Parallel Mean Shift Accuracy and Performance Trade-Offs

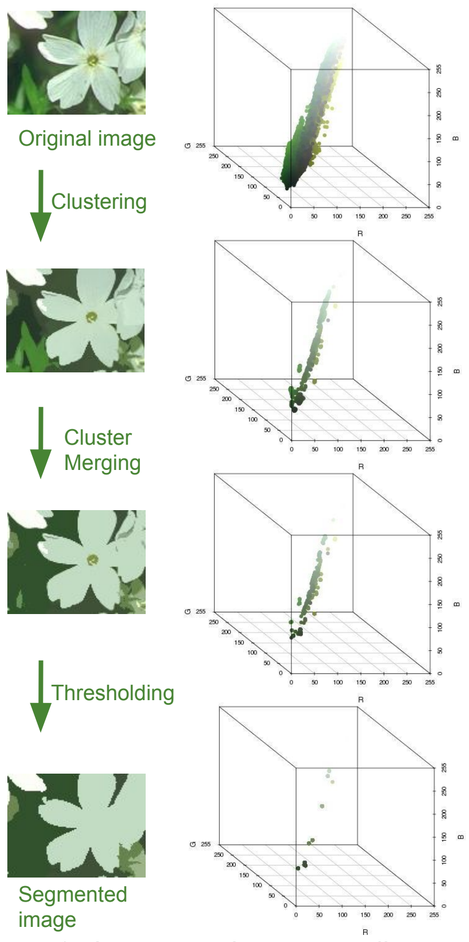
Student: Kirsty Duncan
Supervisors: Greg Michaelson & Robert Stewart

Aim:

To decompose the algorithmic parameters that affect the accuracy and parallel runtimes of mean shift segmentation.

The joint spatial-range domain is represented by the image space, with feature space information associated with each point.

Mean Shift Segmentation:



1. Images and corresponding feature spaces at each stage of the segmentation algorithm.

Clustering: For each point;

1. define window of radius w_r in the range dimensions and w_s in the spatial dimensions around this point, and calculate the mean shift vector;
2. if the vector is non-zero, move to the point and return to step 1;
3. if the vector is zero, this point is the peak of the original point.

All points with the same peak are merged in the output, forming clusters of peaks around the space.

Cluster Merging: To merge each cluster into a single region. All peaks within w_s & w_r of each other are merged.

Thresholding: If a region contains less than a threshold number of pixels, merge it with the closest region of above- threshold size.

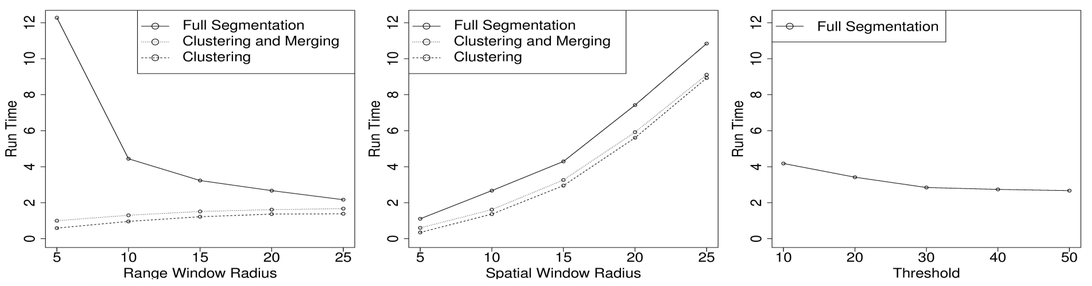
Experiment:

We identify three parameters as having an effect on mean shift segmentation computational cost and accuracy of the output:

1. **Range window radius w_r**
2. **Spatial window radius w_s**
3. **Threshold**

Run time and **Probabilistic Rand Index** accuracy were measured for various values of the above parameters on images from the *Berkeley Segmentation Dataset and Benchmark*.

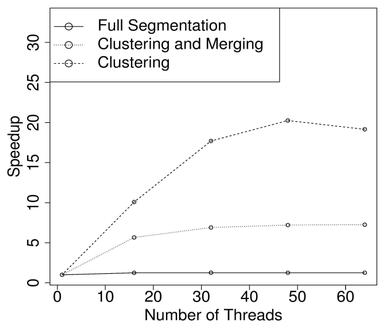
Results



2(a) RT vs. range window
2(b) RT vs. spatial window
2(c) RT vs. threshold
2(e) PRI vs. range window
2(f) PRI vs. spatial window
2(g) PRI vs. threshold

Range Window Size:

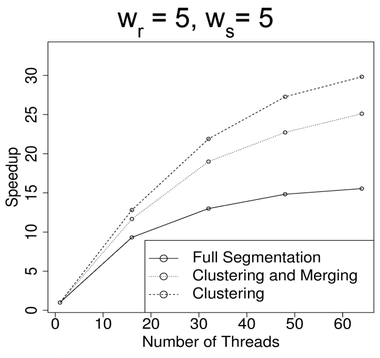
- Large w_r increases computational costs of clustering. Small w_r greatly increases the cost of thresholding.
- A w_r of 20 is desirable, as the PRI decreases significantly for larger radius, and the run time levels off.



3(a) Speedup v Threads $w_r = 5$, $w_s = 5$

Spatial Window Size

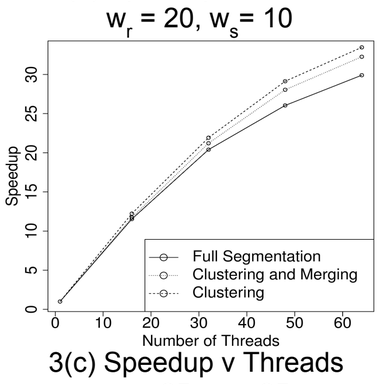
- Run time increases with the square of w_s.
- A w_s of 10 is desirable, as the PRI does not increase substantially by increasing w_s past 10.



3(b) Speedup v Threads $w_r = 20$, $w_s = 10$

Thresholding

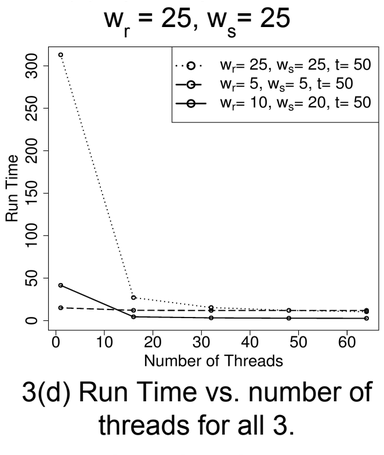
- Increasing the threshold decreases the number of small superfluous segments, increases performance without affecting the PRI, while improving the visual output.



3(c) Speedup v Threads $w_r = 25$, $w_s = 25$

Parallelism

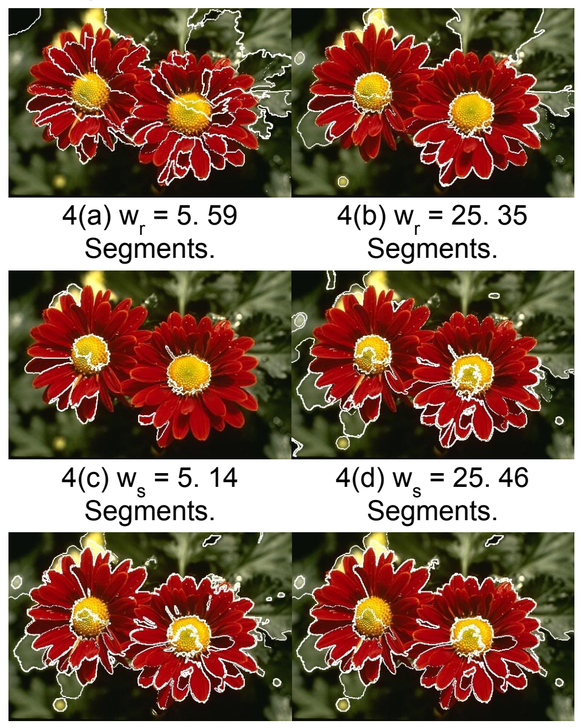
- When both windows are small, thresholding dominates the run time.
- When both windows are large, clustering dominates; however the sequential run time is an order of magnitude larger than for small or medium windows.
- For parallel runs, the sweet spot outperforms the other cases significantly.
- The thresholding step is optional. Considering the significant increase in runtime, and decrease in the PRI for large w_r and small w_s, thresholding is only worthwhile in application specific contexts.



3(d) Run Time vs. number of threads for all 3.

Visual Results:

- For small window radii, the thresholding phase over- compensates.
- Thresholding has a subtle effect for reasonable window sizes despite its significant run time.



4(a) w_r = 5. 59 Segments.
4(b) w_r = 25. 35 Segments.
4(c) w_s = 5. 14 Segments.
4(d) w_s = 25. 46 Segments.
4(e) threshold 10. 117 Segments.
4(f) threshold 50. 44 Segments.

Conclusion:

We analysed the mean shift segmentation algorithm when performed on the image space, rather than the feature space. Effects of varying the algorithmic parameters on the run time and PRI were measured and a sweet spot between the two was identified.

Implementing the algorithm in this way reduces its complexity and allows wider feature space windows to be used without a polynomial increase in run time.