Seam Carving

Parallelizing a novel new image resizing algorithm

Jacob Stultz

Motivation

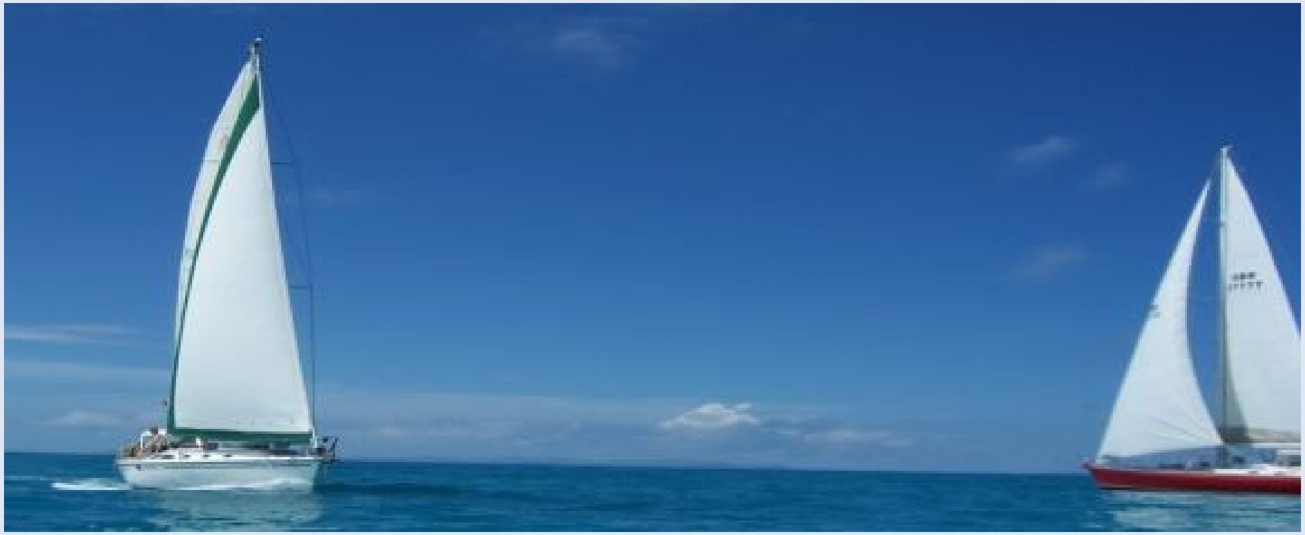
- Automated image resizing
  - Websites, other resizable Uis
- How to prevent:
  - Loss of important image information
  - Distortion

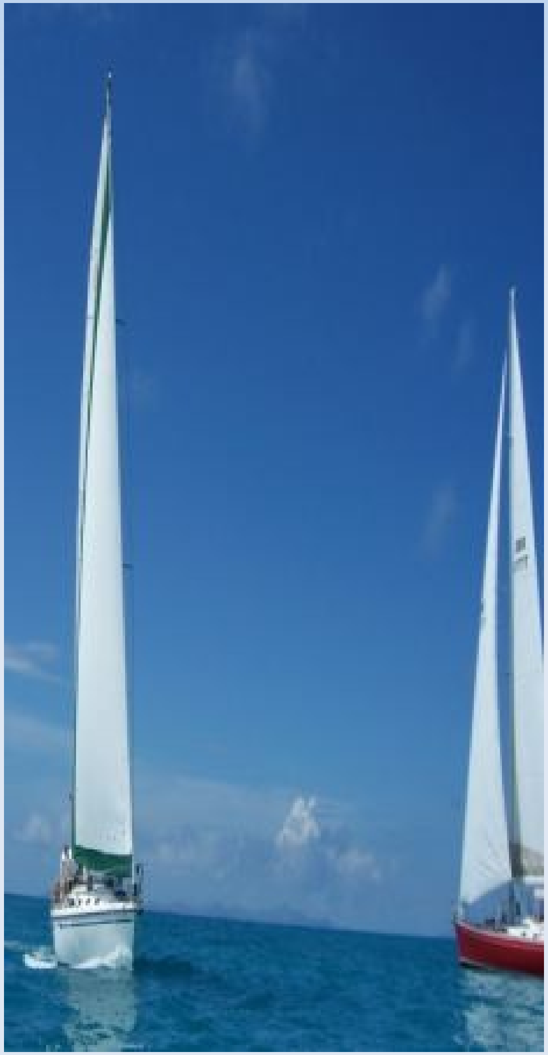
Existing Methods

- Cropping
  - Maintains aspect ratio
  - Must find image focus
  - Not always feasible

Existing Methods

- Image Scaling
  - Simple
  - Potential distortion
  - Only decent for scale factor of 0.5 to 2





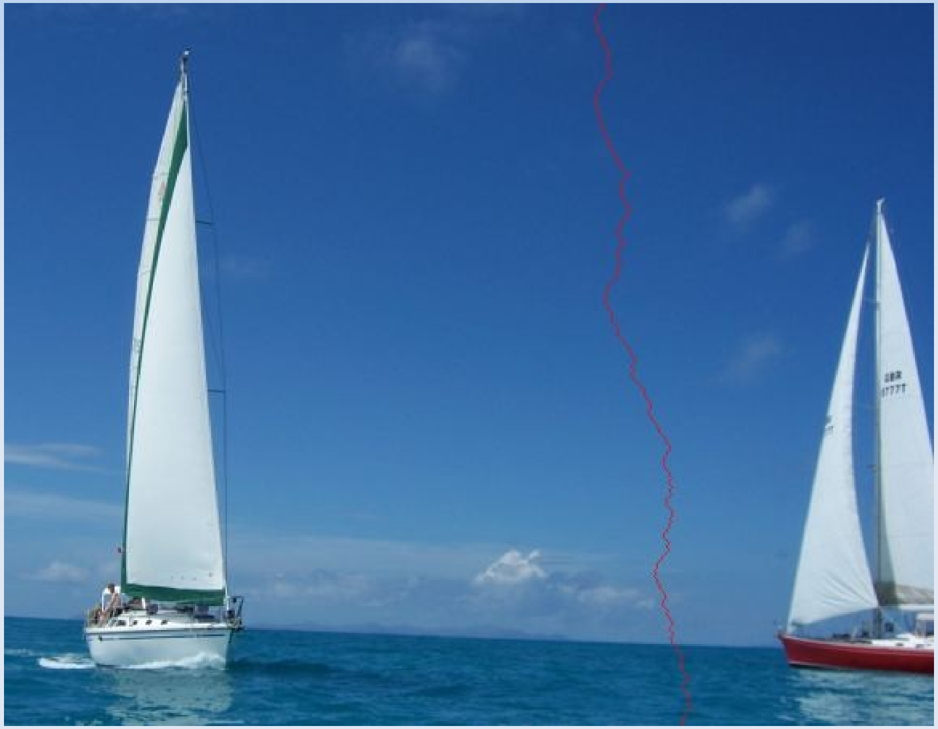
New Approach

- Determine importance of image components
  - Remove least important columns/rows
  - Remove least important pixels per column/row

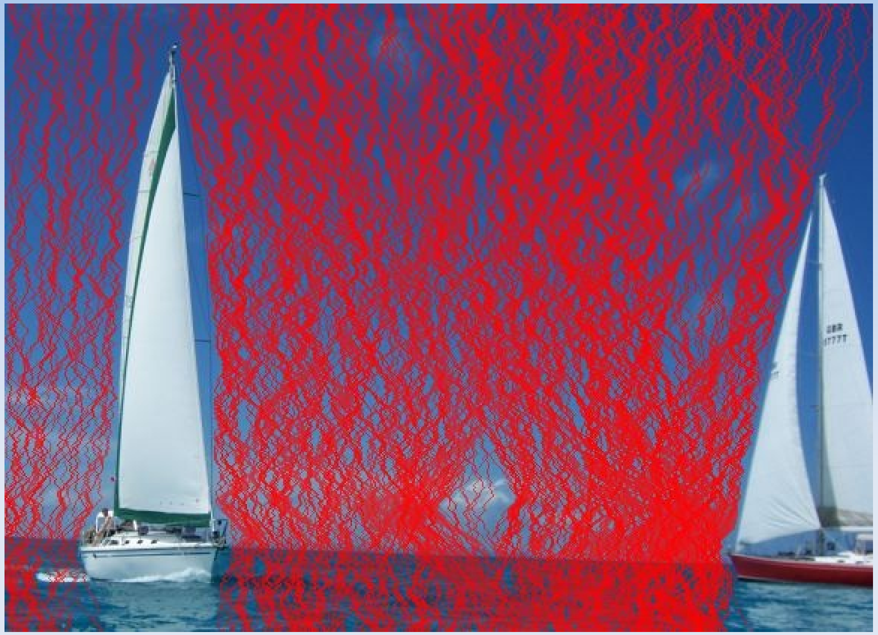
- Both methods are subject to distortion
- Pixel importance may vary within columns/rows

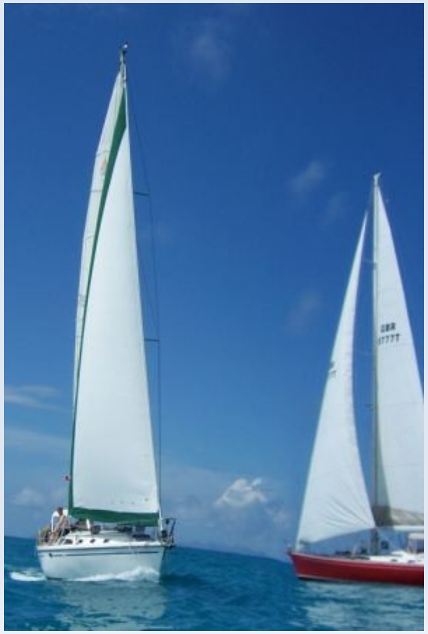
Better Approach

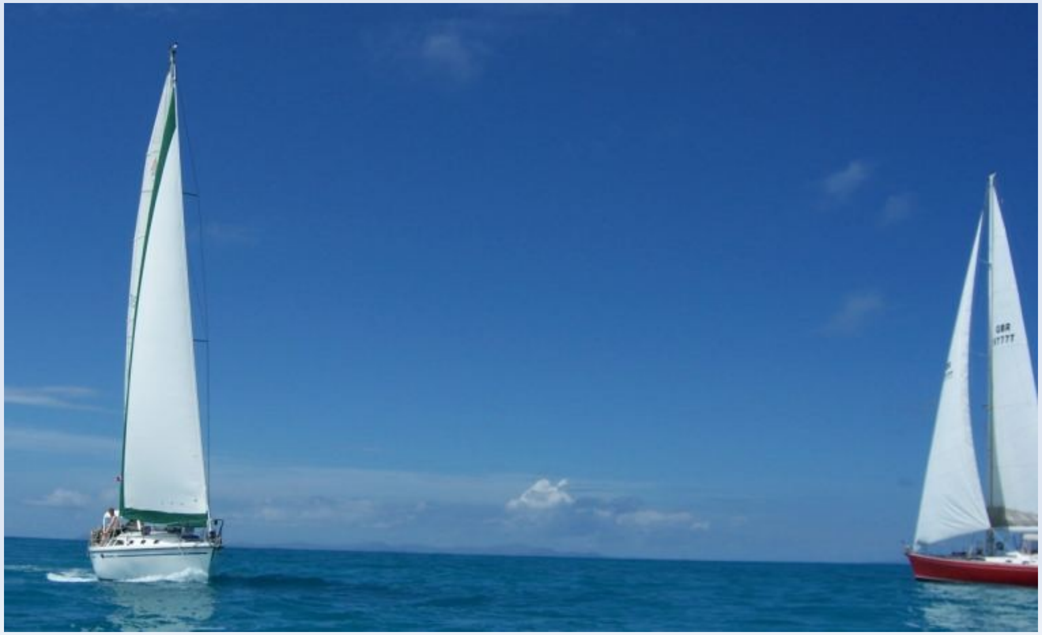
- Seam Carving
- Find and remove the least important seams, or paths across the image
- Compromise between previous two methods



Seam Carving







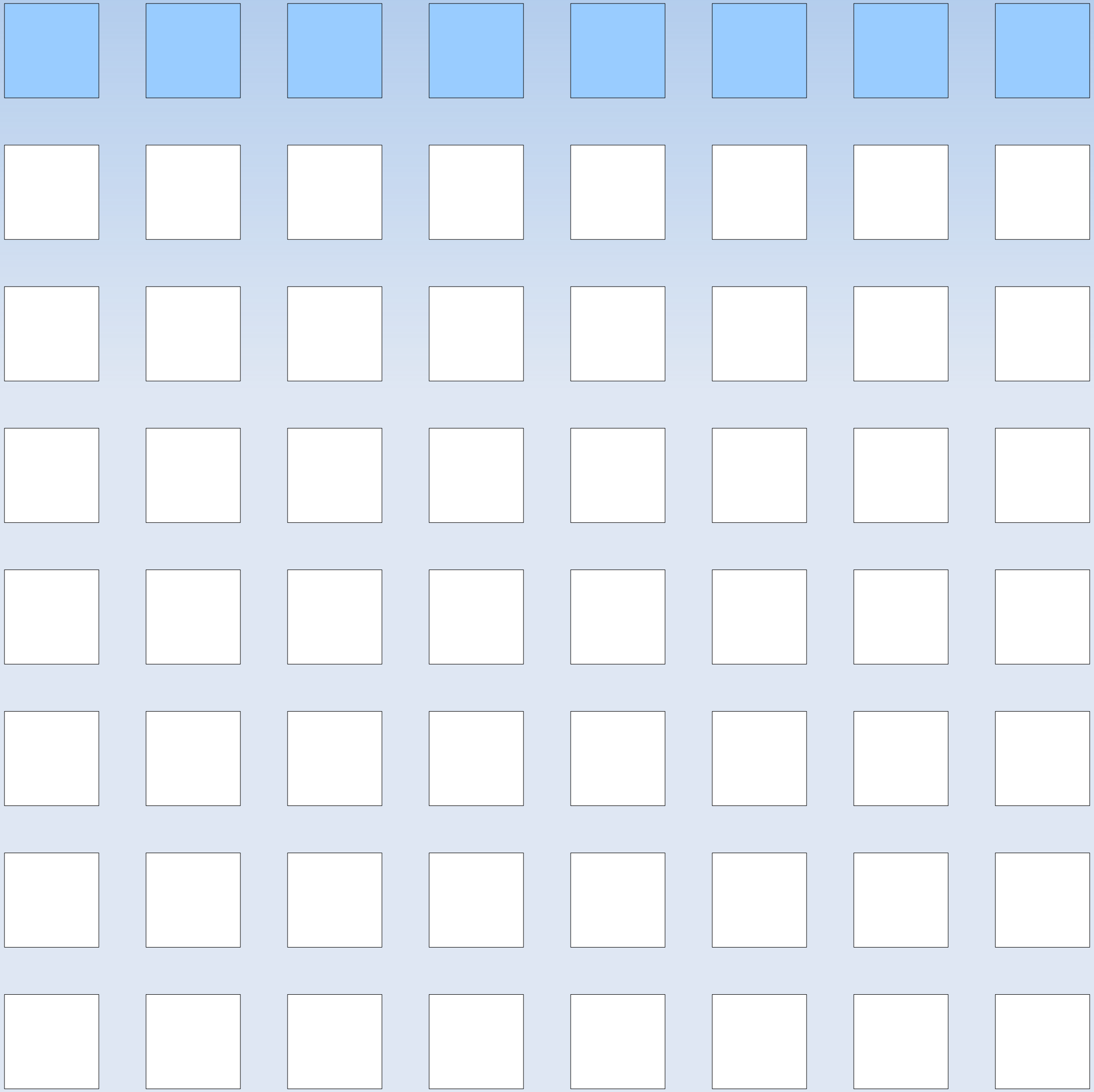
Seam Carving

- How to determine pixel importance?
  - “Energy” function
  - $E(x,y) = |d/dx(x,y)| + |d/dy(x,y)|$
  - Other functions possible
  - Find lowest energy path

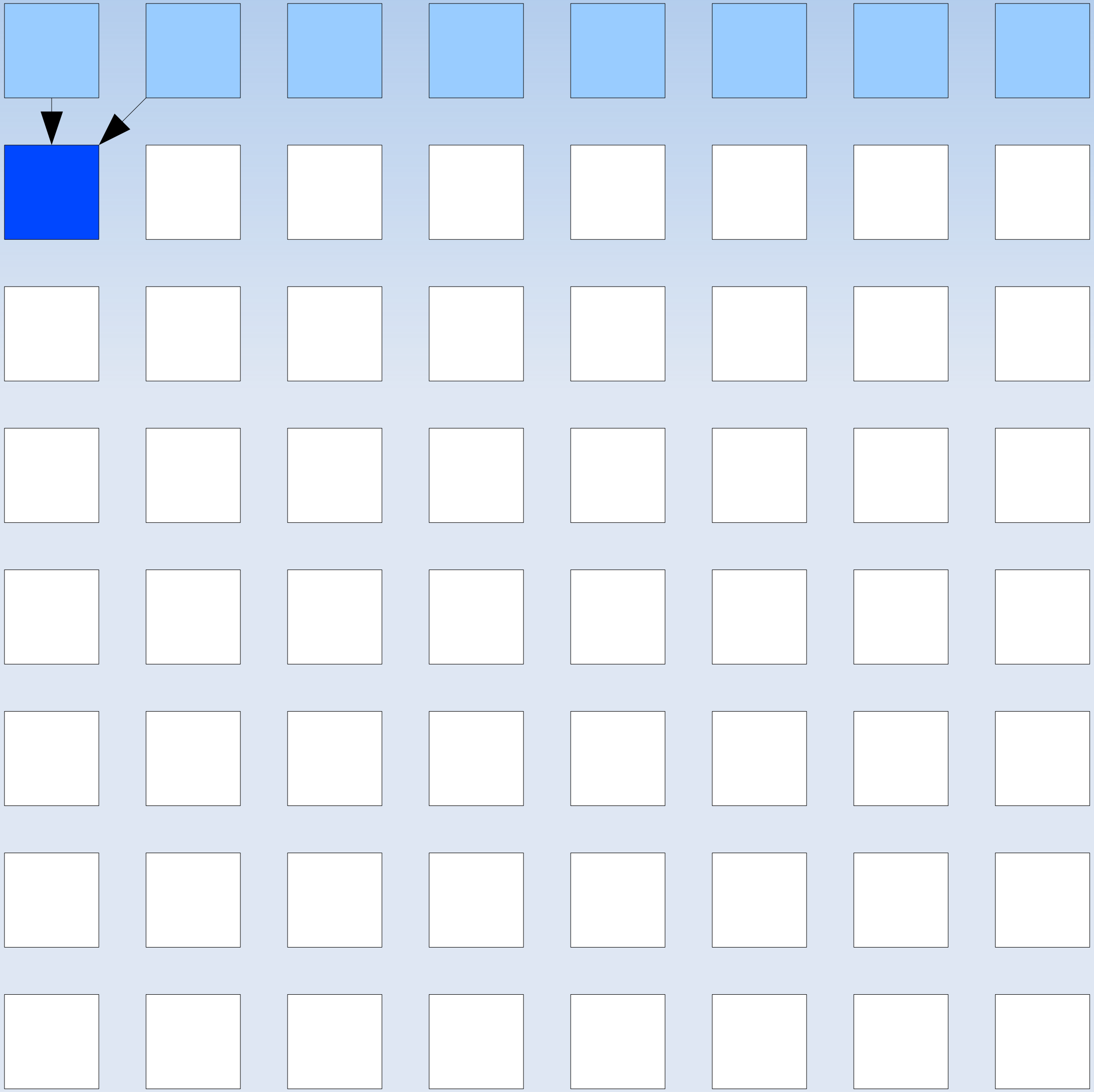
Seam Carving Algorithm

- Calculate pixel energies
- Minimum path starting from each column or row
  - Dynamic Programming:
  - $$M(x, y) = E(x, y) + \min[M(x-1, y-1),\; M(x, y-1),\; M(x+1, y-1)]$$
- Remove lowest energy seam
- Repeat

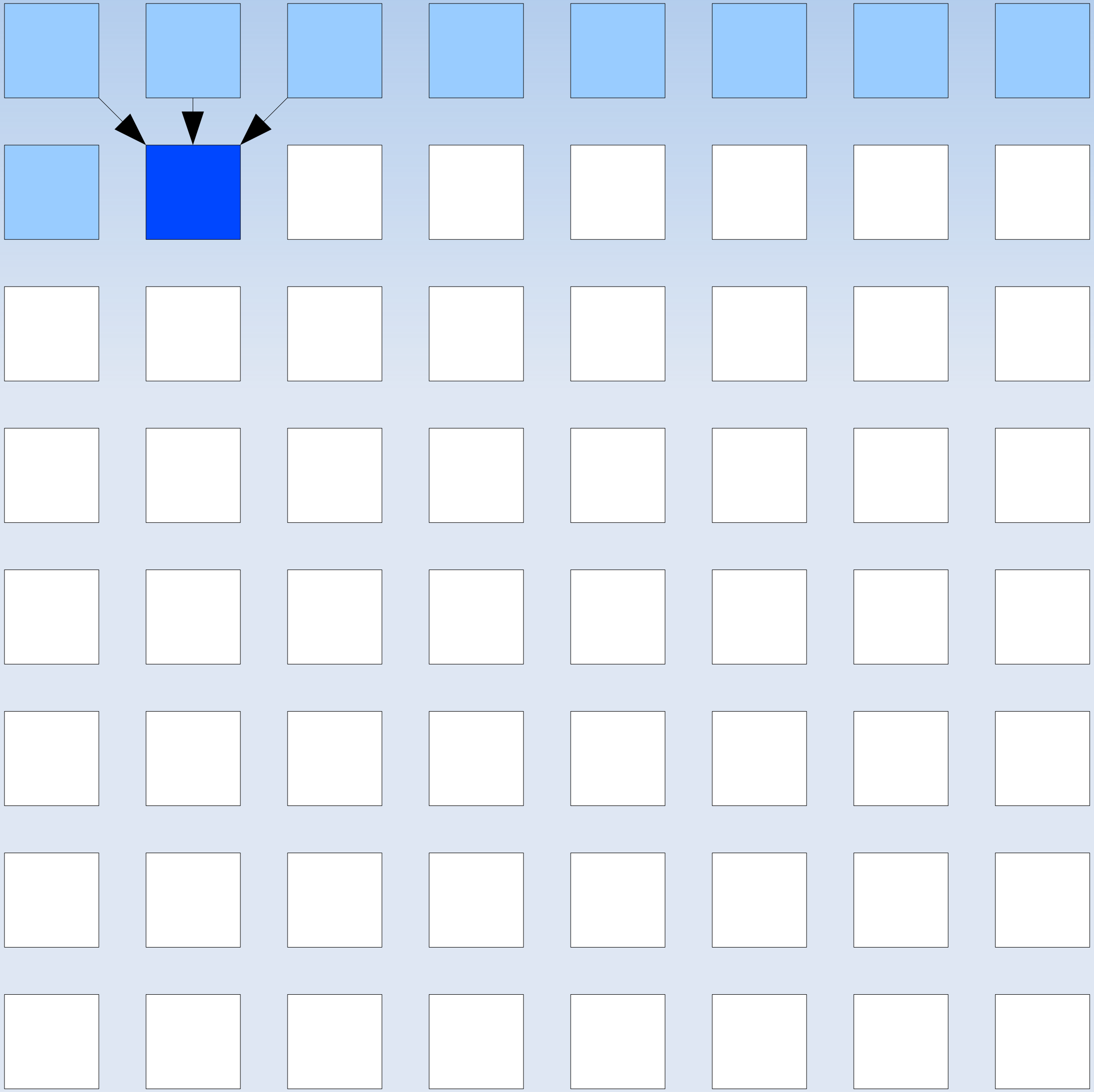
Seam Carving Algorithm



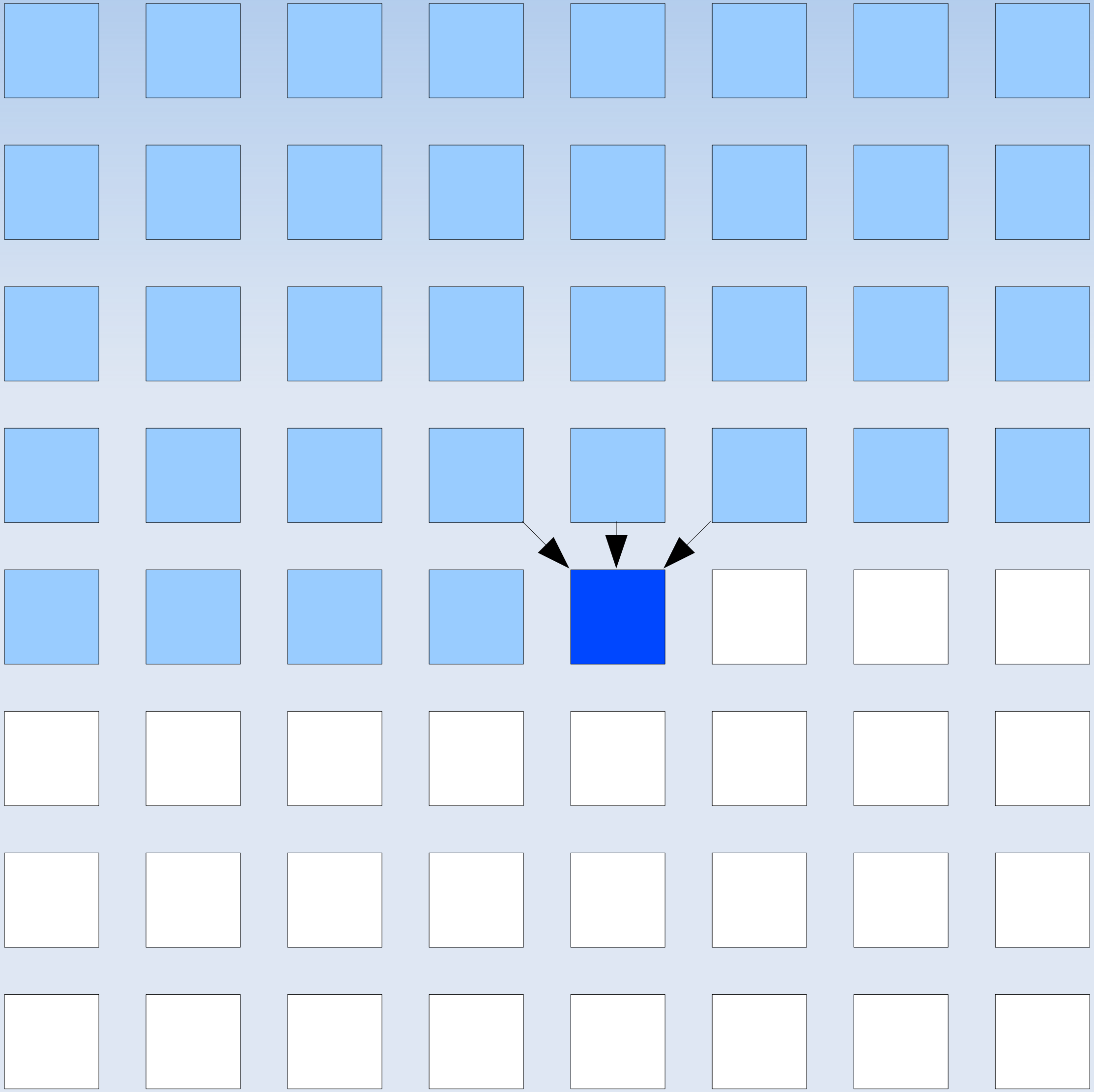
Seam Carving Algorithm



Seam Carving Algorithm



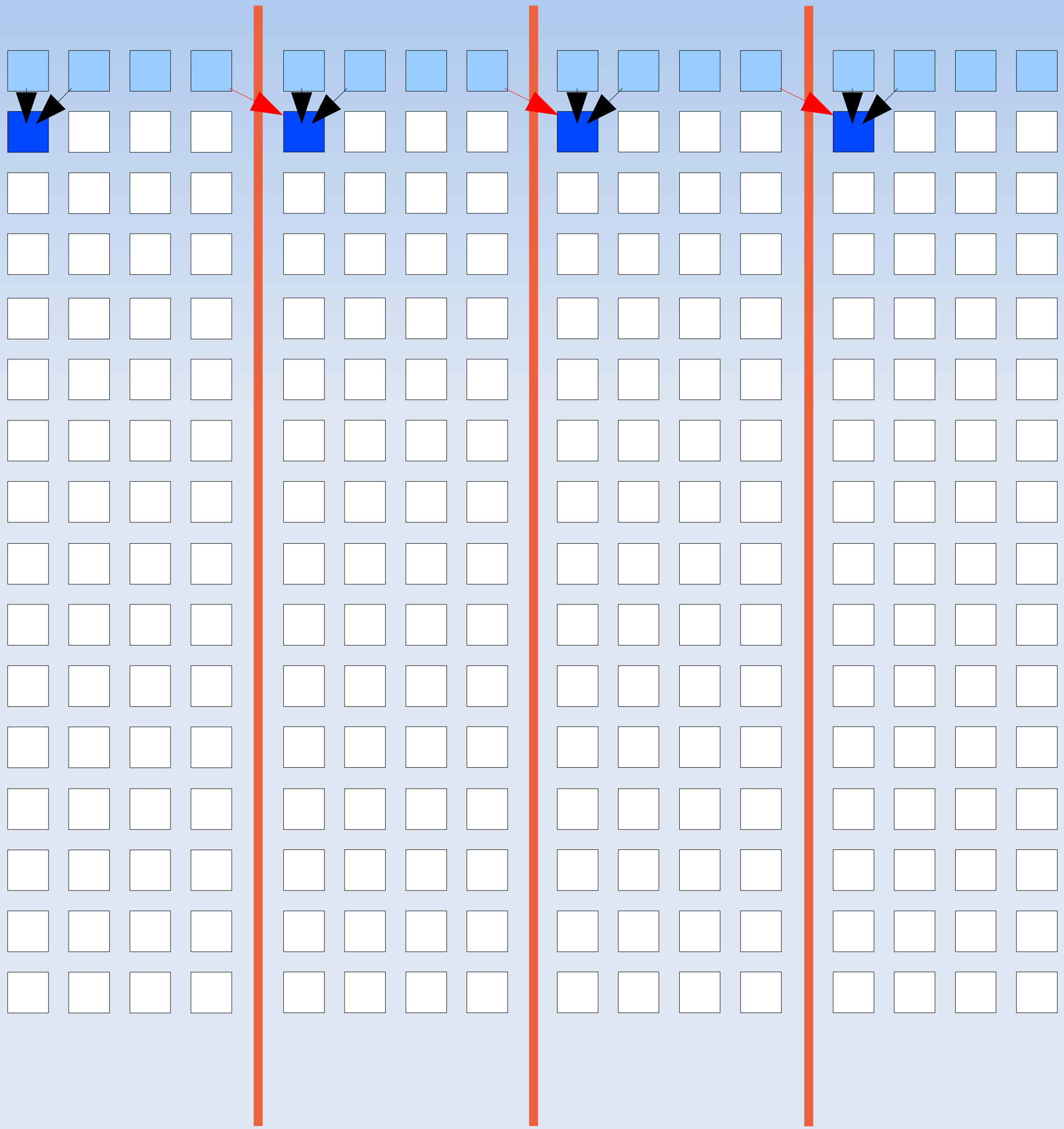
Seam Carving Algorithm



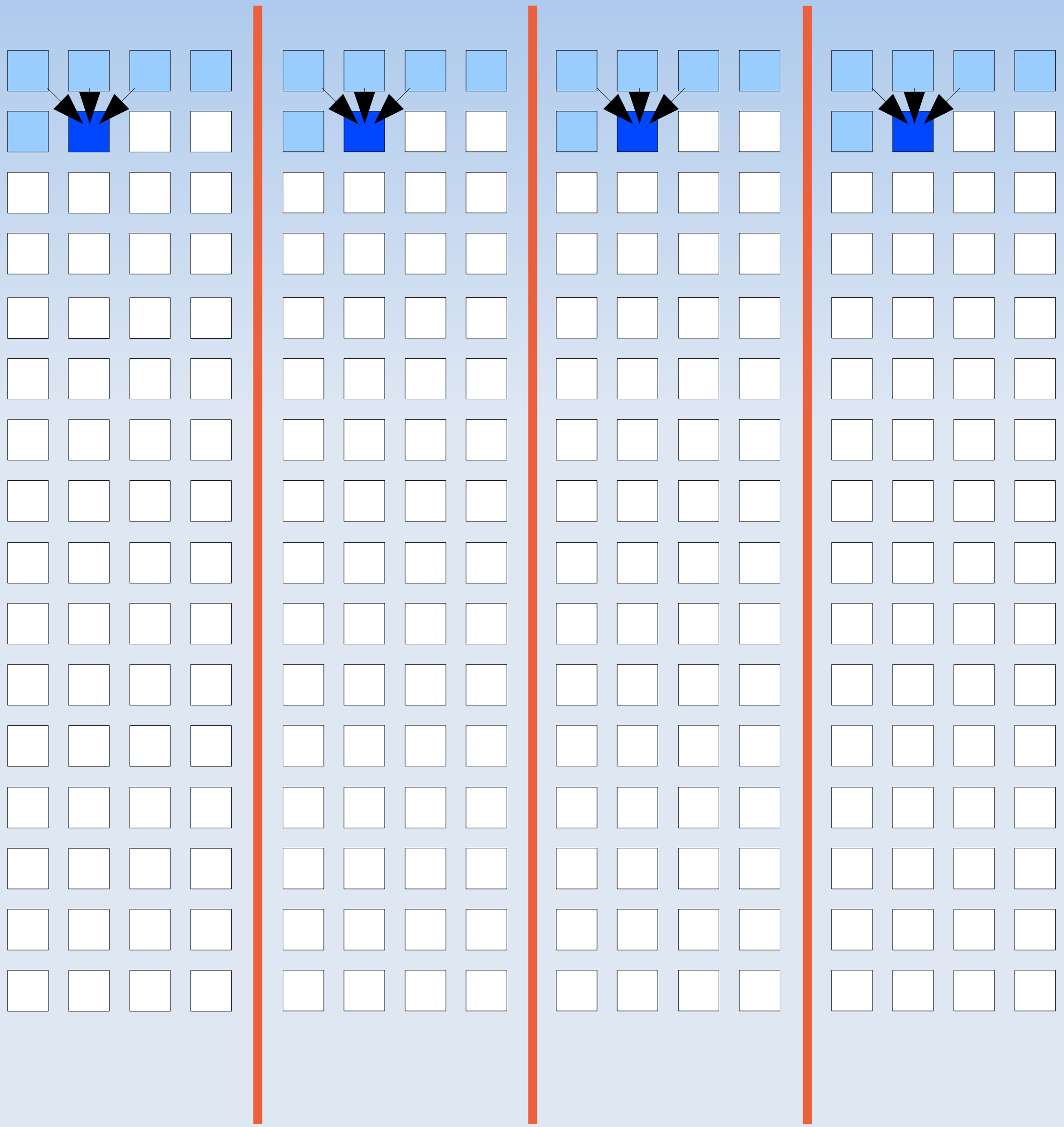
Parallelization Opportunities

- Energy and Minpath:
  - Most computation/data intensive
- Energy:
  - Trivially parallelizable
  - Only fully executed once
- Minpath:
  - Somewhat harder to parallelize
  - Repeated many times

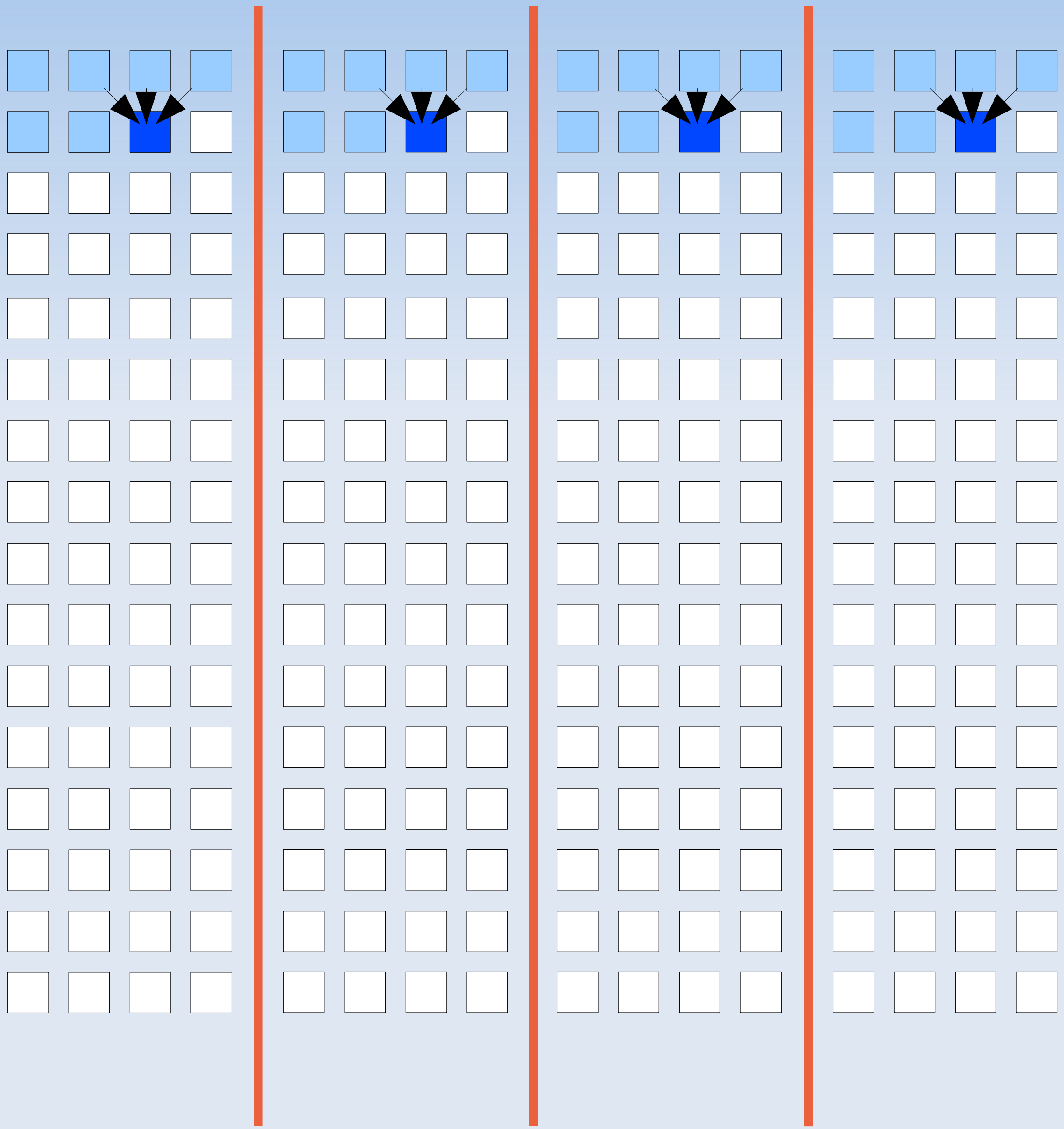
Parallelized Minpath



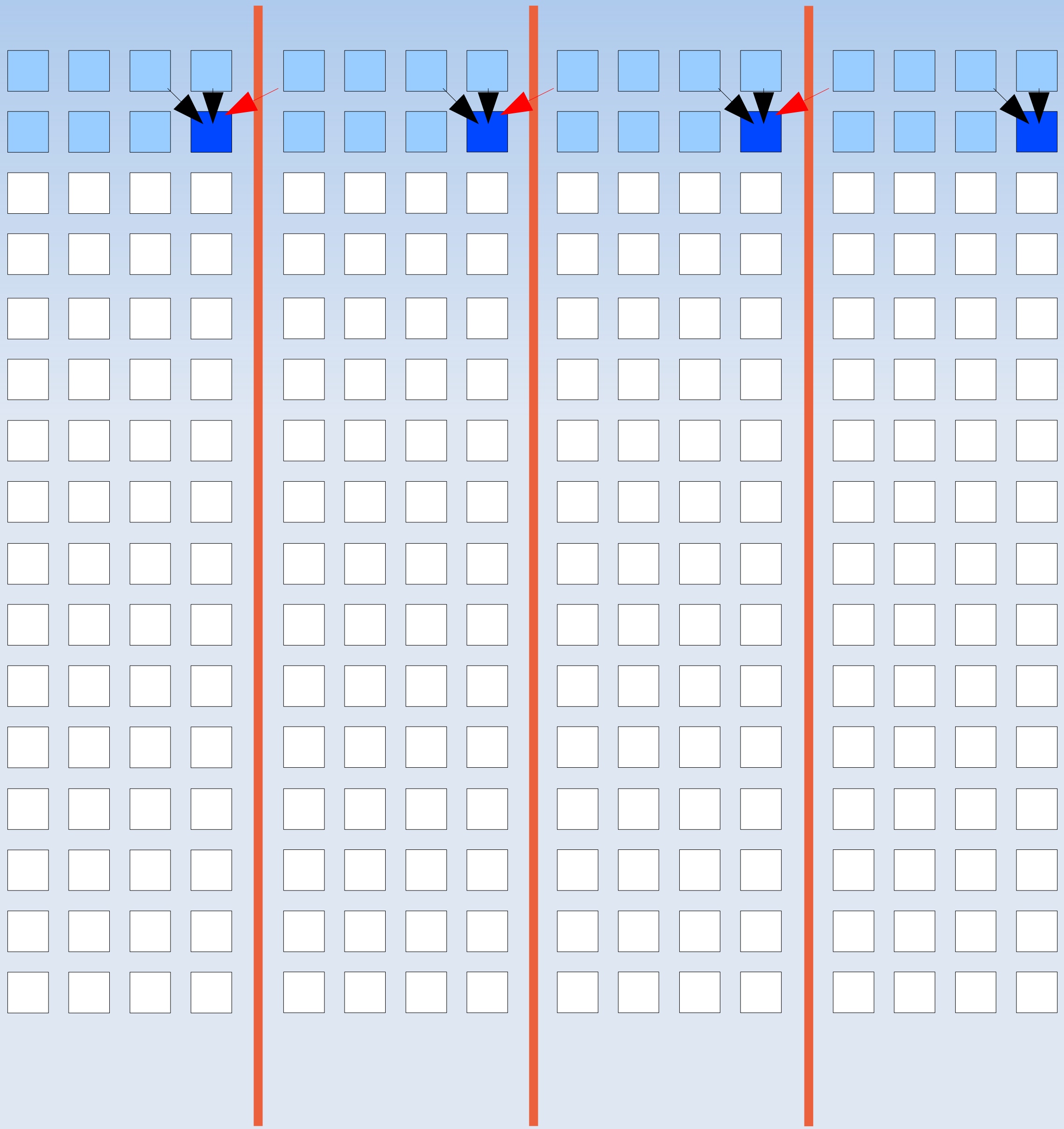
Parallelized Minpath



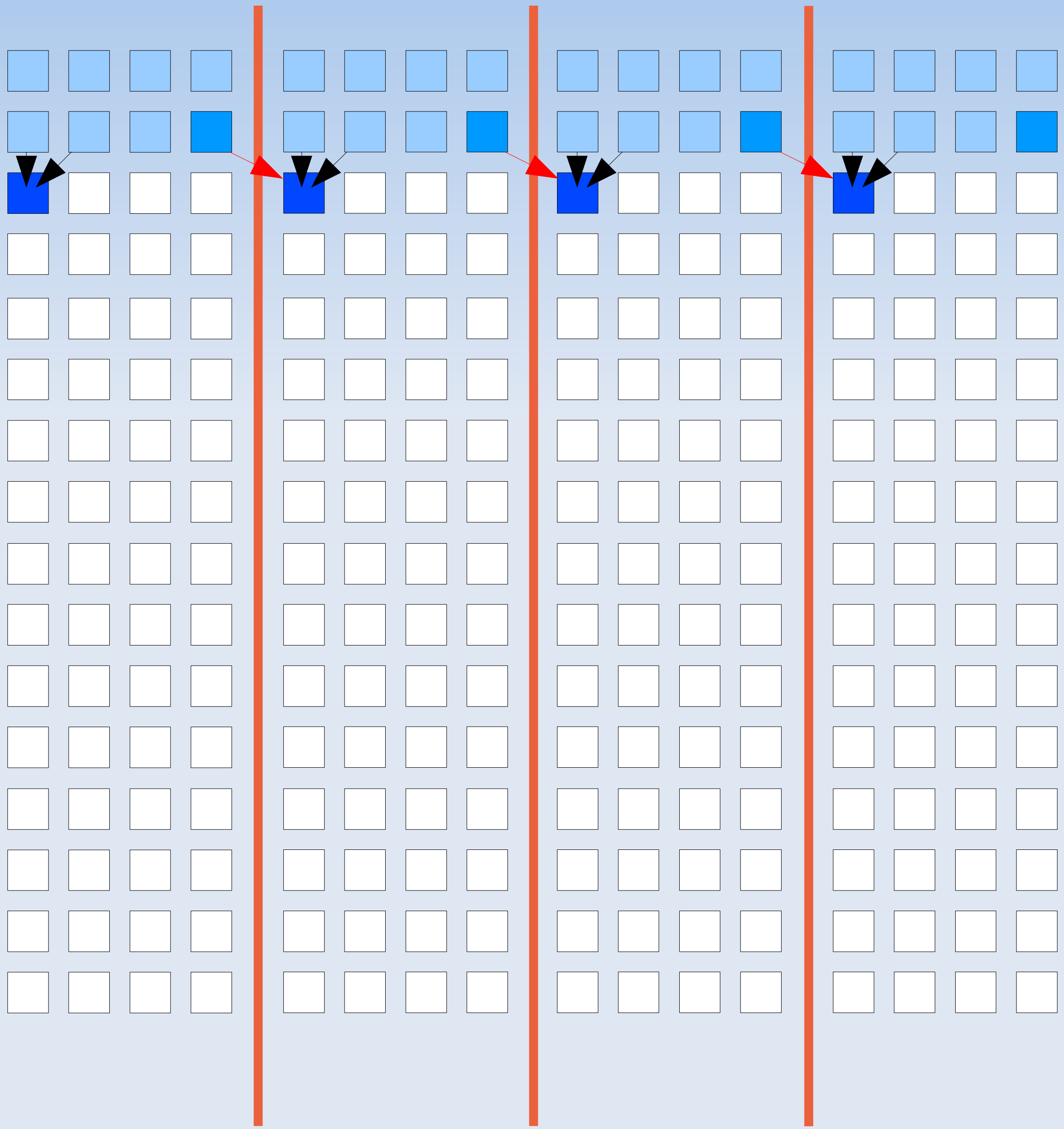
Parallelized Minpath



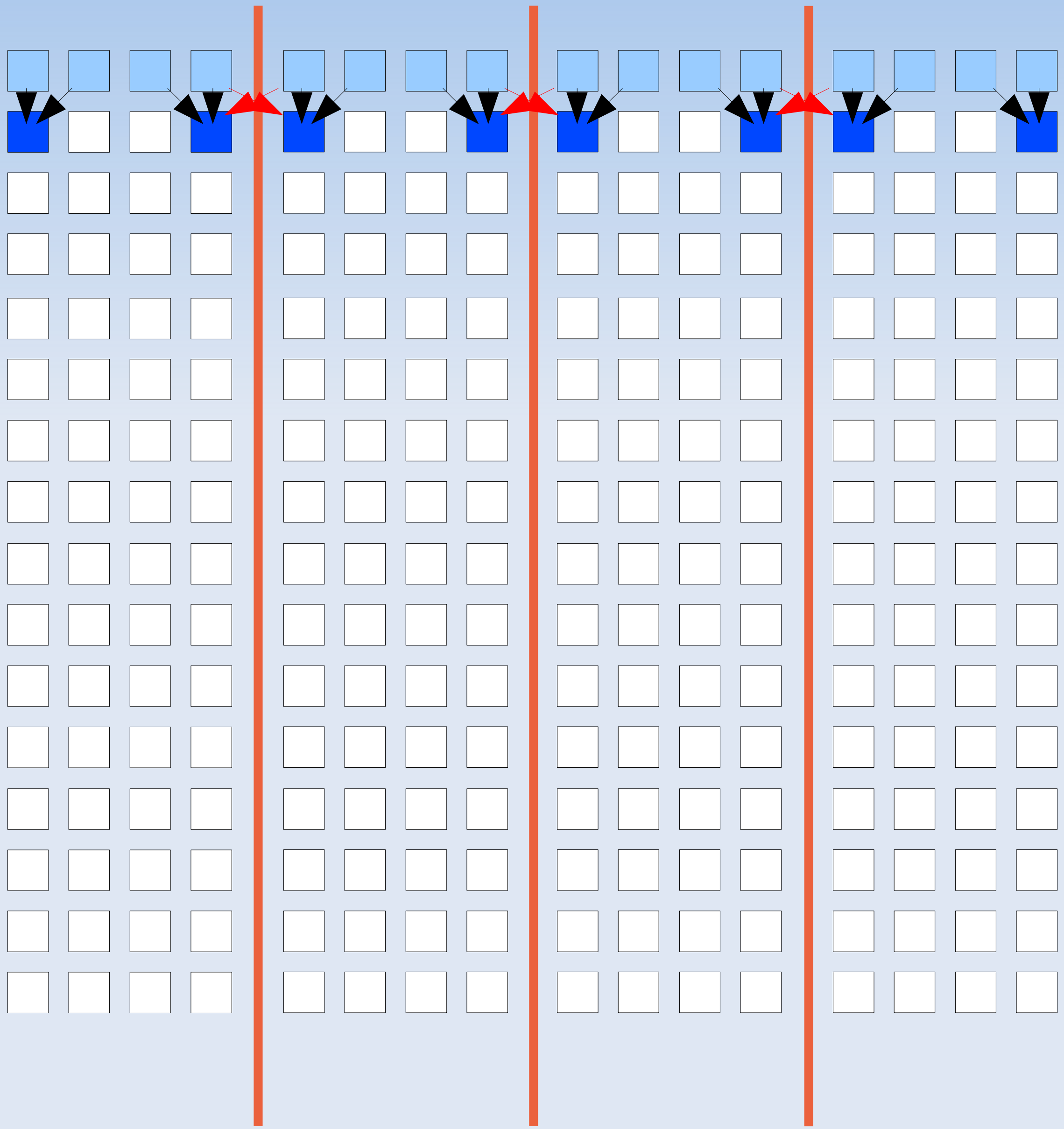
Parallelized Minpath



Parallelized Minpath



Parallelized Minpath, Take 2



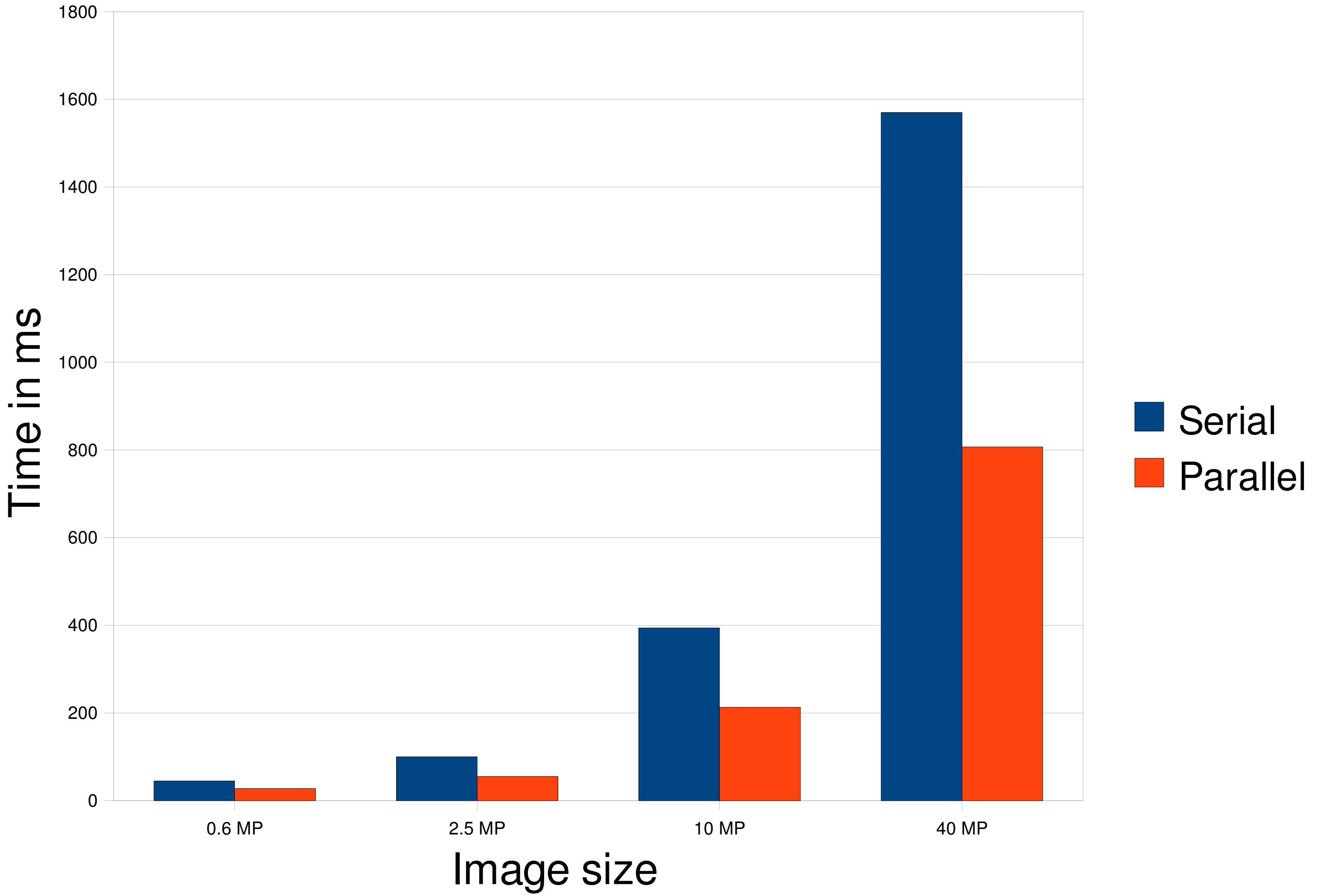
Algorithm Analysis

- Energy Calculation
  - Serial time: O(width * height)
  - Parallel:
    - Time: O(width * height / P) (P = number of processors)
    - Communication: O(height * P) (Initially)

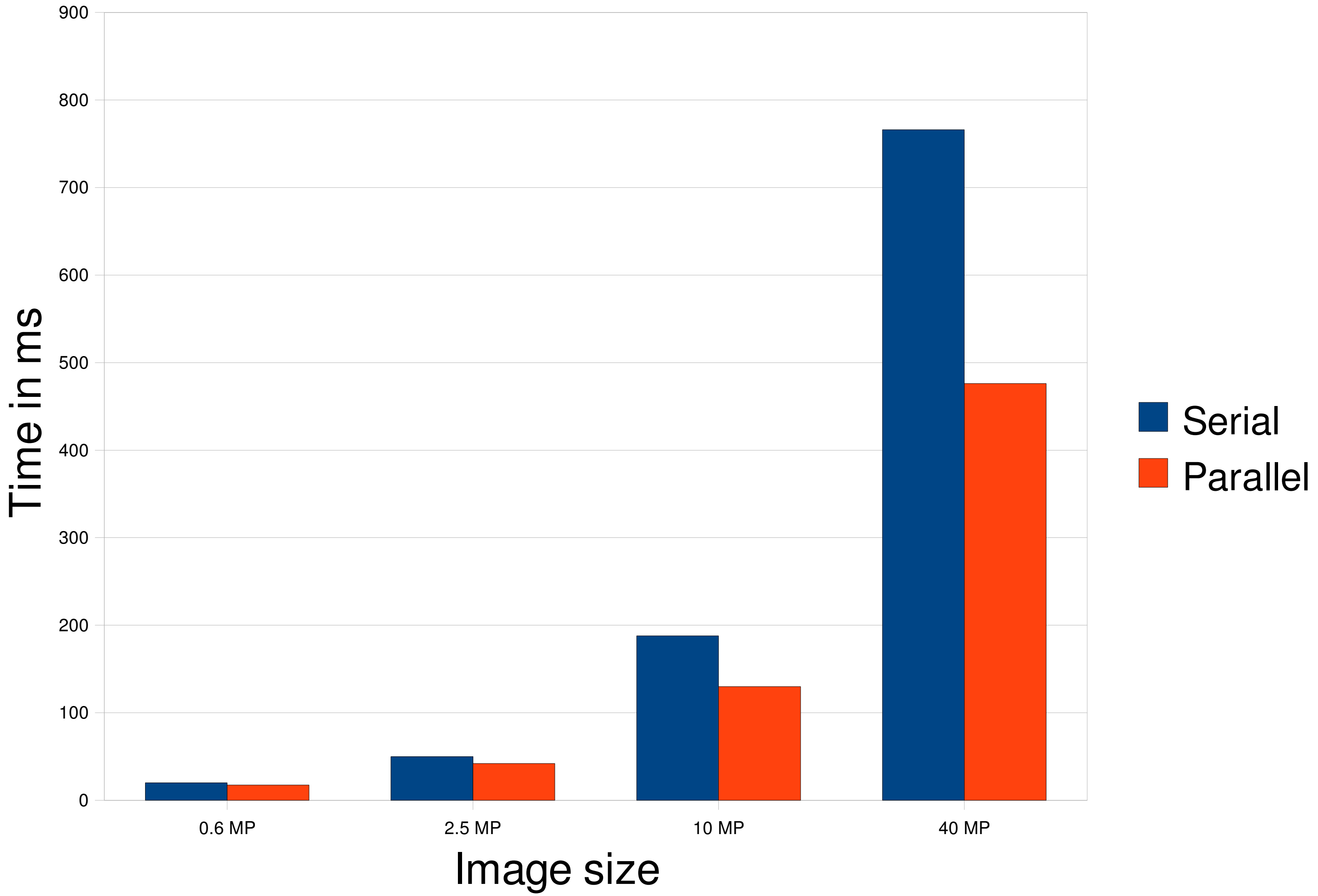
Algorithm Analysis

- MinPath Calculation
  - Serial time: O(width * height)
  - Parallel:
    - Time: O(width * height / P) (P = number of processors)
    - Communication: O(height * P) (Synchronized)

Energy


1800
1600
1400
1200
1000
800
600
400
200
0
Time in ms
0.6 MP
2.5 MP
10 MP
40 MP
Image size
Serial
Parallel

Minpath


900
800
700
600
500
400
300
200
100
0
Time in ms
0.6 MP
2.5 MP
10 MP
40 MP
Image size
Serial
Parallel

Conclusions

- Easily parallelized
- Communication limited smaller image sizes
- Effectiveness of algorithm varies widely

Future Work

- Test with more processors
- Photoshop / The GIMP Integration

- More features to parallelize:
  - Feature removal
  - Multi-size images