INTRODUCTION
The algorithm proposed by Rachel Silva computes the count of the minimum (s, t)-cuts in an undirected weighted planar graph without preprocessing the graph using a max-flow algorithm. It can be used for image segmentation, where each minimum (s, t) -cut in the planar graph representing the image is equivalent to a distinct segmentation in an image. In this project Rachel Silva’s algorithm is implemented for image segmentation and experiments are performed to evaluate the performance of the algorithm and to prove the hypothesis that ‘Though for most images with identical segmentations, generating a single cut might be sufficient; but there exist images with a number of distinct segmentations, for which it is necessary to count and sample through the minimum (s, t) -cuts.’

ALGORITHM

0. Given an undirected weighted planar graph G.
1. Make s & t incident to single faces f₁ & f₂, respectively.
2. Construct dual of G’ where f₁ → s₂ and f₂ → t₂ & find the right-most shortest path p between s₂ & t₂.
3. Make copy uᵢ for each vertex vᵢ in p, which is connected to vertex x positioned below p for all the edges (x,vᵢ), which are then deleted.
4. Count the shortest paths between (uᵢ,vᵢ) pairs.

Time Complexity: \( O(V^2 \log V) \).

IMPLEMENTATION

Programming languages used:
The algorithm was initially implemented in Python and was later implemented in Java, as the implementation in Java was a lot faster than the implementation in Python.

Probabilistic sampling:
The number of minimum (s, t) -cuts in an image can be exponential, and thus it is necessary to sample through them. Choosing an arbitrary might not always give the best result. Thus, probabilistic sampling is used where a minimum (s,t) -cut is chosen by initially choosing the \((u,v)\) pair probabilistically and then probabilistically choosing the shortest path between the \((u,v)\) pair.

Scribbles for source and sink:
Instead of choosing a single pixel each for source and sink, a better image segmentation can be achieved by choosing multiple pixels (scribbles) for the source and the sink.

Edge weight function:

\[
Luminance(R,G,B) = \frac{1}{(\text{LumDiff}+1)} * 1000 \\
\text{MaxDiff} = (0.7152*G) + (0.2126*R) + (0.0722*B)
\]

The maximum threshold MaxDiff chosen for LumDiff is 20, as there is no need to consider any difference in luminance more than 20 due to the nature of images in real world.

RESULTS

- The algorithm does not produce good quality segmentations for images having similar colored object and background.
- If the object is very close to the border, the segmentation goes outside the border rather than segmenting the object completely.
- The quality of the image segmentation decreases with decrease in image’s resolution.
- The count of minimum (s,t)-cuts increases as the maximum threshold MaxDiff for LumDiff decreases; whereas the quality of the segmentations remains constant for MaxDiff > 20 and it decreases as the MaxDiff decreases below 20.

EXPERIMENTATION

- Selecting multiple pixels (scribbles) rather than selecting a single pixel each for source and sink is very helpful in generating a better desired image segmentation as it enables customizing the segmentation by drawing outlines.
- Most of the images tested had identical segmentations, and a very few images had distinct segmentations. Though the distinction between the segmentations was not significant, these images are a proof that images with distinct segmentations do exist and thus, it is necessary to count and sample through the cuts.

FUTURE SCOPE

- Using edge weight functions that are based on the color differences rather than the gray-scale differences, might improve segmentation quality.
- Testing against many more images to yield more images with significant distinct minimum (s,t)-cuts to further strengthen the proof for the hypothesis.

CONCLUSION

Image segmentation using Rachel’s algorithm is much simpler to understand and implement; which can be used to count and probabilistically sample through the minimum (s, t)-cuts for images with multiple distinct minimum (s, t)-cuts.

References: