COMP9517

Lab 4, T1, 2021

The lab files should be submitted online.

Instructions for submission will be posted closer to the deadline.

Deadline for submission is Week 6, Monday 22 March 2021, 16:59:59 AEDT

Objectives: This lab presents a revision of important concepts from weeks 4 and 5 and aims to make you familiar with implementing specific algorithms.

Materials: The sample images to be used in this lab are available in WebCMS3 under Lab → Lab4, along with this lab spec. You are required to use OpenCV 3+ with Python 3.

Submission: Question 3 is assessable **after the lab** and is **worth 2.5% of the total course marks**. Please submit your Jupyter notebook with the solutions, intermediate steps and final output images via webCMS3 by the deadline. Submission link will be opened just prior to the lab session.

Background information:

1 Image Segmentation

Image segmentation can be thought of as labelling pixels in an image. It is an important research topic in computer vision and comes in many different flavours: interactive segmentation, semantic segmentation, instance segmentation, and many more. In this lab the MeanShift clustering algorithm and the Watershed algorithm will be used to solve unsupervised image segmentation.

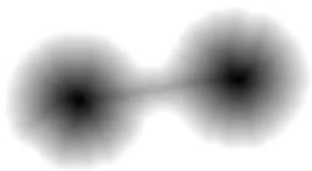
1.1 MeanShift

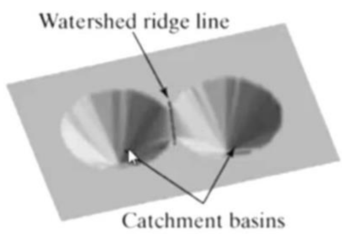
It is a clustering algorithm that assigns pixels to clusters by iteratively shifting points towards the modes in the feature space, where a mode is a position with the locally highest number of data points (highest density). A visualisation can be seen here.

1.2 Watershed

It 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. Bright areas are 'higher', acting as hills or as a mountain ridge.





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

QUESTION 1: WATERSHED

Given the two input image ball_set.jpg, use Watershed transformation to segment the grayscale version of the image.
Hint: Use [Watershed segmentation](#) from scikit-learn.

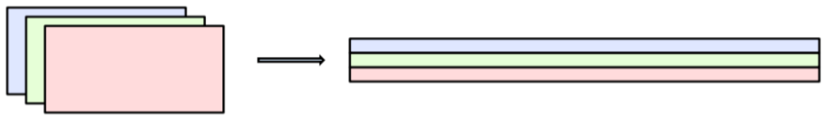
1. Convert the image to grayscale.
2. Calculate the distance transform of the image. Note: this is a vital step of the Watershed algorithm. Visualising this step may help you understand how the algorithm works! Plot the result of the distance transform to see what is happening under the hood.
3. Generate 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.
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 the above Step.

QUESTION 2: MEANSHIFT

Use the MeanShift algorithm to segment the input image ball_set.jpg.

Hint: Use [MeanShift clustering](#) from scikit-learn.

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.



2. Then you can use the new flattened colour sample matrix (width x height x channels) (10,000 x 3 if your original image was 100px x 100px) as your variable for classification.
3. Use the MeanShift fit_predict function to perform clustering and save the cluster labels, which we want to observe.

QUESTION 3: Watershed and Meanshift

Apply the segmentation algorithms MeanShift and Watershed on the planets.jpg image for marking. A plotting template has been provided so you can compare the two algorithms side by side. The following format for comparison should be used for the submission: