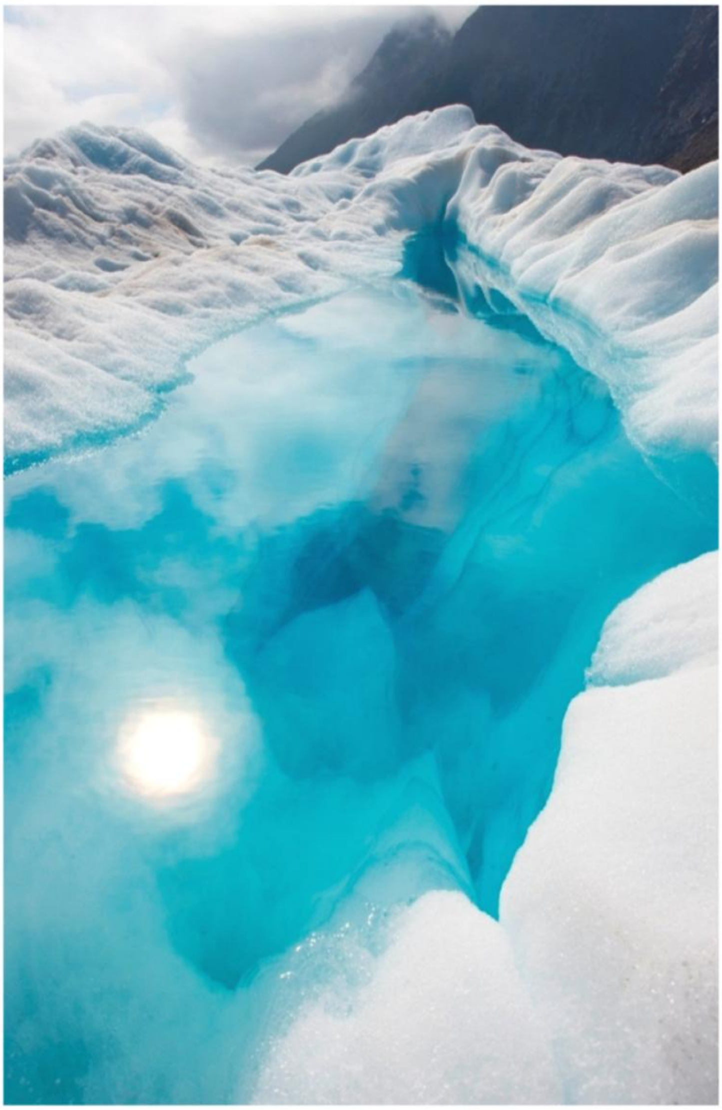

Lens and Lighting

SENSORS AND DIGITALIZATION EXPERIMENT NO 3

DANIEL GONZÁLEZ ADELL
NAYEE MUDDIN KHAN DOUSAI

CONTENT

TABLE OF FIGURES

Objective

The goal of this practical lab is to study the application of different lens with different extension rings and lighting configurations how it will be implemented at various conditions and see how this affects our experiments.

Introduction

The experiment lens and lighting is mainly focused to observe different lens with respect to various change of lightning's by changing the background and angle of capturing image. We will conduct the experiment in three sections as lens, extension rings and lightning's. The observed results are tabulated accordingly with the sample objects like fork, coins and oil filter.

Equipment

For the proper arrangement of the experiments, we will consider:

- PC Computer
- EO Digital Camera
- 12 V Power Supply
- Video cables
- Industrial lenses
- Extension rings
- Lighting devices
- Industrial parts

Lens

The first step is setting an appropriate environment to perform surface inspection of an object like fork with a working distance of 500 mm. We proceed with trial and error method to find the exact lenses which can capture the complete image of fork. The setup is labeled in figure1. Before we start capturing the images of the object, there should be a check of:

- Connect the camera cables
- Start the camera manufacturer software

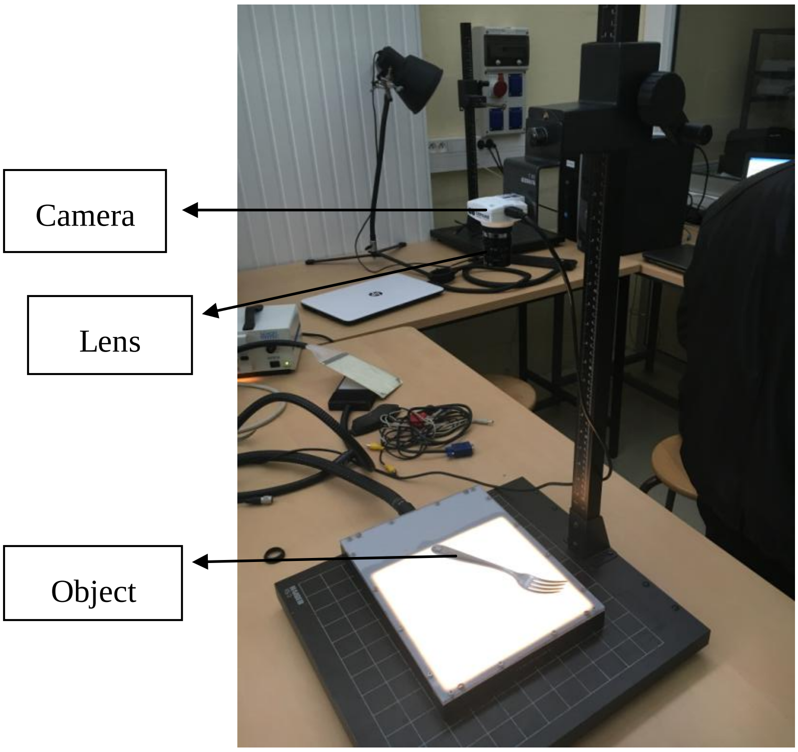

Figure 1 Camera at working distance

Once the setup is properly arranged as per the figure 1, we will proceed to take the pictures of object with the working distance of 500mm and the calibrate lens to capture the complete image of fork. Then we can start grabbing images.

The lens chosen for performing all the tasks in lab is 8 mm lens labeled in figure 2.

Figure 2 Lens used for capturing images

After calibrating the lenses we will proceed to have the surface inspection of the object by taking the pictures using the provided manufacturer software. The picture is labeled as figure 3, which has to compute the spatial resolution.

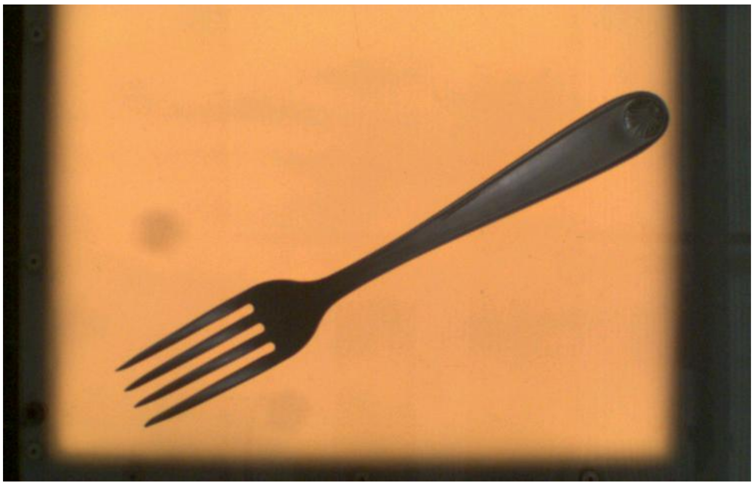

Figure 3 Fork Image with 8mm Lens

$$\text{Spatial resolution} = \frac{\text{Number of pixels along width or height}}{\text{measure of width or height in mm}}\ \text{pixels/mm}$$

Once we have validated with the setup, we can proceed to compute the spatial resolution as follows:

We know that the camera is a 752x480 pixels camera, so we compute:

(We take a look at the image of the fork and we conclude that the size spanned in diagonal is 29 cm).

$$PPI = \frac{diagonal\ resolution}{diagonal\ size\ (photo)} = \frac{\sqrt{752^2 + 480^2}\ px}{29\ cm \cdot 2{,}54\,\frac{in}{cm}} = 78.18\ ppi\ (pixels\ per\ inch)$$

Extension rings

Once we get the results of spatial resolution for the 8mm lens, we have provided with different extension rings to test the image capturing effects. We started to try all the provided rings from 5mm to 40mm. The one which we choose in our experiment is 5mm because of the better focal length to capture the images.

The extension rings are used in all types of photography, to focus closer to a certain scene. They contain zero optical elements and their only purpose is to move the lens further from the plane of pixels. We can combine different extension rings in order to find the best closeness to the element we want to make photo of. In our case, we have preferred to use 5mm extension ring as it would be a good choice with the constraint of 500-mm working distance. The other extension rings would make the scene to make the fork a blur image or an image with greater zoom rate. The images are labeled below for different extension rings.

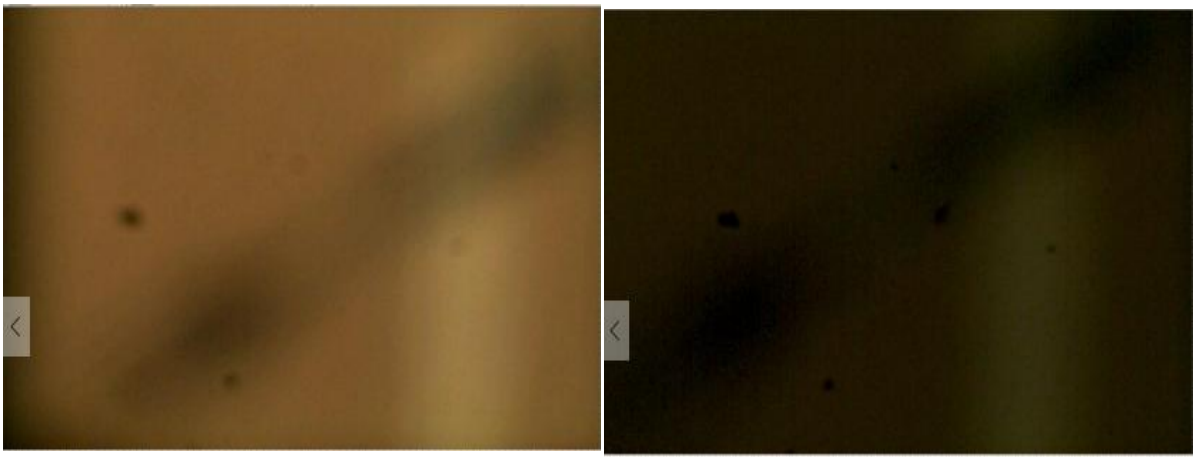

Figure 4 Images obtained with rings of 10 and 20 mm

In conclusion about rings, we could say that basically these rings are used to enabling the movement of the lens along the axis orthogonally to the image plane.

Lighting

In this section, we will study the effects of different lights into a scene and its contributions to a better photography. We will focus this part into silhouettes analysis and defect detection on shiny surfaces.

Silhouette Analysis

The requirement here is to obtain a good view of the contours of the fork.

For this task, we decided to back-illuminate the fork, as we had the proper lighting device to do so. We also realized that the less light we had coming from the environment, the better shape analysis we would be able to do, as the scene wouldn't be "contaminated" by other lights than the one we want to use. We can observe the result of setup in figure 5 and figure 6 represents the shape analysis.

Figure 5Silhouette setup

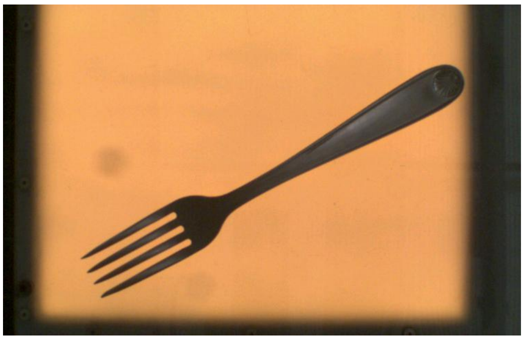

Figure 6 Result of Silhouette image

From the silhouette analysis, we can easily determine the shape of the object, and so we can easily do any segmentation operation, as the contrast is well set by the back-illumination itself.

Defect detection on shiny surface

Our shiny surface will be another fork that the initial setup provides us with. We will now setup our system in order to observe holes and scratches on the surface of the object.

To carry out this observation, our setup must be carefully chosen, as it depends on the angle from which the light comes that we will be able or not to visualize the scratch, such that the light will be deviated from the normal reflection and captured with the lens labeled in figure.

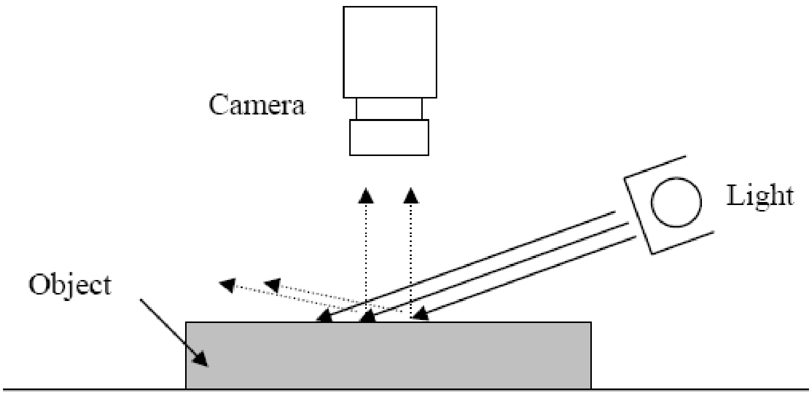

Figure 7 Effect of scratches

For this reason, we set our light source (fiber optics source) at 20 cm high and separated 10 cm from the bottom of the fork, which was the studying focus.

We tried to find the appropriate lens for taking a closer image of the scratches, but after several unsuccessful tries in which we didn't find a good lens out of the set, we decided to bring the structure itself closer to the object, as this would have the same consequences as using a new lens. The workspace was changed to 40 cm now with 8 mm lens without any extension ring.

Figure 8 Scratches on the fork

Basic processing

The goal of the third part of the lab is acquiring images of colored objects with the appropriate light in order to be able to perform a better segmentation of those colored objects once the images have been taken.

The objects that we are given for segmentation are:

- A center of oil filter
- Some coins
- LED's (for lighting up the scene)

First of all, we will do a first approach to the problem by acquiring a sample image of a colored object (a red pen) to see how the previous light used behaves in order to allow us to make segmentation after:

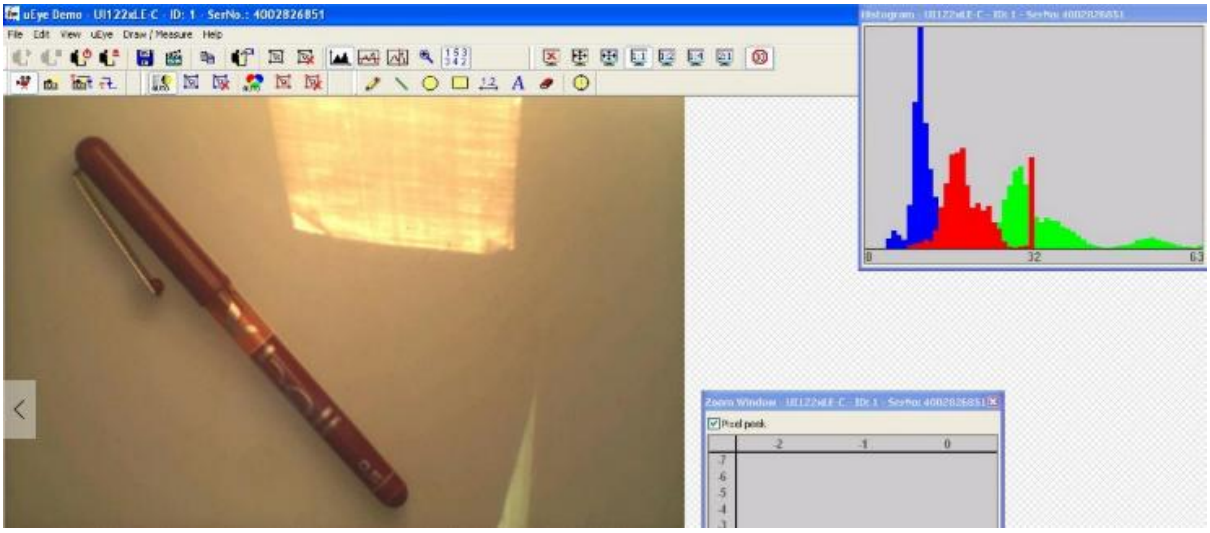

Figure 9 Screenshot of Red pen

As we see, the UEye software allows us to obtain the histogram of the image. We can appreciate that the colors are separated enough to try a histogram-based segmentation approach.

After that first approach, we can focus on our coins and oil filter in order to segment the place by detecting the coins and the filter's center.

Figure 10 Oil filters under blue and red light

Led's were helpful in this task, as the reflectance of the surface of the filter was greater in the center of it. As it is seen in the photo, is easy to recognize the center of the filter more precisely with blue light).

With coins, we were unsuccessful trying to apply led light in order to have better conditions for the segmentation of the picture, so we decided to apply the same retro-illumination we used in the previous experiments. It actually gave excellent results, as it is seen in the photo, for setting the photo ready for segmentation 'a posteriori'.

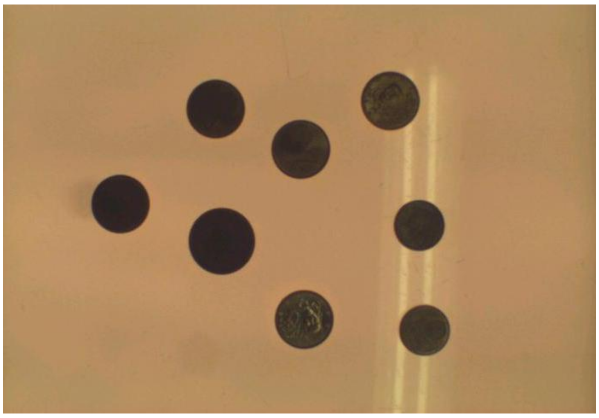

Figure 11 Coins picture

We have used the back-illumination for very-low lights, as after trial and errors we decided this was the optimum one for achieving the post processing.

Conclusion

We have conducted various experiments on different objects by changing lenses and lightnings. The results are tabulated with different images and at the end the coins picture is segmented using python by the black body presence.

SEGMENTATION OF THE COINS PICTURE USING PYTHON

```
%matplotlib inline
%pprint off

from __future__ import print_function, division

import numpy as np
import matplotlib.pyplot as plt

import matplotlib.cm as cm

from skimage import transform     #importing transform to be able to rescale
from skimage import img_as_ubyte
from skimage.color import rgb2gray
```

```
from skimage.io import imread, imshow
from skimage.measure import label, regionprops
from skimage.morphology import binary_closing, binary_opening, disk, binary_erosion, binary_dilation, diamond, octagon, remove_small_objects, remove_small_holes
from skimage.filters import threshold_otsu, threshold_adaptive, threshold_isodata, threshold_yen, threshold_li
from skimage.transform import rescale

from skimage import draw

Path = './images/'
image_name = 'nu_coins.png'
coins_im = imread(Path.__add__(image_name), as_grey=True)
coins_im = img_as_ubyte(coins_im)
coins_im = transform.rescale(coins_im, 1/4)  #reescaling of the original image
plt.title('Grayscale coins')
imshow(coins_im)
```

OUTPUT:

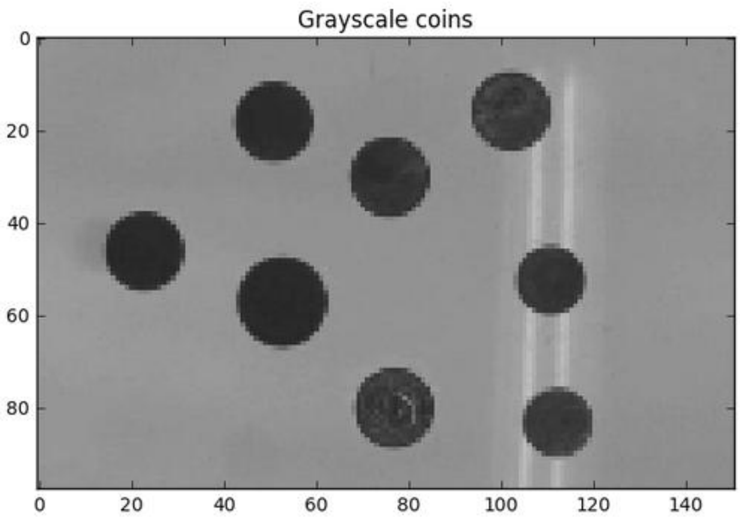

Figure 12 Greyscale Coins picture

```
coins_im = 1- coins_im #inverting colors
local_otsu = threshold_otsu(coins_im, 256) #applying Otsu
global_otsu = coins_im >= local_otsu
 plt.figure()
plt.title('Coins with Otsu applied')
imshow(global_otsu)

selem = disk(2)  #defining a structuring object type disk of radious 2
coins_opening = binary_opening(global_otsu, selem)
plt.figure()
plt.title('Image after one opening')
imshow (coins_opening)

coins_dilated = binary_dilation(coins_opening, selem)
coins_dilated2 = binary_dilation(coins_dilated, selem)
coins_closing = binary_closing(coins_dilated2, selem)

coins_clean = coins_closing
plt.figure()
plt.title('Image finally cleaned')
imshow (coins_clean)
```

OUTPUT:

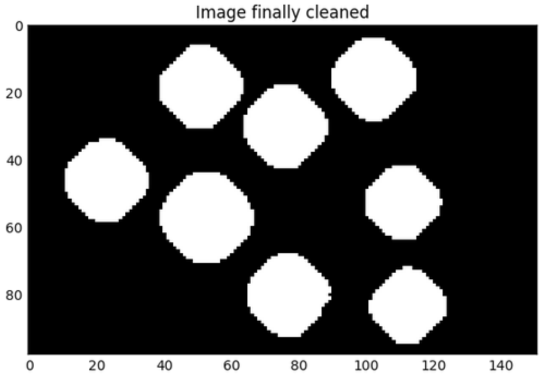

Figure 13 Coins with segmentation

labels, num = label(coins_clean, neighbors=4, background=0, return_num=True, connectivity=True)

plt.title('Segmentation of the image')

imshow(labels)

print ('The number of regions found is :', num)

print(' ')

props = regionprops(labels)

print(len(props))

for i in range(0,len(props)):

    print (props[i].centroid)   #showing the locations of each region found

print (' ')

for i in range(0,len(props)):

    print (props[i].equivalent_diameter/2) #showing the radious of each region found

for i in range (len(props)): #drawing of the cirles out of the regions and their centroids

    circles_coord = draw.circle_perimeter(int(props[i].centroid[0]), int(props[i].centroid[1]), int(props[i].equivalent_diameter/2))

    labels[circles_coord[0], circles_coord[1]]= 255

plt.figure()

plt.title('Recognising circles out of the segmentation')

imshow(labels, cmap= cm.Greys_r)

OUTPUT:

```
The number of regions found is : 8

8
(15.695175438596491, 102.02631578947368)
(18.186252771618626, 50.780487804878049)
(30.0, 75.702407002188181)
(46.052747252747253, 22.701098901098902)
(52.5, 110.58064516129032)
(57.008912655971479, 52.5222816399287)
(79.986666666666665, 76.544444444444451)
(83.264864864864862, 112.19189189189188)

12.047792665040703
11.98155910843366
12.060995729457511
12.034575115625177
10.881694613449236
13.363077719938119
11.968268412042981
10.852403323135505
```

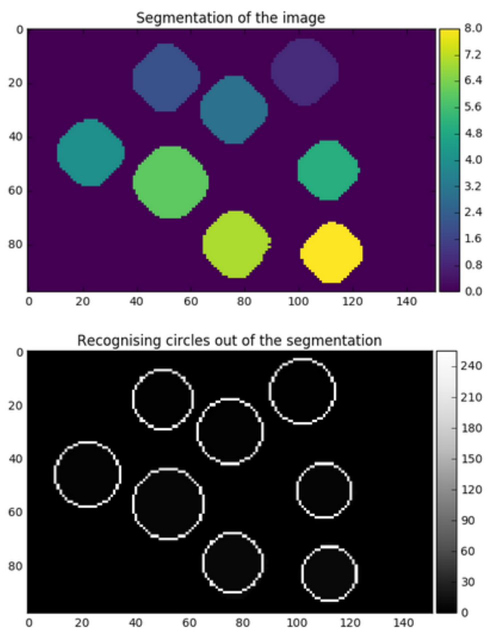

Figure 14 Coins with final segmentation result