

## Homework 4 (version 4)

**Question 1** – You are given a grayscale image. Your objective is to colorize it with pseudo-colors, in such a way that dark areas appear purple, and bright areas appear bright orange.

a) You are free to select any **non-linear functions** you desire, as long as they provide the aforementioned appearance **and** they are **derivable** in their definition domain. Provide the graphical plots of the red(p), green(p) and blue(p) functions you use for colorization and their definitions; (p stands for the grayscale pixel value), as well as the colorized image. (10 points)

b) Prove that the colorization functions that you have selected are derivable in their definition domain. (10 points)

**Question 2** – The mpeg7ShapeB dataset (attached) is a standard shape dataset made up of 70 classes; each with 20 samples. The first 10 samples of each class are in the training folder and the remaining 10 in the testing folder.

Conduct a nearest neighbor classification using the Fourier coefficients of the shapes, and fill in a table such as the following the classification accuracies (%) that you obtain:

Number of coefficients	5	10	20	50	up to you...
Euclidean distance					
Manhattan distance					
Chi-square distance					

External libraries are prohibited. (40 points)

**Question 3** – Implement in Java the **watershed transform** (pp 276-277 of P. Soille's book). (20 points)

Then develop a segmentation pipeline based on the watershed transform for segmenting the attached coins image, so that every coin is represented in its entirety by exactly one region (20 points).

Good luck.

## Appendix

For those of you that continue using the Java library, some info:

Creating a color image of size 200x100 pixels:

```
Image color = new BufferedImage(200, 100, 3);
```

Accessing the color channel's pixel values at location x,y:

```
int red = color.getXYCByte(x, y, 0);  
int green = color.getXYCByte(x, y, 1);  
int blue = color.getXYCByte(x, y, 2);
```

Setting the color pixel value at location x,y to a new color [100,5,250]:

```
color.setXYCByte(x, y, 0, 100);  
color.setXYCByte(x, y, 1, 5);  
color.setXYCByte(x, y, 2, 250);
```

Creating a integer valued image of size 200x100 pixels, whose every pixel can have a signed 32-bit integer value:

```
Image intImage = new IntegerImage(200, 100, 1);
```

Accessing the value of the pixel at location x,y:

```
int val = intImage.getXYInt(x, y);
```

Setting the value of the pixel at location x,y to 155:

```
intImage.setXYInt(x, y, 155);
```