

# COMP9517: Computer Vision

## 2020 Term 2

### Assignment Specification

### Maximum Marks Achievable: 10

This assignment is **worth 10% of the total course marks**.

The assignment files should be submitted online.

Instructions for submission will be posted closer to the deadline.

**Deadline for submission is Week 3, Friday 19 June 2020, 23:59:59.**

**Deliverables:** You should submit a report (maximum 5 pages) explaining the approach you have taken in Tasks 1, 2 and 3, and including the sample input images and the intermediate and final output images obtained. You must also submit your Python source code files.

**Submission:** The assignment files (report and source code) should be submitted online via WebCMS3. Instructions for submission will be posted closer to the deadline.

**Tip:** You are strongly advised to use OpenCV 3+ with Python 3+. Jupyter notebook files are preferred as source files for submission.

#### Objectives:

This assignment is aimed at familiarisation with basic image processing methods. It also introduces you to common image processing and analysis tasks using OpenCV. After completing this assignment, you will know how to:

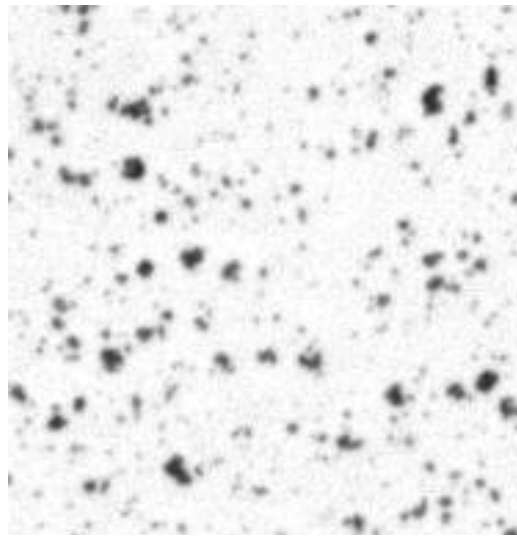
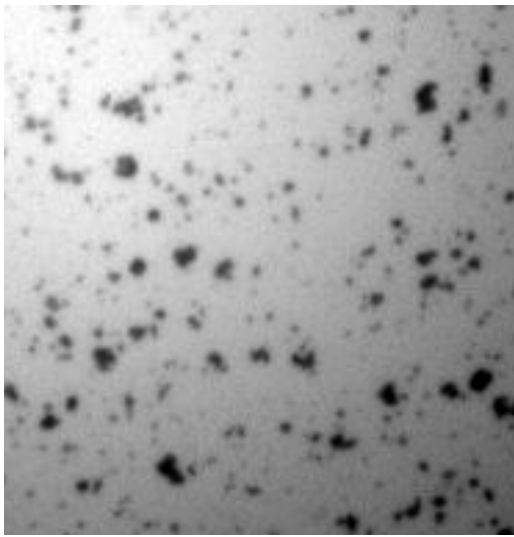
1. Open and read image files.
2. Display and write image files.
3. Perform mathematical operations on images.
4. Apply basic image filtering operations.
5. Perform image adjustment and restoration.

#### Description:

Many computer vision applications use image processing techniques to suppress artifacts and enhance relevant features to prepare images for further quantitative analysis.

The goal of this assignment is to create an image processing algorithm that can open a digital image, perform a sequence of pixel manipulation operations (step-by-step as listed under **Tasks**), in order to remove background shading artifacts.

Below is an image (on the left) from a biological experiment in which the researchers want to measure several properties of the dark particles. But these measurements are impeded by the shading in the background. Thus, the shading needs to be removed (as on the right).



### Tasks:

This assignment consists of three tasks, described below. Each task depends on the previous one, so you need to do them in the given order.

#### **Task 1 (4 marks): Background estimation**

Open the given image **Particles.png** which, similar to the example shown above, has a background shading pattern that needs to be removed. The first step to get rid of the shading is to make an accurate estimate of the background of the image.

Write an algorithm that performs the following two steps. First, it should create a new image, let us call it image A, with the same size (number of pixel rows and columns) as the input image, which we call I. Second, the algorithm should go through the pixels of I one by one, and for each pixel (x,y) it must find the **maximum** gray value in a neighbourhood around that pixel, and write that maximum gray value in the corresponding pixel location (x,y) in A. The resulting image A is called a **max-filtered** image of input image I.

The neighbourhood should be of size  $N \times N$  pixels, where  $N$  is a free parameter of the algorithm. Experiment with different values of  $N$  and mention in your report what is the smallest value of  $N$  that causes the dark particles in I to disappear altogether in image A. Also explain why this value of  $N$  causes the particles to disappear.

The max-filtering causes the gray values in A to be higher than the actual background values in I, so a correction is needed. Extend your algorithm to create another image, which we call image B here, of the same size as I and A. Now let the algorithm go through the pixels of A one by one, and for each pixel (x,y) find the **minimum** gray value in an  $N \times N$  neighbourhood around that pixel, and write that minimum gray value in (x,y) in B. The resulting image B is

called a **min-filtered** image of the image A.

In your report, include image B computed from **Particles.png**.

### **Task 2 (2 marks): Background subtraction**

Now that your algorithm can estimate the background B of an image I, removing the shading artifacts from I can be done simply by subtracting B pixel by pixel from I, resulting in the output image O. Extend your algorithm to perform this subtraction.

In your report, include image O computed from **Particles.png**.

### **Task 3 (4 marks): Algorithm generalisation**

Open the given image **Cells.png** which, similar to **Particles.png**, has a background shading pattern that needs to be removed. There are three main difference between the two images: 1) the sizes of the images are different; 2) the sizes of the objects (cells versus particles) are different; 3) in **Cells.png** the objects are bright and the background is dark, whereas in **Particles.png** the objects are dark and the background is bright.

Make sure your algorithm can deal with input images of arbitrary size. Dealing with larger objects in images is a matter of changing the value of the neighbourhood parameter N. But as you will see, your algorithm will not be able to remove the shading from **Cells.png**. To make your algorithm work for that image, you need to reverse the max-filtering and min-filtering.

Extend your algorithm with another free parameter, named M. If the user sets  $M = 0$ , the algorithm should perform max-filtering (image I to A), then min-filtering (image A to B), then subtraction ( $O = I - B$ ). And if the user sets  $M = 1$ , the algorithm should perform first min-filtering, then max-filtering, then subtraction.

In your report, explain why **Particles.png** requires  $M = 0$ , and **Cells.png** requires  $M = 1$ . Also mention what is a good value of N for **Cells.png** and why. And include images B and O computed from **Cells.png** in your report.

**Copyright:** UNSW CSE COMP9517 Team

**Released:** 5 June 2020