COMP9517: Computer Vision

2021 T3 Lab 4 Specification

Maximum Marks Achievable: 2.5

This lab is **worth 2.5% of the total course marks**.

The lab files should be submitted online.

Instructions for submission will be posted closer to the deadline.

Deadline for submission is Week 7, Monday 25 October 2021, 23:59:59.

Objective: This lab revisits important concepts covered in the lectures of Week 5 and aims to make you familiar with implementing specific algorithms.

Materials: The sample images and template code to be used in the tasks of this lab are available in WebCMS3. You are required to use OpenCV 3+ with Python 3+.

Submission: The tasks are assessable **after the lab**. Submit your source code as a Jupyter notebook (.ipynb) with output images (.png) in a single zip file via Moodle by the above deadline. The submission link will be announced in due time.

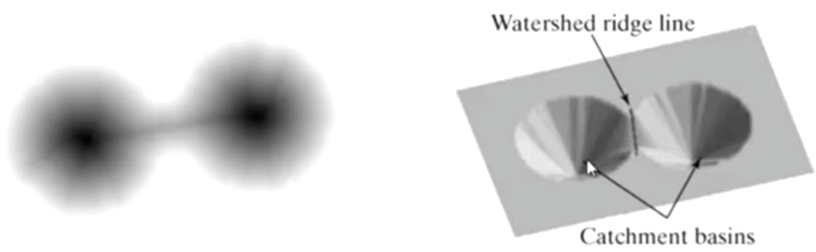
Image Segmentation

The goal of image segmentation is to assign a label to each pixel in an image, indicating whether it belongs to an object (and which object) or the background. It is one of the key research topics in computer vision and there are many different approaches: interactive segmentation, semantic segmentation, instance segmentation, and more.

In this lab the MeanShift clustering algorithm and the Watershed algorithm will be used to solve unsupervised image segmentation.

MeanShift is a clustering algorithm that assigns pixels to clusters by iteratively shifting points towards the modes in the feature space, where a mode is a position with the locally highest number of data points (highest density). A visualisation can be seen here.

Watershed is a transformation that 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; that is, their edges touch. It 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' and act as troughs, whereas bright areas are ‘higher’ and act as hills or a mountain ridge.

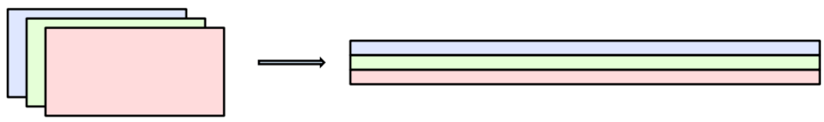


Visualising the Watershed: The left image can be topographically represented as the image on the right. Adopted from [Agarwal 2015](#).

Task 1 (1 mark): Use the MeanShift algorithm to segment images.
Images to be used for this task: **clever-shapes.png** and **coke-bottles.png**

Hint: Use MeanShift clustering from scikit-learn.

Step 1. Once you have 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. Then you can use the new flattened colour sample matrix (e.g. 10,000 x 3 if your original image was 100 x 100) as your variable for classification.

Step 3. Use the MeanShift `fit_predict()` function to perform a clustering and save the cluster labels, which we want to observe.

Submit the segmented images.

Task 2 (1 mark): Use Watershed transformation to segment images.
Images to be used for this task: **clever-shapes.png** and **coke-bottles.png**

Hint: Use Watershed segmentation from scikit-learn.

Step 1. Convert the images 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. Plot the result of the distance transform to see what is happening under the hood.

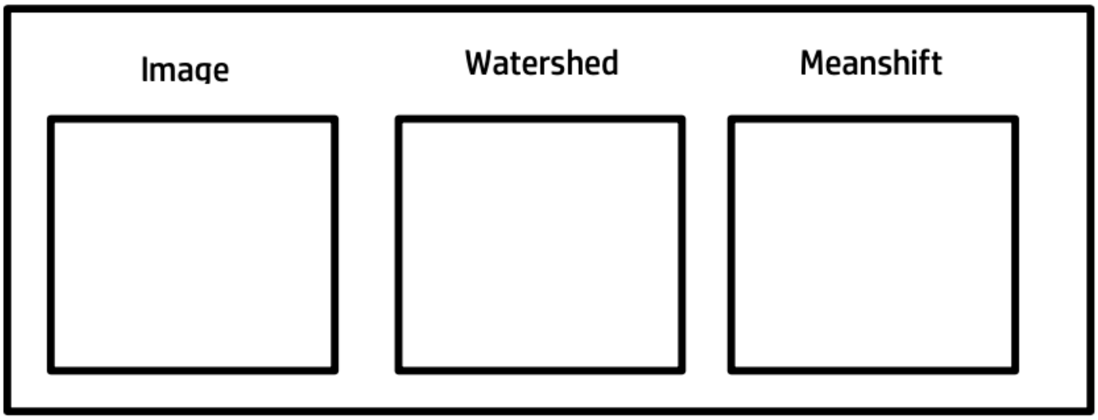
Step 3. Generate the Watershed markers as the ‘clusters’ furthest away from the background. This can be syntactically confusing, so make sure to check the example code in the link above.

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.

Submit the segmented images.

Task 3 (0.5 mark): Compare MeanShift and Watershed segmentation results.
Image to be used for this task: **blueberries.png**

Step 1. Apply the MeanShift and Watershed algorithms on the image as in the previous tasks and show the original images and segmentation results side by side using the provided template code. This will look something like this:



Step 2. Notice that Watershed does not work very well for this image. Do some image manipulation to improve the Watershed segmentation.

Submit the segmented images from both steps.

Coding Requirements and Suggestions
In your Jupyter notebook, the input images should be readable from the location specified as an argument, and all output images and other requested results should be displayed in the notebook environment. All cells in your notebook should have been executed so that the tutor/marker does not need to execute the notebook again to see the results.

References

[Segmentation using Watershed Algorithm in Matlab]

[Meanshift Algorithm for the Rest of Us (Python)]

Released: 12 October 2021