COMP9517

Lab 4, T2 2019 (09/07/2019)

The goal of this lab is to experiment with popular segmentation techniques (week 5 lectures). Using the sample images, segmentation methods using Watershed and MeanShift are to be developed.

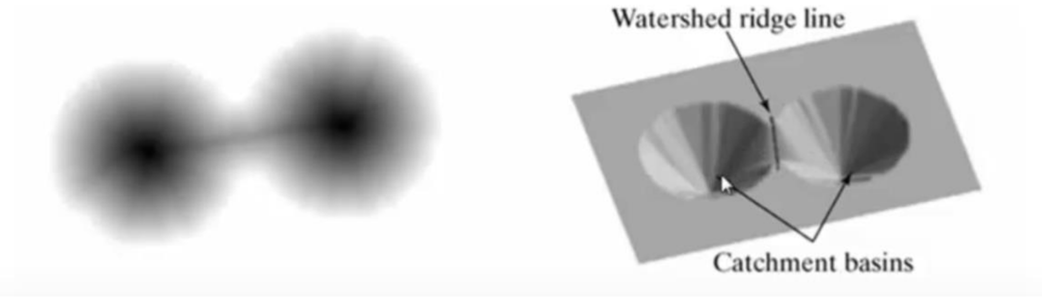
Code and results should be submitted via WebCMS3 for marking (2 marks). Submission is due at 23:59:59 on July 9th, 2019. Submission instructions will be posted prior to the lab.

Part 1: Segmentation

MeanShift, the "mode-seeking" algorithm is a clustering algorithm that assigns pixels to clusters by iteratively shifting points towards the mode, where the mode is the value with the highest number of datapoints (chioka.in). A visualisation can be seen here.

Watershed is a transformation which aims to segment the regions of interest in a grayscale image. This method is particularly useful when two regions of interest are close to each other—i.e, their edges touch.

This technique treats the image as a topographic map, with the intensity of each pixel representing the height. For instance, dark areas are considered to be 'lower' in height, and can represent troughs. On the other hand, bright areas are 'higher', acting as hills or as a mountain ridge (Vayada, D., 2018).

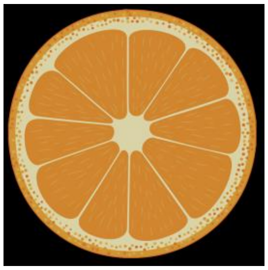


Visualising the watershed: Left image can be topographically represented as the image on the right. Source: (Agarwal, 2015)

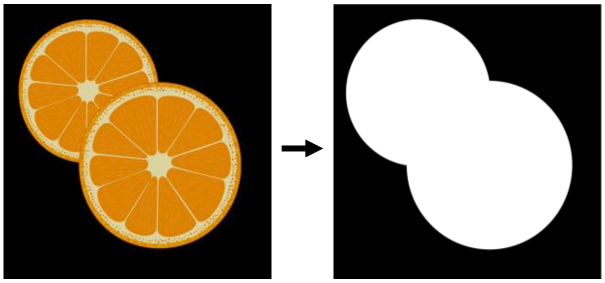
Image segmentation can be thought of as labelling pixels in an image. It is a large area of research in computer vision, and it comes in many different flavours: interactive segmentation, semantic segmentation, instance segmentation and many more. In this tutorial, we shall adapt the MeanShift clustering algorithms and the Watershed algorithm to solve unsupervised image segmentation.

Part 2: Questions

The sample images "orange_half.png" and "two_halves_binary.png" are used in the following questions.



orange_half.png

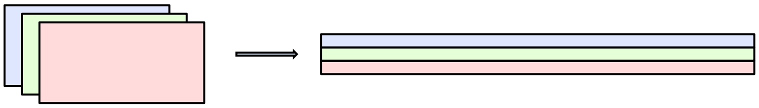


Perform global thresholding **two_halves_binary.png**

Question 1. Use the MeanShift algorithm to segment the two images.

Hint: Use MeanShift clustering in scikit-learn.

Step 1. Once you read the images into numpy arrays, extract each colour channel (R, G, B) so you can use each as a variable for classification. To do this you will need to convert the colour matrices into a flattened vector as depicted in the image below.



Step 2. You can then use the new flattened colour sample matrix (10000 x 3 if your original image was 100x100) as your variable for classification.
Step 3. Use the MeanShift's fit_predict function to perform a clustering and save the cluster labels (which we want to observe).

Question 2. Use a Watershed transformation to segment the grayscale versions of the two images.

Hint: Use Watershed segmentation from scikit-image.

Step 1. Convert the image to grayscale.
Step 2. Calculate the distance transform of the image.

Note: this is a vital step of the watershed algorithm. Visualising this step may help you understand how the algorithm works! If you have time, plot the result of the distance transform to see what's happening under the hood.

Step 3. Generate the watershed markers as the ‘clusters’ furthest away from the background. (This can be syntactically confusing so make sure to check out the example code in the link above)
Step 4. Perform watershed on the image. This is the part where the image is “flooded” and the water sinks to the “catchment basins” based on the markers found in step 3.

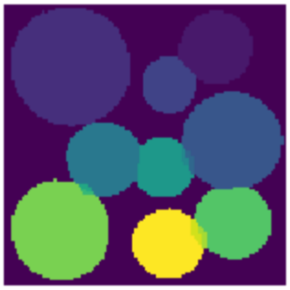
Question 3. *Assessable.* Submit the segmentation results from MeanShift and Watershed for marking. A plotting template has been provided so you can compare the two algorithms side by side. This will look something like:

Image
MeanShift Labels
Watershed Labels
?
?

Part 3: Extension Questions (not marked)

The sample images “coins.png” and “two_halves.png” are used in the following questions.

1. Use the MeanShift and Watershed algorithms on these images (as per part 2)
2. Notice that watershed doesn’t work very well for either image. Do some image manipulation to optimise watershed segmentation. The output of two_halves should look like the watershed output for two_halves_binary in part 2. The output for coins should look like:



If you get stuck (or just want to know the solution), look inside the *help* directory!

References

Agarwal, R. (2015). Segmentation using Watershed Algorithm in Matlab [Video]. Retrieved from https://www.youtube.com/watch?v=K5P5rjDiZzk

Meanshift Algorithm for the Rest Of Us, http://www.chioka.in/meanshift-algorithm-for-the-rest-of-us-python

Vayada, D. (2018). Intuitive image processing—Watershed segmentation. Medium.

Half orange image modelled after: https://www.vectorstock.com/royalty-free-vector/half-of-fruit-orange-vector-17862634