EXERCISES FOR IMAGE PROCESSING I
PROBLEM SHEET 5

Due date: 10.12.15 before 12:00h

Topics: Image Compression and Segmentation

Submission: Please send your solutions via email to seppke@informatik.uni-hamburg.de.

I THEORETICAL PROBLEMS 10 P.

1. (Lossless) Image Compression

When documents are archived, pixels are classified as either 'print' or 'background'. For 'print', one distinguishes between the colors black, red, yellow, green, and blue. Based on long-standing statistics, it is known that a document is covered with 'print' only by 10%. The colors occur with the following probabilities:

black: 80%
red: 5%
yellow: 1%
green: 2%
blue: 12%

a) What is the entropy of such documents?
b) Design a Huffman Code for the pixels.
c) What is the mean length of a code word?
d) What is the redundancy of the following 4-bit code:

black	red	yellow	green	blue	white
0000	1000	1001	1011	1100	0001

2. Image Segmentation

a) When may simple threshold-based segmentation approaches be applicable for image segmentation, when not? Give at least one example for each.
b) Does the bi-modality of the image histogram guarantee successful image segmentation if threshold-based image segmentation is applied and the threshold is selected in the minimum between the two histogram maxima?
c) Explain the need of component labeling for object detection after thresholding.
d) What is the most typical problem in edge detection approaches? How does the Canny-algorithm try to enhance the segmentation quality?

1. (Lossy) Image Compression

Use the Karhunen-Loève transform to code 2x2 images with pixels x_1, x_2, x_3, x_4 by means of 3 values y_1, y_2, y_3 such that the original images can be reconstructed with minimal error.

The pixels have zero mean and the following covariance matrix: $\frac{1}{4}\begin{pmatrix} 15 & 5 & 9 & 3 \\ 5 & 15 & 3 & 9 \\ 9 & 3 & 15 & 5 \\ 3 & 9 & 5 & 15 \end{pmatrix}$

a) Compute the 3 x 4 transform matrix A_3.

b) What is the mean quadratic error (MSE) for x_1' when it is reconstructed using A_3^T

Useful Python commands (more commands in the NumPy/SciPy reference):

```
import numpy as np                  #makes numpy accesable as 'np'
from numpy import linalg as la      #makes the Linear Algebra Module
accessable as 'la'
a = np.matrix([[a11, a12, ...],
                             [a21, a22, ...],...])
ew, ev = la.eig(a)
# ev[:,i] is the eigenvector to the eigenvalue ew[i]
# ev*ev.T is the matrix multipl. should give the id. matrix in this example
```

2. Operators for Edge detection

Implement the following operators for greyvalue images as input. Use the definitions of the algorithms in the slides 10-14 of lecture 11. The result may be different for each operator and thus is given separately below:

a) Robert's Cross operator → two images, one representing the gradient magnitude,
the other the gradient direction.

b) Sobel operator, with same output.

c) Kirsch operator, with same output.

d) Laplacian operator → one image of the strength of the second derivative of the image.

Apply all of your implemented operators to the image misc.lena() and compare the results. Which operator would you prefer for segmentation of this image?

Useful Python commands:

```
import numpy as np
from scipy import misc

img = misc.lena()

#try to write your filter functions (a-c) to return two values,
#the magnitude and the angle image.
def sobel(image):
     …
     return gradient_magnitude, gradient_angle

g_magnitude, g_angle = sobel(img)
```