Measuring intensity of objects

Juan Manuel Castillo Nievas (0518896), Adra Sánchez Ruiz (0520169). Lappeenranta University of Technology. January, 17th 2018

***Abstract*—In this research paper we present an imaging measurement. Our task is to measure the intensity of objects that appear in the images. In order to achieve that, we are going to do some calibration of the data related to intensity.**

***Index Terms*—Intensity, Image Measurement, Calibration, Image Preprocessing.**

I. Introduction

The intensity of an object is the amount of light that the object receives. [1] The amount of light is reflected in the pixel values that the object has. The idea of this research paper is to identify an object on an image and calculate the mean of its intensity. We are going to follow different steps in order to achieve that.

II. Calibration

In order to calibrate the data, we have taken a set of bias images, dark images and flat-field images with the different cameras. This calibration follows a practical procedure that we studied in this course. First, we compute the mean image of the bias images and subtract it from all the rest. Then, we compute the mean of the dark current images and subtract it from the raw image and flat-field images. Then, we sum up the flat-field images and we normalize the result to mean 1. Finally, we divide the raw image by the normalized flat field. The calibration is represented in the *calibration_data.m* file.

A. *Dark images*

For taking dark images [2], we have taken pictures with the same conditions (exposure time, ambient temperature, etc.) without any light, that is, with the sensor cap. With this technique, we obtain only the signal of the electrons that come from the dark current.

B. *Flat-field images*

For taking flat-field images [2], we have taken pictures of a blank surface (it can be the sky in the dusk or dawn or the computer screen displaying a blank image). This makes the pictures be brighter in the center. [3] This images are taken with the same focus, camera orientation and optical setup.

C. *Bias images*

A similar process as dark images [4], but this time we need to use the fastest shutter speed of our camera. The images are taken with the sensor cap. The fastest shutter speed of our camera is 1/4000 s.

III. Identifying The Object

To identify an object in an image, we need to use the thresholding [5]. We need to segment the image in order to identify objects with similar properties. With the thresholding, we create a binary image, so pixels below the threshold are zero and the others are one. Pixel values that have the one value identify the object in the image. That means that first, we have to convert the RGB image to a gray image. This process is made in *detect_object.m* file.

IV. Setup/Environment and Equipment

We have made some pictures with a *Sony Alpha A390*. Specification of this camera is specified in the webpage [6]. We used a white background with objects that have dark colors to make a higher contrast between the background and the objects. The objects are in front of the camera and we have 2 lights, one of them is located above the object permanently and the other is changed according to the information shown on the caption below each image on point VI.

V. Source Code

The idea of this section is to explain the code of our Matlab functions. The main function is called *main.m*. The correct way of using the main function is writing main(name_of_image). This is the function that returns the mean of the intensity of the object. This function reads the image and calls another function called *calibration_data.m*. This function is responsible of calibrating the image according to the dark images, flat field images and bias images that we have explained in section II. After calibrating the data, *detect_object.m* is called. This function calculates the threshold for the image and then convert the image to a binary image, so the object has white pixels and the background has black pixels. Then it calculates the area of the object and put a rectangle bound around the object, so we can see that the object is well identified. The final step of this function is to calculate the mean of the light intensity of the pixels in the gray image that corresponds to the white pixels (pixels that belong to the identified object).

VI. Results

As we have calibrated the image before its intensity is measured, first we are going to present the difference between the original image and the calibrated image.

Figure 1. Light source located in front of the object.
Original image on the left side. Calibrated image on the right side.

Figure 2. Light source located on the left side.
Original image on the left side. Calibrated image on the right side.

Figure 3. Light source located on the right side.
Original image on the left side. Calibrated image on the right side.

Figure 4. Light source located on the right side.
Original image on the left side. Calibrated image on the right side.

Figure 5. Light source located in front of the object.
Original image on the left side. Calibrated image on the right side.

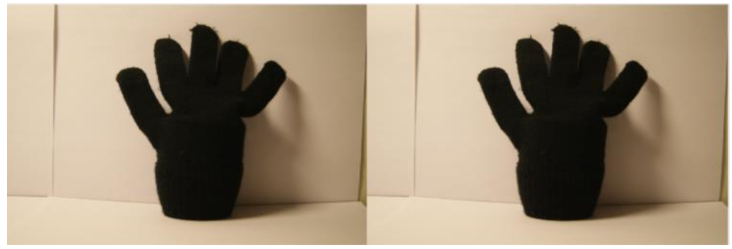

Figure 6. Light source located on the left side.
Original image on the left side. Calibrated image on the right side.

Figures 1 to 6 show the original image and calibrated image of our images, from left to right respectively.

Then, the next step is to calculate a threshold in order to segment the image and identify the object. We have used the Matlab function *graythresh* to calculate the threshold.

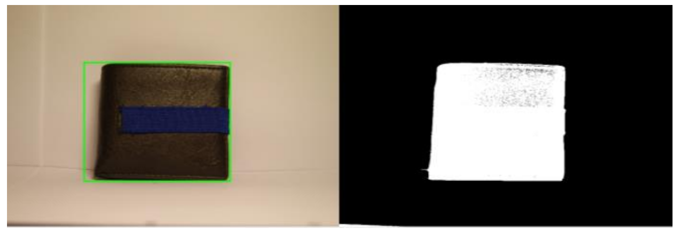

Figure 7. Separating the background from the object.

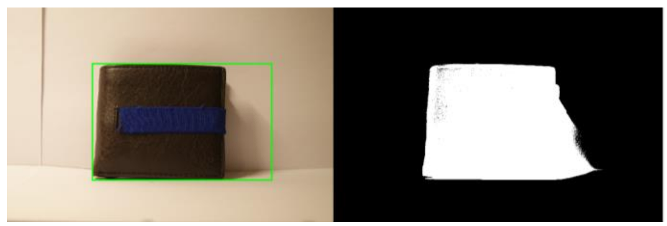

Figure 8. Separating the background from the object.

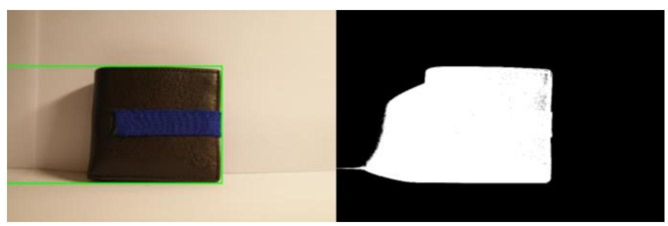

Figure 9. Separating the background from the object.

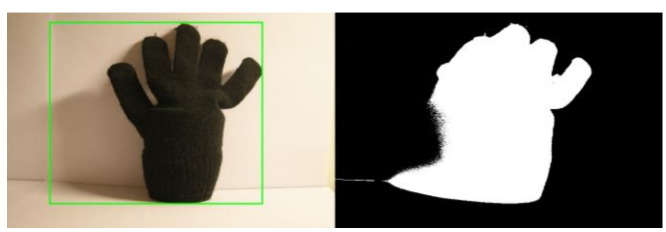

Figure 10. Separating the background from the object.

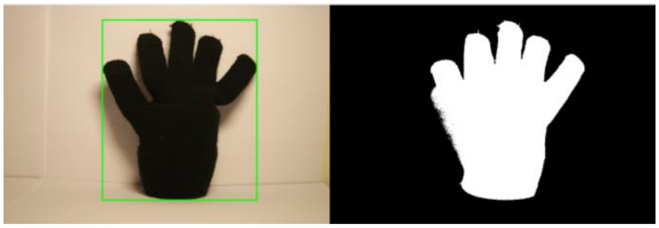

Figure 11. Separating the background from the object.

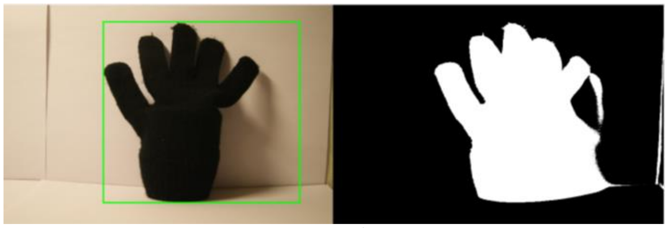

Figure 12. Separating the background from the object.

Figures 7 to 12 show the calibrated image with a rectangular bound around the detected object. On the picture on the right we can also see the binary image. White pixels represent the object and black pixels represent the background. Due to lightning angle changes in some pictures the objects are not correctly detected because of the shadow, so we may have some errors when calculating the intensity. The final step is to compute the mean of the pixels in the original image that are white pixels in the binary image. But our image is represented in RGB, so in order to get the light intensity we have converted the RGB to gray, so the pixels represent the light intensity.

The mean of the light intensity of each image is:

Image number 1 has 46.5812, image number 2 has 54.9400, image number 3 has 57.9191, image number 4 has 58.4580, image number 5 has 28.8583 and lastly image number 6 has 38.7766.

When analyzing the mean intensity changes of the first object (Figure 1 to 3 and 7 to 9) we notice that when the light source is located in front of the object the intensity is lower than when it is placed on the sides. Nevertheless between the last two images there is not a big difference in the intensity mean.

On the other hand the other object of study shows different results, as the shadow plays an important role in the calculation of the intensity. However, we can still see a less intensity value when the object receives the light directly from the front.

References

[1] Image Intensity Understanding, Berthold K. P. Horn, Massachusetts Institute of Technology, August 1975 [online]. Available: https://people.csail.mit.edu/bkph/AIM/AIM-335-NEW-OPT.pdf

[2] Raw Astrophotography Data [online]. Available: http://www.rawastrodata.com/pages/typesofimages.html

[3] Flat-Field Frame Calibration [online]. Available: https://diffractionlimited.com/help/maximdl/Flat-Field_Frame_Calibration.htm

[4] How to shoot Bias frames with a DSLR [online]. Available: http://dslr-astrophotography.com/shoot-bias-frames-dslr/

[5] Lecture 4: Thresholding. Bryan S. Morse, Brigham Young University, 1998-2000 [online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf

[6] Sony Alpha 390 https://www.cnet.com/products/sony-alpha-a390-with-18-55mm-lens/specs/