The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, SPRING SEMESTER 2018-2019

INTRODUCTION TO IMAGE PROCESSING

Time allowed ONE hour

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced

Answer ALL THREE Questions

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn your examination paper over until instructed to do so

ADDITIONAL MATERIAL: **NONE**

INFORMATION FOR INVIGILATORS: **NONE**

1. Image Compression

(a) List the three types of redundancy that can be exploited by image compression techniques.

(3 marks)

Coding redundancy
Spatial redundancy
Psychovisual redundancy

(3 marks)
[knowledge]

(b) Explain with the aid of an example how run-length encoding can be used to reduce the amount of memory needed to store a binary image. What type of redundancy does run-length coding exploit?

(7 marks)

Run length encoding, in it pure form, would store a set pairs of the form (0,N) or (1,M), signifying N 0s or M 1s. A common extension for binary images is to reduce storage space by taking advantage of the fact that 0 must be followed by 1, and vice versa, so (0, N), (1,M) becomes 0, N, M. For full marks the student should mention both approaches and show a correct example. [6 marks]

Run length encoding exploits spatial redundancy [1 mark]

(7 marks)
[knowledge]

(c) An image has the following normalized histogram. Derive a Huffman code for each pixel value, showing how you obtained your code.

Pixel value	Normalised Frequency
0	0.35
1	0.2
2	0.15
3	0.1
4	0.1
5	0.05
6	0.05
7	0

(10 marks)

Huffman coding builds a binary tree in which symbols to be coded are nodes.

The algorithm:
Create a list of nodes, one per for symbol, sorted in order of symbol frequency (or probability)
REPEAT (until only one node left)

{
Pick the two nodes with the lowest frequencies/probabilities and create a parent of them
RANDOMLY assign the codes 0,1 to the two new branches of the tree and delete the children from the list
Assign the sum of the parents' probabilities to their parent and insert it in the list
}
The path from root to node gives the code for corresponding symbol

Because there is a random component to the algorithm, many different codings are possible. For full marks the student should apply this method correctly to the data provided

(10 marks)
[application]

2. Intensity Transforms and Filtering

(a) What is gamma correction, when is it needed, and why?

(4 marks)

Gamma correction is a point transform which raises each pixel value to a power: $g(x,y) = f(x,y)^{\gamma}$

It is needed when displaying images on a screen. This is because the hardware used effectively applies an intensity transform. When a voltage V, proportional to image intensity, is sent to display hardware the intensity displayed is $L \approx V^{2.5}$ depending on device. Gamma correction modifies the voltage sent so that the hardware displays the desired intensity. In this case $\gamma = 1/(2.5)$.

(4 marks)
[knowledge]

(b) In the context of linear intensity transforms, what are gain and bias? How do they affect a) the appearance of an image and b) its histogram?

(6 marks)

In a linear intensity transform $g(x,y) = a.f(x,y) + b$. a is the gain, b the bias. Gain controls the contrast, b the brightness of the image. Changing the gain stretches the histogram in the x-direction, and flattens it. Changing the bias shifts the image's histogram, but maintains its shape.

(6 marks)
[knowledge]

(c) Explain how you convert a very dark image whose grey levels were in the range 10–50 to one that made use of the full range of available pixel values (0-255). What is this process called?

(4 marks)

$G(x,y) = 0 + (f(x,y) - 10)((255 - 0)/(50 -10))$
This is called contrast stretching.

(4 marks)
[application]

(d) Compute the result of applying

i) a 3 x 3 mean filter
ii) a 3 x 3 Sobel filter measuring gradient in the vertical direction (i.e. highlighting horizontal lines)
iii) a 3 x 3 median filter

to the central pixel of the image fragment shown below

7	8	4
3	6	8
8	5	1

(6 marks)

Mean filter = (7 + 8 + 4 + 8 + 6 + 3 + 8 + 5 + 1)/9 = 5.5
Sobel = (1x7 + 2x8 + 1x4 – 1x8 – 2x5 – 1x1) = 27 -19 = 8
Median = 1 3 4 5 ***6*** *7 8 8 8 = 6*

(6 marks)
[application]

3. Segmentation

(a) Consider the image fragment below (in which the numbers represent intensity values)

3	4	2	2
2	3	3	1
1	2	3	2
1	1	2	3

Using letters of the alphabet to label regions, show how this image fragment would be segmented by the *Watershed* algorithm.

(10 marks)

For full marks the student should show how the process proceeds; sorting the pixels into ascending order on value and simulating the flooding process. As there are pixels with the same value, ordering in the list and so the final result may vary.

(10 marks)
[knowledge/application]

(b) Why can the sort process incorporated in the Watershed algorithm be computed in linear time?

(2 marks)

Linear time sorting is possible because we have digital images, with a discrete number of grey values. This means there are a limited number of gradient values we can compute. A bin sort can therefore be used.

(2 marks)
[understanding]

(c) In most situations image gradients are computed and used to form the terrain input to the Watershed method. Why is the gradient magnitude an appropriate function to use in this way?

(2 marks)

A gradient is appropriate as we want edges in the image to form the boundaries between regions, and regions to consist of uniform areas in the image. Since gradient is high at edges and low in uniform areas it gives a good surface for immersion simulations such as watersheds.

(2 marks)
[understanding]

(d) Image texture and noise often mean that a gradient-based terrain contains many local minima. Describe how this might adversely affect a watershed-based segmentation and how these problems could be reduced.

(6 marks)

Noise makes the watershed algorithm over-segment; i.e. produce very many small regions that are defined more by the noise than the underlying image structure.

This can be addressed in a number of ways, and any plausible approach will recieve marks. Common methods are to smooth either the original image or the gradient image with e.g. a gaussian filter, or use a differential operator that incorporates such smoothing. Alternatively, the basic watershed rules can be modified so that e.g. no new pools are formed above a certain level.

(6 marks)
[understanding]