TNM087 - Image Processing and Analysis

Task 6 - Vignetting

Background:

Read Chapters 2.2.3 and 10.1.3 vignetting in Szeliski: Computer vision

416 10 Computational photography

Figure 10.5 Single image vignetting correction (Zheng, Yu, Kang *et al.* 2008) © 2008 IEEE: (a) original image with strong visible vignetting; (b) vignetting compensation as described by Zheng, Zhou, Georgescu *et al.* (2006); (c–d) vignetting compensation as described by Zheng, Yu, Kang *et al.* (2008).

10.1.3 Vignetting

A common problem with using wide-angle and wide-aperture lenses is that the image tends to darken in the corners (Figure 10.5a). This problem is generally known as *vignetting* and comes in several different forms, including natural, optical, and mechanical vignetting (Section 2.2.3) (Ray 2002). As with radiometric response function calibration, the most accurate way to calibrate vignetting is to use an integrating sphere or a picture of a uniformly colored and illuminated blank wall.

An alternative approach is to stitch a panoramic scene and to assume that the true radiance at each pixel comes from the central portion of each input image. This is easier to do if

Task:

Given two input images compare their vignetting properties. This is done by averaging the gray values in concentric rings in the images. For the images CanonImage.jpg and HolgaImage.jpg add a short interpretation of the result as a comment at the end of your m-file submission. What does the result tell you about the quality of the Canon and the Holga lenses used to capture these images?

Syntax:

```
function [ Profile1, Profile2 ] = Vignette( Im1, Im2, norings )
```

Im1 and Im2 are two images and norings is an optional input parameter describing the number of concentric rings to use (this is the length of the vectors Profile1 and Profile2)

Hints:

You can assume that the images are gray value images but you can also check and if needed you can convert RGB images to a gray value images.

For an image that is not square you can crop it to a square image (if size equal to the minimum size in the x- and y direction). An input image of size 512x256 is thus first cropped to an image of size 256x256

The center of the rings is at the center of the image. The diameter of the biggest (outer) ring is equal to the length of the image. Pixels outside this outer circle are ignored.

Use meshgrid and cart2pol to generate the rings

Extension:

In this case it is assumed that the center of the image (related to the center of the lens) is the midpoint. In a more general case one should consider the location of the midpoint as a free parameter of the method. For homogeneous images (like the example images) one way to do this is to compute the center of mass https://en.wikipedia.org/wiki/Center_of_mass

Note that the calculations here are very similar to the calculations that will be used in the last (Task8) lab. In Task8 rings in the frequency domain will be used.