Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code:	CM3113
Module Title:	Computer Vision
Lecturer:	Prof. Paul Rosin
Assessment Title:	Stroke-Based Image Rendering
Assessment Number:	1
Date Set:	Tuesday 23rd October 2018 (week 4)
Submission date and Time:	20th November 2018 at 9:30am (week 8)
Return Date:	14th December 2018 (week 11)

This coursework is worth 30% of the total marks available for this module. The penalty for late or non-submission is an award of zero marks.

Your submission must include the official Coursework Submission Cover sheet, which can be found here: `https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf`

Submission Instructions

Description		**Type**	**Name**
Cover sheet	Compulsory	One pdf file	`[Student number].pdf`
Source code	Compulsory	One or more Java or C files packaged as a single zip file	`Code_[Student number].zip`

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part.

Assignment

Design a Java or C program that performs stroke-based rendering of an image, i.e. it “paints” a picture based on an input image. The inspiration for the coursework comes from this conference paper (`http://users.cs.cf.ac.uk/Paul.Rosin/CM3113/paint-relax.pdf`) but I have simplified the goals for you.

The program takes two inputs: a colour PPM format image to be copied in a painterly way, and a PGM image mask of the brush stroke (black = brush, white = background, i.e. non-brush). You can assume that the brush stroke is aligned horizontally (i.e. width of stroke is > height of stroke). The main output is a PPM image - the “painted version” of the input.

The program applies the brush at different scales and orientations to the image, based on the input image’s edge magnitude and orientation. Edge magnitude and orientation should be calculated by:

- converting the colour image to gray level
- blurring the image
- applying the Sobel edge detector.

The input brush is copied:

- to create various versions at $S = 5$ different sizes, scaled by factors $1/S, 2/S, \dots S/S = 1$; just use the simplest sampling scheme
- each of the scaled brushes is rotated around the centre of the image by regular increments through 360° degrees to generate $A = 16$ orientations.

Now start painting. The basic process is:

- start with an empty canvas (i.e. a black output image)
- apply strokes to this canvas in a coarse to fine (i.e. large to small) brush order
- for each brush size randomly generate N positions for the strokes in the image, where $N = P/D$, $P =$ number of pixels in image, $D =$ density; apply the brush at each position if
  - it is the right size; the size should be **inversely** related to edge magnitude, i.e. use the smallest brush at the strongest edges, and the largest brush at the weakest edges
  - applying the brush improves the canvas; i.e. adding that brush stroke makes the canvas **more** similar to the input image; measure dissimilarity as the summed absolute difference in RGB values of all corresponding pixels in the input and output images.

To make a brush stroke:

- get the brush at the correction orientation (i.e. closest to the edge orientation at that point in the image)
- compute the mean RGB (red, green, blue) values in the input image at the pixels specified by the brush stroke mask
- copy the mean RGB values into the canvas at the pixels specified by the brush stroke mask.

In addition to the final rendered version of the image generate the following intermediate output images:

- the magnitude and orientation computed by the Sobel edge detector (file names: magnitude.pgm, orientation.pgm)
- the full set of rotated and scaled brushes (file names: brush-s-a, where s is the size index and a is the orientation index)
- the strokes that make up the final rendered output, but rendered in separate images for each brush size (file names: intermediate-0.pgm, intermediate-1.pgm, … intermediate-4.pgm)

In order to gain higher marks you need to add one novel extension or additional feature (such as those suggested here):

- consider different strategies for determining the placement of strokes
- use a weighted brush that applies semi-transparent strokes
- deal with unpainted areas of the canvas

Your program should be able to run in the School's labs.

Please note: You should write all the code yourself, and not use third-party libraries.

Learning Outcomes Assessed

1. Understand the technical details of a range of techniques for image manipulation.

Criteria for assessment

Credit will be awarded against the following criteria.

1. 20% – For implementation of colour to gray conversion, image blurring and computation of Sobel edge detector.
2. 20% – For generation of rotated and scaled brushes.
3. 40% – For synthesis of brush strokes in the canvas at the appropriate location, scale and orientation, and with the appropriate colour.
4. 20% – For the design and incorporation of other features beyond the basic coursework specification.

The following is an indication of the level of attainment against the appropriate award:

1st	(70-100%)
2.1	(60-69%)
2.2	(50-59%)
3rd	(40-49)
Fail	(0-39%)

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on 14th December 2018 via Grade Centre. This will be supplemented with oral feedback in the lecture in week 11.