Image Stacking by Quantum Annealing

Tristan Cook[1]

[1]QxBranch Inc.

1 Abstract

Image stacking is the process of taking a large set of images and breaking it down into a few subsets to find the optimal subset for a purpose. For night-time photography, that may be identifying the optimal noise filter; for geophysical imaging, it could be identifying erroneous features; for microscopy, it could be focus selection. There are several classical algorithms used to select these subsets of images; however, it is a hard problem and most efficient methods are approximate by necessity. In this work, we approach the challenge of image stacking using an extended quantum MAXCUT clustering algorithm to build stacks based on correlated images. We present a discussion on the performance and scaling of the algorithm and propose directions for further research.

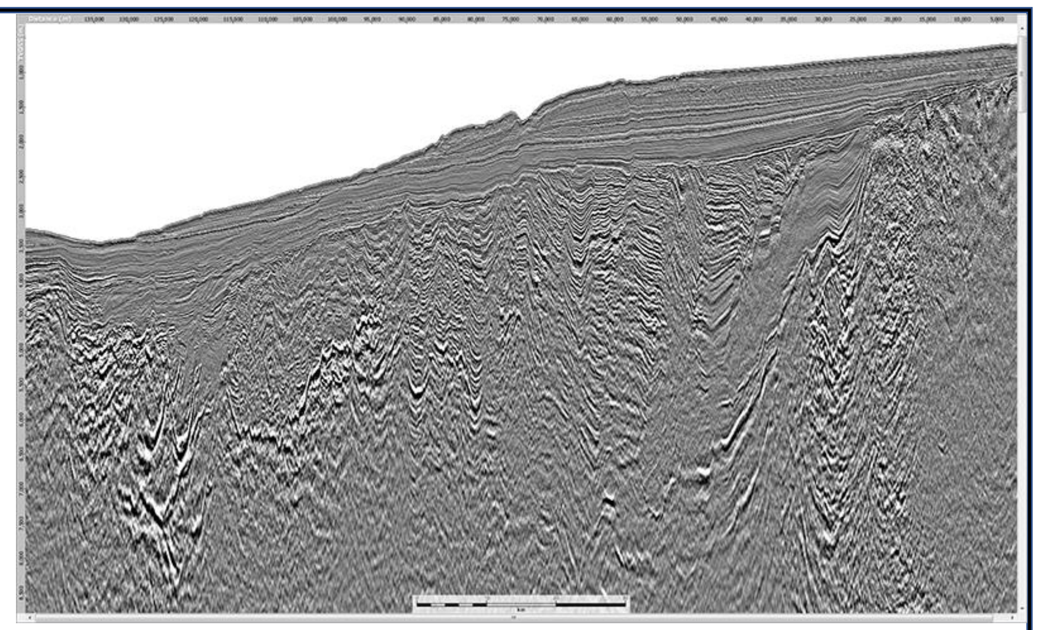
2 Motivation

Image Stacking

Image stacking is a challenging optimization problem with a number of practical applications including:

- Seismic imaging
- Astro / night-time photography
- Microscopy

The premise behind image stacking is to take a set of images of the same data, landscape, or object, and combine those images into a so-called "stack" that reduces the noise present in the images.



Example Seismic Image (Source: Geoscience Australia)

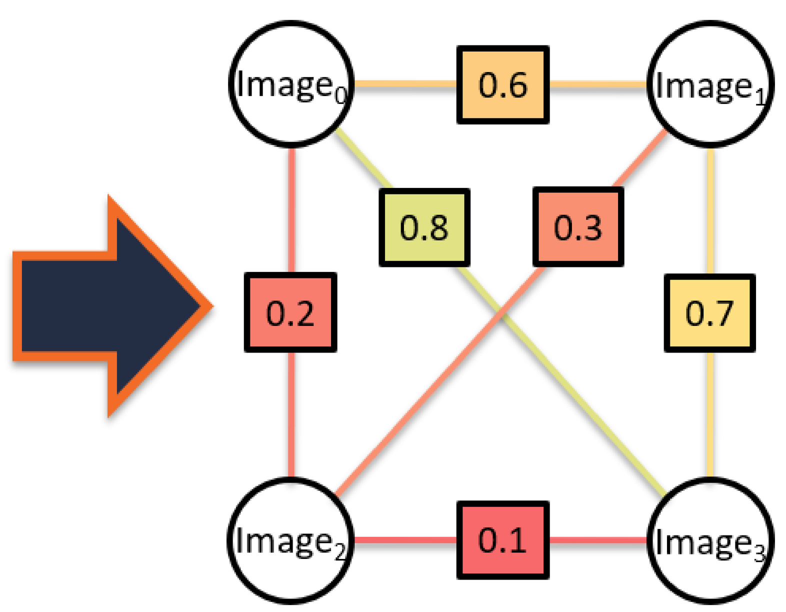
In this work we present a quantum approach to image stacking using the QCut quantum clustering algorithm first presented in Smelyanskiy et al 2012, to cluster sets of images into stacks. This QCut algorithm extends the MAXCUT algorithm from just 2 clusters to K clusters.

We discuss the formulation of Hamiltonian, mapping of an image stacking problem to the Hamiltonian, and some pre-processing methods for reducing resource needs and degenerate solutions.

Translation to a Graph Problem

Translating our image stacking problem to a graph problem is quite simple; each image can be treated as a vertex in a graph, with the vertex value indicating which stack it belongs to. This value is solved for by using a cost function to weight the edges between the vertices, and then minimizing the total cost of the graph. The selection of this cost function is determined by the features you are trying to cluster for, and is outside the scope of this poster. For these examples we have simply used a correlation value based on the Euclidean distance between the pixel vectors of each image. In the example graphs, these images are plotted as points with those distances between them.

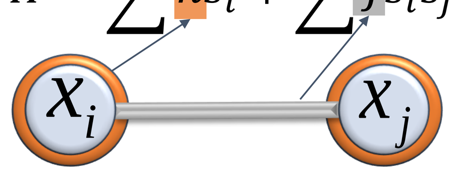
	Image 0	Image 1	Image 2	Image 3
Image 0	1	0.6	0.2	0.8
Image 1		1	0.3	0.7
Image 2			1	0.1
Image 3				1



Adiabatic Quantum Optimization Refresher

- Commonly shortened to AQO
- Minimizes an energy Hamiltonian, Ising Spin Glass to the right is one example form
- h is the bias on the variable
- J is the bias on the coupler

$$H = \sum h s_i + \sum J s_i s_j$$



3 Methodology

Extending Quantum MAXCUT

The original weighted MAXCUT Hamiltonian:

$$H_{MAXCUT} = \frac{1}{2} \sum_{u,v\, u\neq v}^{V} C_{uv}(1 - s_u s_v)\,, s \in \{-1,1\}$$

- V is the set of vertices on the graph
- C_{uv} is cost of the edge connecting vertices u & v
- S is the set of Ising qubits, with values of either -1 or 1

To extend the MAXCUT Hamiltonian to multiple clusters, QCut (Smelyanskiy et al 2012) requires additional variables:

- K is the set of clusters (chosen before execution)
- X is the set of QUBO qubits, with values of either 0 or 1

This extended Hamiltonian uses "one-hot" encoding, which means that for each vertex, we have a qubit for each cluster, and the vertex is assigned to whichever cluster's qubit is 1. To achieve this, we need two main components in the Hamiltonian.

First, we have an optimization term, which minimizes the edge weight between all vertices in each cluster. It does this by checking each edge on the graph to see whether the vertices at each end are in the same cluster:

$$H_{opt} = \sum_{k}^{K} \left(\sum_{u,v\, u\neq v}^{V} C_{uv} x_{uk} x_{vk} \right), x_{vk} = \begin{cases} 1, \text{if } v \in k \\ 0, \text{otherwise} \end{cases}$$

Next is a penalty term, which applies an energy penalty whenever a constraint is violated. In this case, the only constraint is that for a solution to be valid, only one cluster's qubit should be 1 for each vertex. This is a soft constraint, because if it is violated, we can label that vertex an outlier. The penalty term should equal zero if exactly one qubit is high, and more than zero otherwise. The penalty term can be scaled using a coefficient to tune the tolerance for outliers on a problem-by-problem basis:

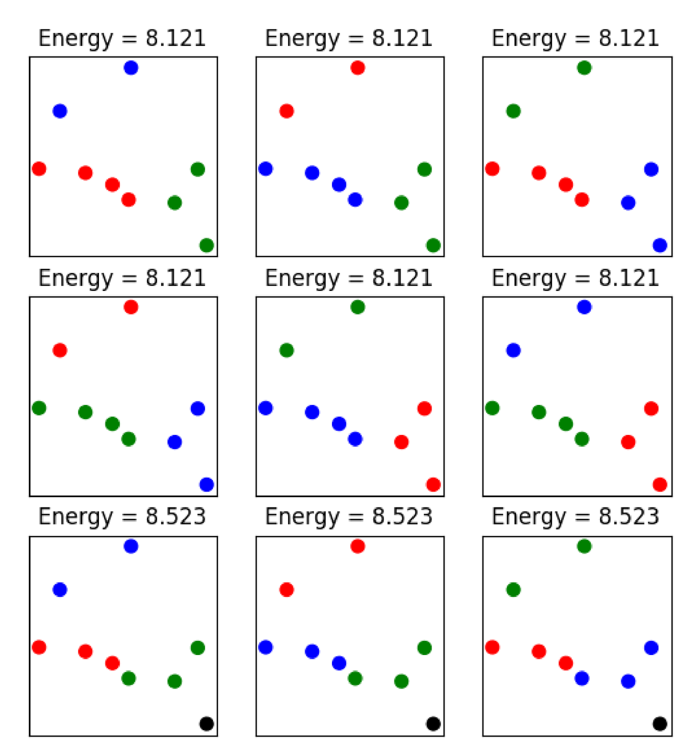
$$H_{pen} = \sum_{v}^{V} \left(1 - \sum_{k}^{K} x_{vk} \right)^2$$

Giving us the complete Hamiltonian to map to the Ising Spin Glass:

$$H = A_{pen} \sum_{v}^{V} \left(1 - \sum_{k}^{K} x_{vk} \right)^2 + \sum_{k}^{K} \left(\sum_{u,v\, u\neq v}^{V} C_{uv} x_{uk} x_{vk} \right)$$

To the right, the solution energy values for a simple example have been evaluated completely, and the nine best solutions identified. It's worth noting that the first six solutions are the same, just with the different cluster colors rotating. This solution degeneracy is a downside of one-hot encoding, and can be mitigated to some extent by preassigning a vertex to a cluster.

In these graphs, black vertices are outliers, and the axes have no units. The points on the graph have been spaced such that the distance between them is proportional to the edge weight costs as previously discussed.



9 Vertices, 3 Clusters, Brute Force Evaluation

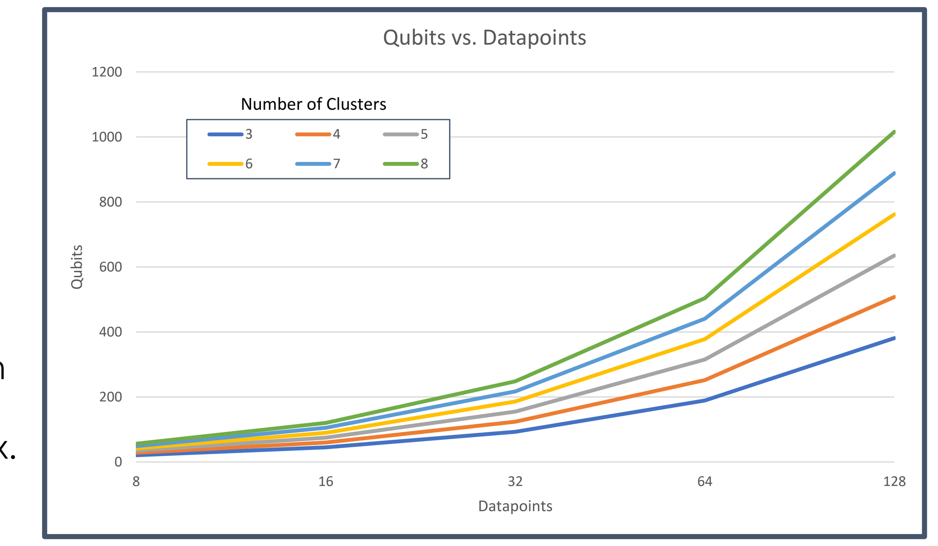
4 Discussion

Qubit Resource Scaling

The number of qubits required to perform this algorithm is given by the following equation:

$$\text{Qubits} = |V||K|$$

There are some additional tricks to improve this scaling, such as the pre-assignment method discussed below, which reduces the qubits require by K for each pre-assigned vertex.

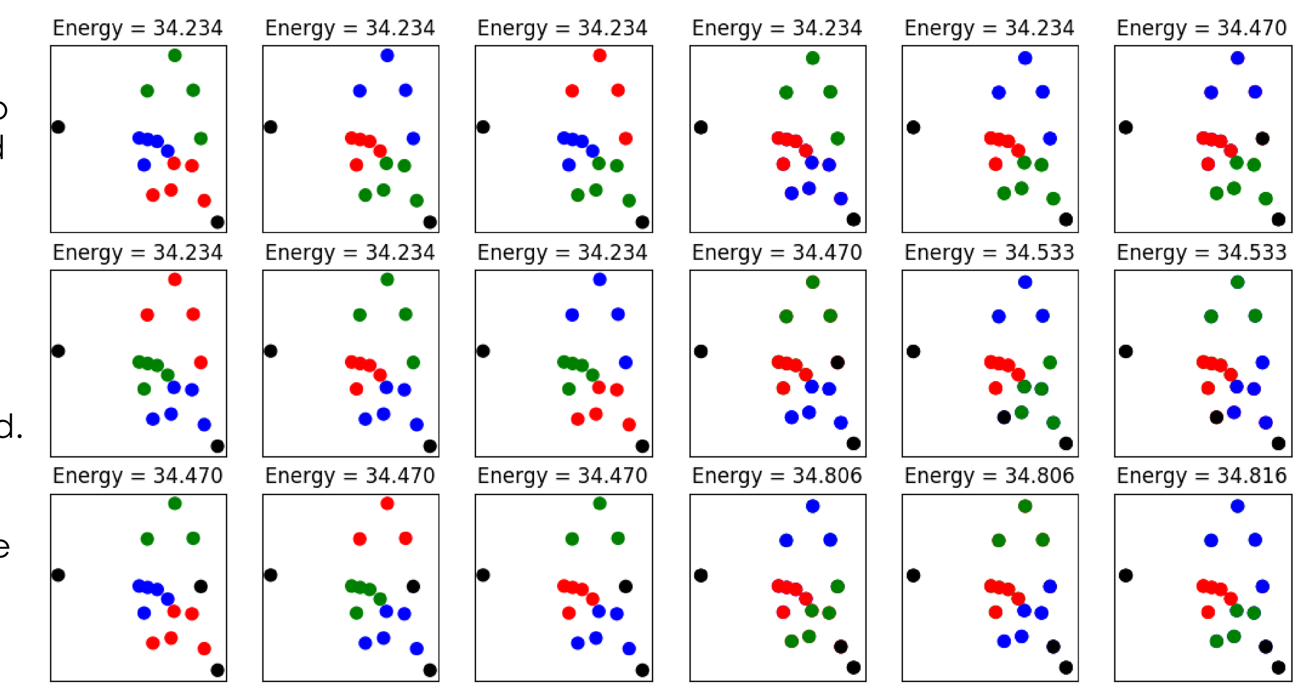


Vertex Pre-Assignment

Vertex pre-assignment is a simple method of pre-processing the data to reduce the number of qubits required and the number of degenerate solutions returned.

The graphs on the left have no pre-assignment, and as discussed previously, the first six solutions are identical with the cluster color shuffled.

For the graphs on the right, an image was assigned to the red cluster before optimization began, and it can be seen that the red cluster does not move, and more varied solutions are returned.

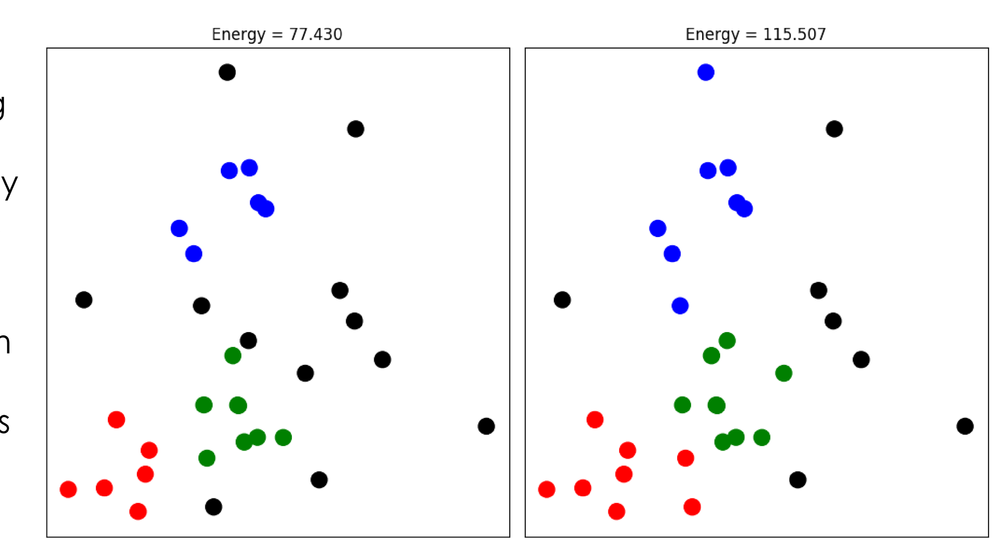


16 Vertices, Without Pre-Assignment — 16 Vertices, With Pre-Assignment

Penalty Coefficient Scaling

The graphs to the right demonstrate the effect of scaling up the penalty coefficient. Both are optimal configurations for their parameter set, and as the penalty term increases, the number of outliers overall decreases, and the cluster size increases.

Experimentation has shown that the penalty scaling term should increase with the number of variables, and decrease as the number of clusters increases. This occurs because the energy of a given cluster is proportional to the number of vertices in that cluster.



32 Vertices, Penalty = 1.0 — 32 Vertices, Penalty = 2.0

Initial Results & Next Steps

Initial results have found that the naïve cost function used for image correlation leads to gaussian noise dominating the clustering results. While useful for identifying efficient noise masks, it is not ideal for identifying human-recognisable features in images. As such, it is mostly likely to apply well to astrophotography from the applications discussed, and the next steps of research will be to experiment with other cost functions for feature extraction.

5 Conclusion

As the number and complexity of quantum algorithms increases, the number of applications they can be applied to does so too. In this poster, we presented an application for the QCut quantum MAXCUT algorithm for quantum k-means clustering, and discussed its application to image stacking. We discuss its performance and present some tools to improve the algorithm. Follow-on work will focus on exemplary use cases and demonstrations.

6 References

1. Smelyanskiy, V. N., Rieffel, E. G., Knysh, S. I., Williams, C. P., Johnson, M. W., Thom, M. C. Macready, W. G. & Pudenz, K. L. (2012). A Near-Term Quantum Computing Approach for Hard Computational Problems in Space Exploration. https://arxiv.org/abs/1204.2821
2. Sigdel, M. S., Sigdel, M., Dinç, S., Dinç, I., Pusey, M. L., & Aygün, R. S. (2016). FocusALL: Focal Stacking of Microscopic Images Using Modified Harris Corner Response Measure. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM, 13(2), 326–340. http://doi.org/10.1109/TCBB.2015.2459685
3. Geoscience Australia; http://www.ga.gov.au/scientific-topics/disciplines/geophysics/seismic

QxBranch thanks the USC Lockheed Martin Quantum Computation Center for its assistance to this project