EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Mathematisch-Naturwissenschaftliche Fakultät

Technische Informatik

Enkelejda Kasneci,
Thiago Santini
enkelejda.kasneci@uni-tuebingen.de,
thiago.santini@uni-tuebingen.de

Eye Movements and Visual Perception

Practical Session 4

Winter term 2017/18 Tübingen, 22. January 2018

Introduction

The aim of this practical session is to calculate and understand salency maps.

Submit your answers to ILAS on a single folder using the **Upload Multiple Files as Zip-Archive** option. This folder should contain all items marked in this document with (Answer in: example.txt) . Name the file **exactly** as specified in this document with your answer inside. No file upload is required for items not marked as such. **Upload a single submission per group.**

Exercise 1: Group Identification

Add the matriculation number of all group members, **one per line**. (Answer in: group.txt)

Exercise 2: Read image, convert it to gray scale and normalize it

1. Read the image "example.png".
2. Convert the image to gray scale $(Red + Green + Blue)/3$. Beware of saturation due to the type of the image data (*uint8*).
3. Normalize the color image such that each channel uses the color range (0-1). What is the formula?
4. Normalize the gray scale image to (0-1). (Answer in: normalized.png)

Exercise 3: Calculate the data for the saliency map

1. Brightness(Y) formula: $Y = 0.2126 * Red + 0.7152 * Green + 0.0722 * Blue$

2. Calculate the brightness for each pixel in the color image and display the respective values. What is the difference to gray scale? (Answer in: brightness.png, diffence.txt)

N1	N2	N3
N4	X	N5
N6	N7	N8

Figure 1: Neighborhood of a pixel.

3. Contrast(C) formula: $C(x,y) = \frac{\sum_{i=1}^{8} |N_i(x,y) - X(x,y)|}{8}$
4. Calculate the contrast for each pixel in the gray scale image and display it (don't forget to normalize the result). (Answer in: contrast.png)
5. Calculate the contrast for each color channel in the color image and combine them to a color image (don't forget to normalize the results).
6. Calculate the color differences $Red - max(Green, Blue)$, $Green - max(Red, Blue)$ and $Blue - max(Green, Red)$ combine them to a color image (don't forget to normalize the results). Values smaller than 0 are set to 0.
7. Calculate the contrast for the color differences and combine them to a color image (don't forget to normalize the results). What is the difference to the contrast on the color channels?

Exercise 4: Calculate the saliency map

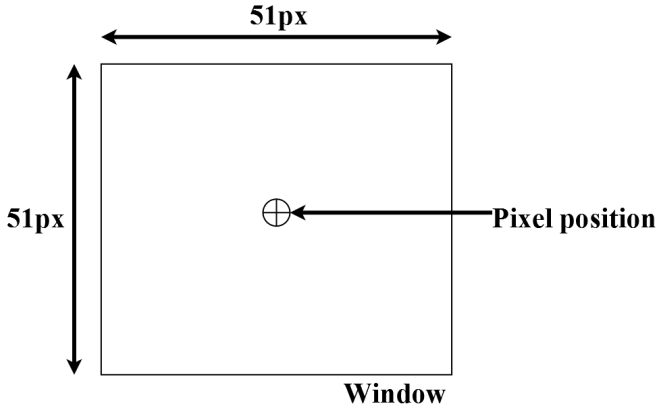

Figure 2: Window for saliency calculation.

1. To calculate the saliency value of a pixel we use a neighborhood window. Each value of the data within this window contributes to the saliency value of this pixel. For the window size use 51 pixels.

2. To take the distance in the window into account we weight each data point by its euclidean distance to the pixel position $Salency(i_1, j_1) = \sum_{i_2=-25}^{25} \sum_{j_2=-25}^{25} \frac{Data(i_1+i_2, j_1+j_2)}{\sqrt{(i_2)^2+(j_2)^2}}$. If the euclidean distance is smaller than 1 it is set to 1.
3. Each saliency map has to be normed such that the sum of it is 1. Why? Whats the formula? (Answer in: saliency-normalization.txt)
4. For visualization normalize the result to (0-1). (Answer in: saliency.png)
5. (Optional) Calculate a saliency map with the data: contrast of gray scale image.
6. (Optional) Calculate a saliency map with the data: brightness.
7. (Optional) Calculate a saliency map with the data: contrast on all color channels.
8. (Optional) Calculate a saliency map with the data: contrast on all color differences.
9. (Optional) Calculate a saliency map with the data: brightness and contrast on all color differences.

Matlab Tips

1. The *im** functions (e.g., *imread* and *imshow*) are your friends when handling images in Matlab.
2. Image channels are accessed through the third index (i.e., *img(:,:,n)* returns the *n*-th channel).
3. The displayed image (using *imshow*) is all black or all white? Check the image values and types. For instance, integer images are interpreted from 0 (black) to 255 (white) while float images are interpreted from 0 (black) to 1 (white). Thus,

   img = ones(100, 'uint8'); imshow(img);

   will display a very dark 100×100 pixels image, and

   img = ones(100, 'single'); imshow(img);

   will display a very bright 100×100 pixels image.