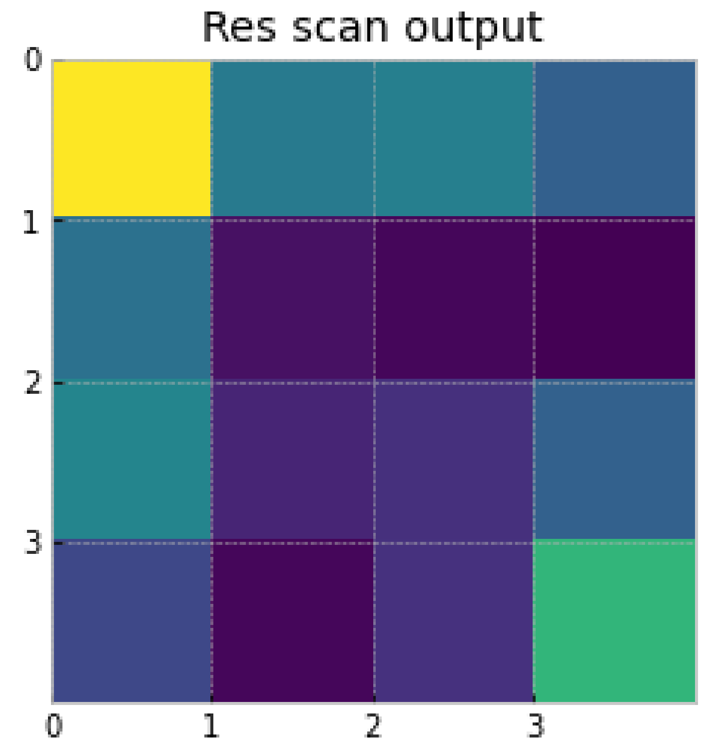
NOISE MITIGATION EXPERIMENTS RESULTS











CONCLUSIONS

- Quantum image processing is good task for beginners in quantum computing.
- Easy to implement; Visualizable; Can be validated by eyes.
- Image processing tasks have big complexity for amplitude encoding.
- Circuits depth is big.
- Results are noisy.

- Notebook can be found here https://github.com/drinkertea/Quantum-Image-Processing

HIGHER SCHOOL OF ECONOMICS

NATIONAL RESEARCH
UNIVERSITY