**Automatic seeded region growing for color image segmentation**

**Frank Y. Shih, Shouxian Cheng**

**Implemented (mostly) in MATLAB by Cole Webb**

This README focuses on the MATLAB version, which is closer to complete than the Python version. Notes for the Python version are at the bottom of this document.

It should be noted here, as in the MATLAB code, that this algorithm as written doesn’t actually work. I am confident that 90-95% of the code needed is here, I just couldn’t figure out the last bit of it.

**Steps to run code**

1. Fire up MATLAB, with the full set of libraries required for this class.
2. Edit the first line of code in ```segment.m``` to point to the location of the image you're analyzing.
3. Click Run, or type ```segment``` in the MATLAB CLI.

**Approximate time to run code**

For the image I used, which was 616x820, it took about 30 seconds to get through the first two components. Things fall apart after that.

**Summary of important code sections**

This project is built around segment.m, which is in six sections labeled as such in the code:

* Component 1, conversion to YCbCr
* Displaying Component 1
* Component 2, generating seeds
* Displaying Component 2
* Component 3, growing seeds into regions
* Component 4, merging regions

These correlate clearly to four distinct steps in the original paper.

This project also relies on five helper functions, each in their own file as MATLAB recommends.

* computeNeighbors.m: Finds all neighbor connections in a labeled regions matrix.
* computeRegionStats.m: Computes size and channel means for a 3-channel image
* findNeighbors.m: Returns values of 4-neighbors of a given pixel in a 1-channel image
* mergeRegions.m: Merges two regions.
* relativeEuclideanDistance.m: Computes the maximum relative Euclidean distance of a pixel to its 8-neighbors in a 3-channel image.

**Explanation of improvements**

No improvements were made. It is my personal opinion, however, that the best thing this algorithm could do is loan its region merging strategy to the watershed algorithm.

**Explanation of changes**

No changes were made.

**Explanation of data structures**

No data structures were designed for this algorithm. MATLAB works really well with matrices, so I ran with that.

I discussed this project with Nicholas Hedges, who is also working on this paper. He provided some valuable insight into how to work with the data structures. We both did our own work.

**Experimental results**

The tests that I ran were mostly focused on confirming that the seed generation component worked. I am of the opinion that this algorithm performs well with gradients and mild noise, but gets significant noise when there are many well-defined things in an image.

**Comparison of results**

Compared to the images in the paper, I can confirm that this algorithm does not work.

Cheap jokes aside, most of the figures that the paper used to demonstrate their algorithm included gradients and/or medium amounts of noise.

**Implementation analysis**

This implementation is, as I stated earlier, about 90-95% complete. I never understood MATLAB very well, I don’t understand Python as well as I thought I did, and if I had the whole thing to do over again, I would do a different algorithm and a different language all together.

But like I said in class, this paper looked like work, and it was, and it looked like fun, and it was.

I didn’t plan ahead enough to segment the entire image. My algorithm would have benefitted from padding the edges of the image, or in only considering neighbors of edge pixels that actually exist, instead of just ignoring the edge pixels.

**Download links**

The below link contains the image and outputs that my testing generated.

<https://drive.google.com/drive/folders/1PIhxgRO5Mw08SEnLhuuQrrAfN1vWIONS?usp=sharing>

**Difficult issues**

This project was a lot of work, even to get it to the state that it is now.

The biggest issue I ran into was understanding how MATLAB handles matrices. I frequently needed to find various entries in matrices, or build matrices of data about other matrices, and filtering and slicing matrices just didn’t work in my head for some reason. I’m sure that if I understood the underlying structures, how things are laid out in memory, that it would be easier, but I’m not there yet.

Also, constantly indexing from one instead of zero was different.

From there, it was just a million tiny misunderstandings. I’ve got the big idea, I’m just clear on all the little details.