Open your mind. LUT.
Lappeenranta University of Technology
LUT Mathematics and Physics 2015-12-4

BM40A1200 Digital Imaging and Image Preprocessing

Lauri Laaksonen and Lasse Lensu

Exercise 12 solutions: Colours and image processing

1. Response to colour stimulus (1 point): The following was assumed: natural daylight as the ambient illumination, modelled with the Commission Internationale de l'Eclairage (CIE) standard illuminant D65, and an object with the given spectral reflectance.

   (a) The cone responses can be determined as follows:

   $$R_{short} = \int sIC_s, R_{medium} = \int sIC_m, R_{long} = \int sIC_l, \tag{1}$$

   where s is the object reflectance spectrum, I is the illumination spectrum and C is the cone sensitivity.

   (b) The element responses of the red-green-blue (RGB) camera sensor array can be determined as follows:

   $$R_{red} = \int sIC_r, R_{green} = \int sIC_g, R_{blue} = \int sIC_b, \tag{2}$$

   where C is the filter transmittance. where s is the object reflectance spectrum, I is the illumination spectrum and C is the cone sensitivity.

   Additional files: `cones.mat`, `D65.mat`, `object.mat`, `rgb.mat`, `s046.m`.

2. Colour image noise (1 point): Colour image noise:

   (a) Noise can be added to colour images to each of the channels of an RGB image, or just to a specific channel such as the intensity channel in the hue-saturation-intensity (HSI) (syn. hue-saturation-value (HSV)) colour space. Matlab function `imnoise` can be used for the both purposes, but the resulting noise and its visual appearance are very different. The examples shown in Fig. 1 focus on Gaussian noise.

   (b) The signal-to-noise ratio (SNR) can be computed in a similar fashion as with gray-scale images, but all the image channels need to be taken into account. Matlab function `psnr` can be used to compute peak signal-to-noise ratio (PSNR) and SNR. The SNR values for the noisy images are also shown in Fig. 1 and 2.

   (c) Since here the focus was on Gaussian noise, the common approach involving linear filtering in the form of averaging was used. The noisy images filtered with an averaging filter with window size of 3×3 are shown in Fig. 1 and 2.

3. Colour measurement (1 point): Colour measurement:

   (a) Synthetic images are good for studying image processing methods because they can be generated as needed, for example, based on a model of a physical object. Their applicability and especially their representativeness can questioned, however, because a model can be a rough version of the true object. To make it simple, the Matlab function `checkerboard` can be used to generate a synthetic image and noise can be added with `imnoise`, see Fig. 3.

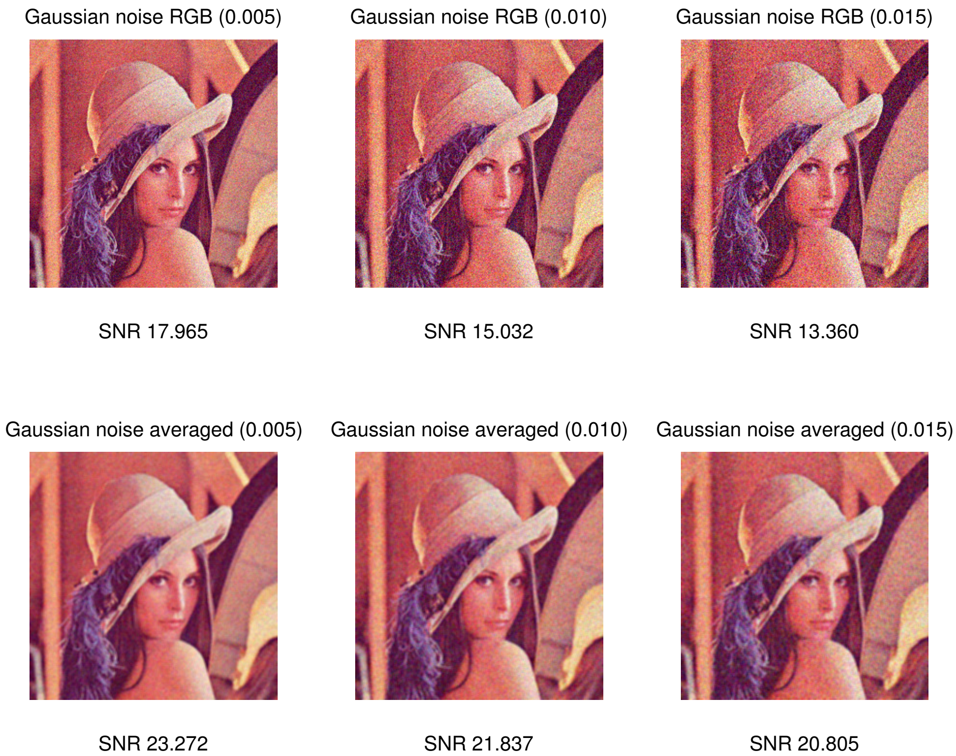

Figure 1: Noisy images with different levels Gaussian noise added to red-green-blue (RGB) colour channels.

Figure 2: Noisy images with different levels Gaussian noise added to the intensity channel of the hue-saturation-intensity (HSI) colour space.

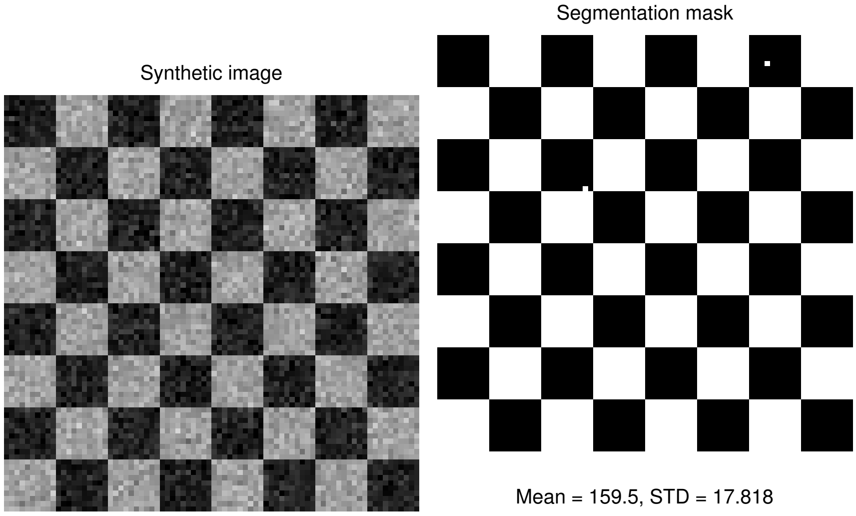

Figure 3: Checkerboard image with noise, and segmented checkerboard image and 1st-order statistics of the pixel values of the bright pixels.

(b) Images captures of physical objects commonly contain noise. Accurately measuring the colour of an object requires (colour) calibration of the measurement setup including the light source, optics and camera, separation of the object pixels from the rest, and giving the measurement result based on a suitable statistic of the pixel values. An example is shown in Fig. 3.

(c) Segmentation by using thresholding is a standard procedure to separate the foreground (object) from the background. The Matlab function `graythresh` implements the Otsu thresholding method and `im2bw` can be used to produce a mask (matrix with logical values) for the more bright part of the image (either the foreground or the background). The success of thresholding depends on the separability of the foreground and background based on the intensity values.

4. Spatial unevenness of image intensity (1 point): Spatial unevenness of image intensity: Due to various reasons, the image intensity values can be unevenly distributed is space.

   (a) It is clear that the illumination is unevenly distributed in the provided image. A suitable model to represent the uneven intensity field is, for example, a radial model centered at the image centre. A radial model estimated based on the pixel values after segmenting the image with Otsu thresholding is shown in Fig. 4.

   (b) The estimated model can be used to correct the illumination field. The result of dividing the image pixel values with the normalised illumination model is shown in Fig. 4.

5. Thresholding and region features (1 point): Thresholding and region features:

   (a) The histogram of the provided image is shown in Fig. 5. Based on the histogram, the first guess for a suitable threshold separating the foreground from the background could be $\frac{60}{255} = 0.235$ because it seems to separate two modes in the histogram.

   (b) The function `graythresh` implements the Otsu method to determine the optimal threshold, and it returns the value 0.365.

   (c) The function `regionprops` can be used to compute various properties of a segmented image. The result of segmentation, the centroids of the segments and the segment with the largest area are shown in Fig. 5.

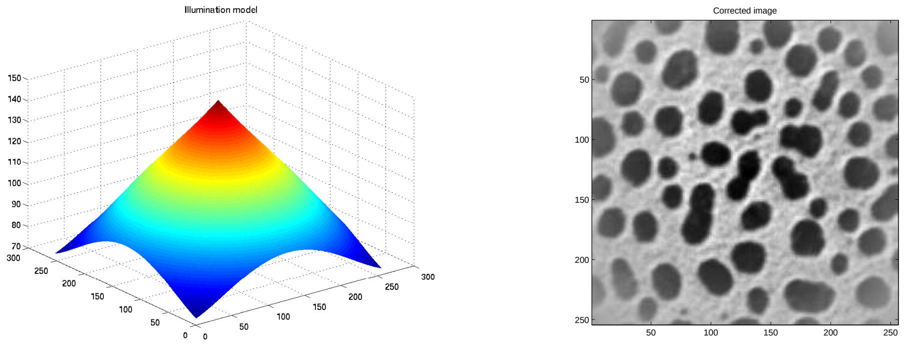

Figure 4: Radial illumination model for the image (left) and the image corrected with the model (right).

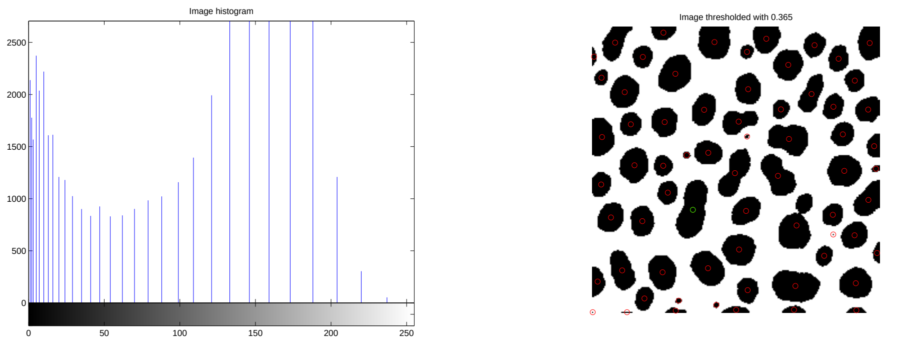

Figure 5: The image histogram (left) and the segmented image (right), the segment centroids (red circles) and the largest segment (green circle).