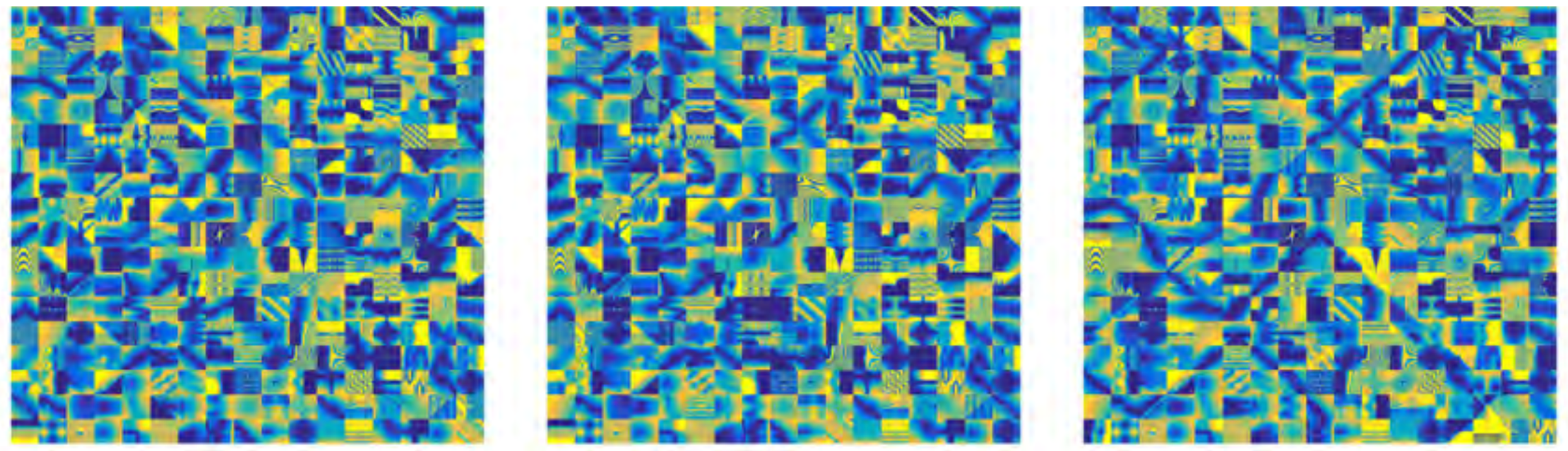
Optimising the Image Mosaicing of Visual Representations of Functions

Hannah Perry Supervised by Professor Kate Smith-Miles and Dr Andrés Muñoz

The University of Melbourne

Background

This project explores the optimisation of mosaics of contour plots of functions. These functions were created via a genetic programming algorithm, initially intended for stress-testing optimisation algorithms in [1].



'Negentropy Triptych', digital, Kate Smith-Miles and Mario Andrés Muñoz, copyright 2019

In the above triptych, image tiles have been swapped by hand to decrease structure and river continuity on the left, and increase structure and river continuity on the right, with the centre being the initial randomised arrangement.

Is it possible to design an optimisation algorithm to automate the process of swapping the images to enhance or destroy the river connectivity?

Previously in [3] a greedy algorithm that used seven neighbourhood swap operators was used to automate the destruction and creation of the river structure. Different objective functions were tried, based on measures of mutual information, connected pixels, and pattern identification.

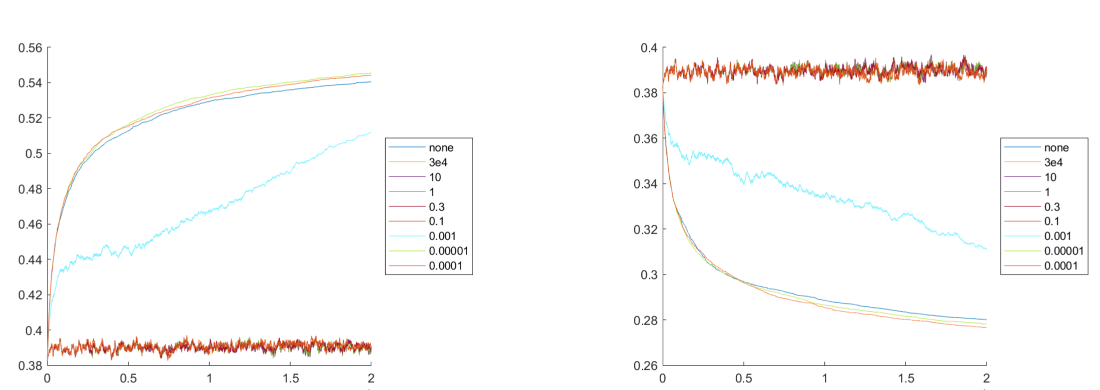
Simulated Annealing

To explore the effect of using a non-greedy algorithm, we implemented simulated annealing into the existing code from [3]. Simulated annealing is a technique that aims to improve hill climbing ability by temporarily accepting a worse solution than the current solution. This can be useful to avoid getting stuck in local minima or maxima if the objective function is non-convex.

The probability of accepting a worse solution depends on objective function at current solution e, solution being considered e_{new}, and current temperature T. For example, in maximisation of the objective function:

$$P(e, e_{new}, T) = \begin{cases} 1 & e_{new} > e \\ \exp(\frac{e_{new} - e}{T}) & e_{new} \leq e \end{cases}$$

Unfortunately simulated annealing did not seem to have a significant impact on the value of the objective function in the solutions found, and visually the results were very similar to the greedy algorithm.



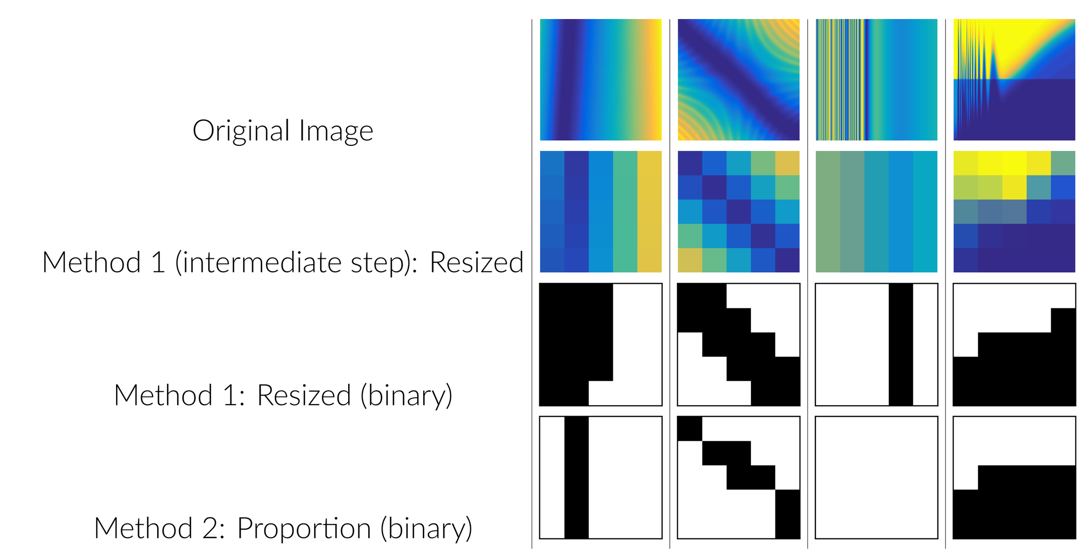
Plot of the average objective function of 10 runs across 20000 iterations of the algorithm, comparing simulated annealing with various starting temperatures to no simulated annealing

Graph Generating Algorithm

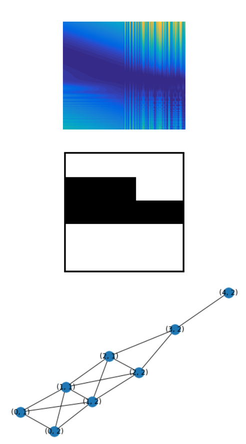
We considered a new approach, of an algorithm that builds up the mosaic, and an objective function that is based on treating the rivers like a graph and looking at paths through adjacent images.

The first step is to simplify each image to reduce complexity so that the algorithm can be run within a reasonable amount of time.

- Method 1: resize image (average pixel value, with interpolation by area), then consider if the pixel is within the blue hue range
- Method 2: consider 'cells' to be separate sections of the image, and look at the proportion of pixels in each cell that are within the deep blue hue range. If the proportion is sufficient then there is river in that cell of the image



A graph from each image can be obtained by considering the cells/pixels that contain enough river (deep blue) to be nodes, which share an edge when adjacent (by 8-connected definition). Then the shortest paths can be calculated between each cell node within the image. We are interested in finding the maximum of the shortest paths from a particular cell node.

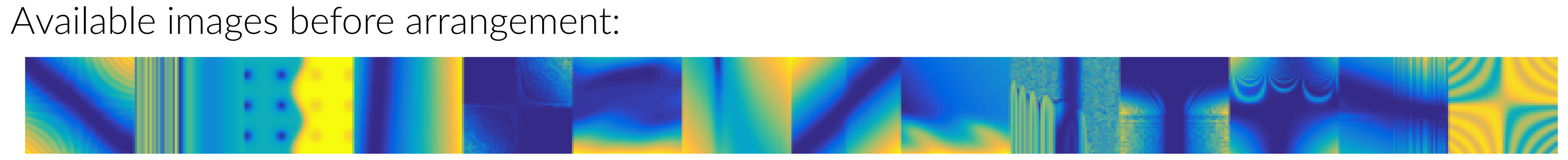


Which graphs should be joined together to create the 'supergraph' that represents the mosaic?

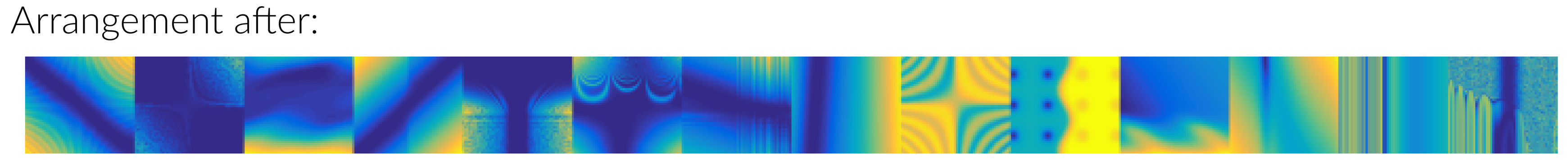
The objective function involves finding the maximum shortest connected path through the cells in each pair of adjacent image tiles, then summing this across all adjacent pairs of image tiles in the mosaic.

Example for selecting the next best image on the right side:

Available images before arrangement:



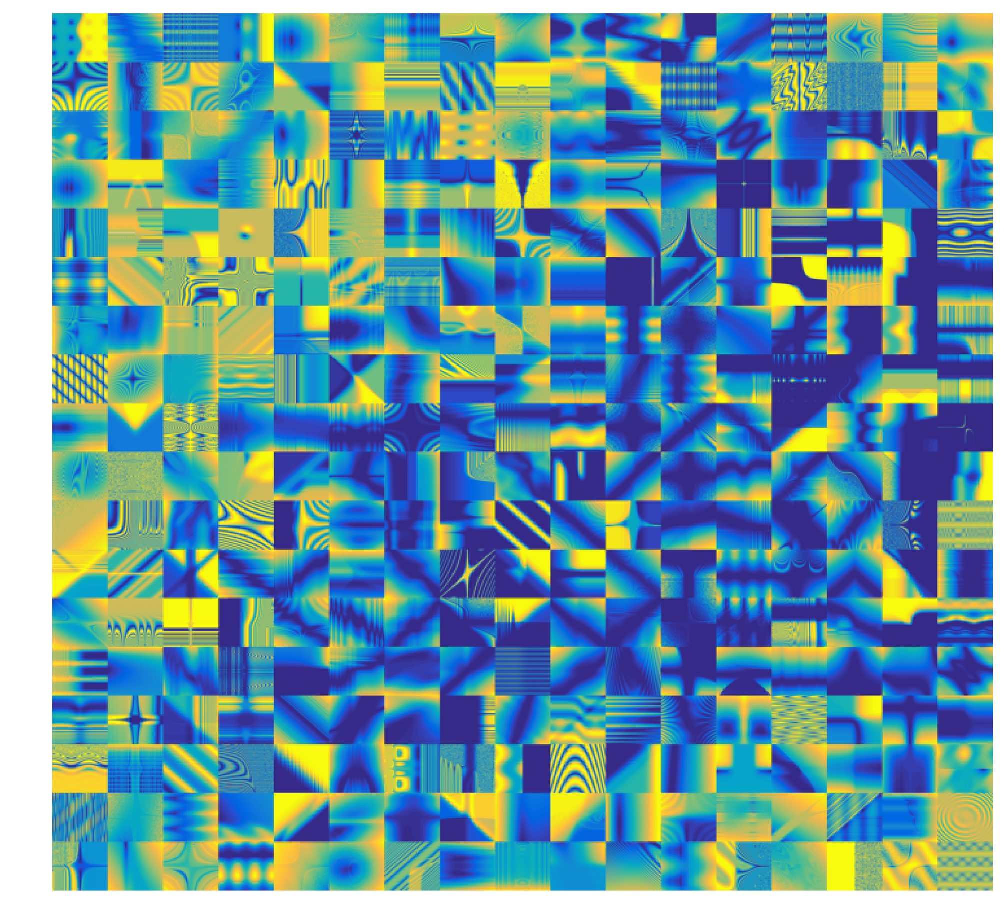
Arrangement after:



Where the algorithm sees the rivers:

The algorithm begins with a random image in the centre of the mosaic, and grows the mosaic by choosing a location to insert an image tile by picking the empty location adjacent to a non-empty location that has the most non-empty river-containing locations surrounding it (randomising for tie breaking)

Results and Observations



Maximising river connections with graph generating algorithm, with each image split into 5 by 5 cells

This example output of the graph generating algorithm being used to maximise the blue river connectivity does visually seem reasonably successful in connecting rivers, however it is sometimes at an angle that disrupts the flow. The image tiles that contain blue are concentrated in the region of the mosaic that grows initially, rather than creating linear structures. For the image tiles with diagonal rivers, diamond and cross patterns tend to occur, rather than continuous lines of river across the mosaic.

Further Discussion

Future directions to take this project would include to consider the aesthetic issues arising from the observation of the behaviour of the graph generating algorithm, and use them to guide further modifications, refinement and extension to the algorithm. For example, choosing the next location in a directional manner than promotes linear structures or incorporating a measure of direction into the objective function. The potential applications of this research include image scrambling and cryptography uses, where it is useful to be able to destroy existing structure in a reversible way.

References

[1] M. A. Muñoz and K. A. Smith-Miles. "Generating New Space-Filling Test Instances for Continuous Black-Box Optimization". In: *Evolutionary Computation* vol. 28. no. 3 (2020), pp. 379–404.

[2] Smith-Miles. *ARC Laureate project MATILDA*. URL: `http://www.matilda.unimelb.edu.au`.

[3] Kate Smith-Miles and Mario Andrés Muñoz. "Human versus computer construction of mathematical artworks on an order-disorder aesthetic spectrum". In: *Journal of Mathematics and the Arts, submitted (under review)* (2021).