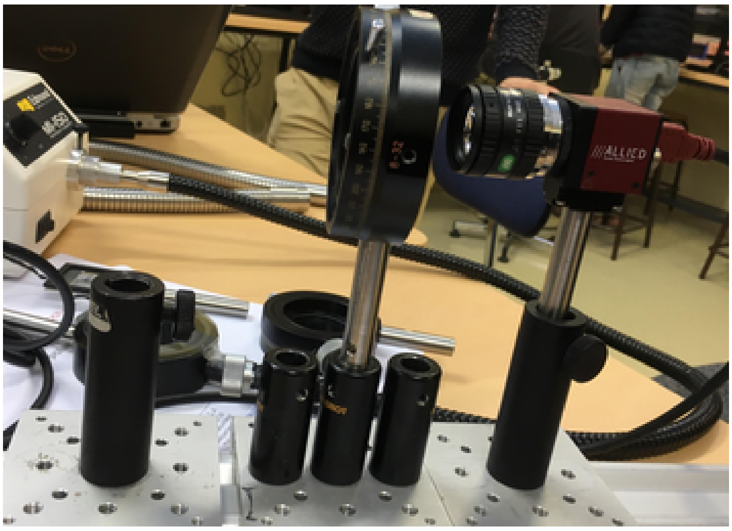
Labs 1 - Polarization Imaging



Submitted by **Jaafer Al-Tuwayyij, Ivan Mikhailov and Selma BOUDISSA**

Contents

List of Figures

Introduction

Advances in sensing technology enable acquisition of a large amount of information from the physical world.
Vision techniques have played a key role in information sensing, but with a limitation that most vision systems can only perceive partial information beyond the visible spectrum.
For instance, one cannot perceive the information carried by a polarized light since human vision systems are not sensitive to polarization. On the other hand, however, some marine and terrestrial animals and insects demonstrate their abilities to sense and utilize polarized lights to navigate, locate, and hunt for prey in their daily activities.
Polarization is a unique characteristic of transverse wave, which is the asymmetry phenomenon of vibration direction and propagation direction. Comparing with conventional image techniques, the polarization imaging technique can detect the polarization information of targets, which will be beneficial to subsequent applications, such as target detection, recognition and identification.
Polarization is a general descriptor of light and contains information about reflecting objects that traditional intensity-based sensors ignore. Difficult computer vision tasks such as image segmentation and object orientation are made tractable with polarization vision techniques. Specularities, occluding contours, and material properties can be readily extracted if the Stokes polarization parameters are available.

Objectives

The goal of this lab is to study and analyze different polarizers on a camera to transform it to a polarization state measurement system,And do some calculations to enhance the polarized images using Wolff's method and the least mean square method, and to understand the effect of different orientations and the effect of a light ring on the output image.

Chapter 1

Simplified polarization imaging

1.1 Wolff's method

We were tasked with implementation pf the Wolff's method based on the images taken by our sensor. Wolff's method is a special case of the Least Mean Square Method. Before taking the images, we needed to manually set the parameters, such as exposure and gain to make the best possible representation of the scene, and obtain 3 images with the pre-determined polarizer orientations 0, 45 and 90 degrees. For the implementation of the Wolff's method in Matlab we used the provided formulas(obtain phi, ro and I), so the calculation and the results were straightforward and can be observed in the attached code.

1.2 Least Mean Square method

Following the Wolff's method, we were tasked to implement Least Mean Square Method (LMS) in Matlab. As a source of data we acquired 8 images with our sensor, each one with a different polarizer orientation (0, 45, 90, 135, 180, 225, 270, 315).

For this task we needed to estimate Stokes vector parameters (s0, s1 and s2) and I, ro and phi, as in the previous task, using LMS

1.2.1 Implementation

For the LMS we used the provided formula, which was implemented in the code: X = $(A^tA)^{-1}A^tY$

The parameters in the formula are stated as followed:

Notations

- N images: $i = 1 \dots N$
- N values for the polarizer angle: $\alpha_1 \dots \alpha_2$
- $M = r \times c$ pixels: $j = 1 \dots M$

For every pixels: $j = 1 \dots M$

$$
\begin{aligned}
P_1^j &= 0.5(s_0^j + s_1^j \cos 2\alpha_1 + s_2^j \sin 2\alpha_1) \\
P_2^j &= 0.5(s_0^j + s_1^j \cos 2\alpha_2 + s_2^j \sin 2\alpha_2) \\
\vdots &= \vdots \\
P_N^j &= 0.5(s_0^j + s_1^j \cos 2\alpha_N + s_2^j \sin 2\alpha_N)
\end{aligned}
$$

EndFor

$$
\underbrace{\begin{bmatrix} P_1^j \\ P_2^j \\ \vdots \\ P_N^j \end{bmatrix}}_{Y} = \frac{1}{2} \underbrace{\begin{bmatrix} 1 & \cos 2\alpha_1 & \sin 2\alpha_1 \\ 1 & \cos 2\alpha_2 & \sin 2\alpha_2 \\ \vdots & \vdots & \vdots \\ 1 & \cos 2\alpha_N & \sin 2\alpha_N \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} s_0^j \\ s_1^j \\ s_2^j \end{bmatrix}}_{X}
$$

The resulting images can be observed in the code.

Chapter 2

Contrast polarization measurement

During this task we were asked to estimate the depolarization properties of the object and remove the specular reflections through taking two different images with two orthogonal directions of a linear polarizer. We also used liquid crystal rotator to programmatically change the orientation of the polarizer. To accomplish this task we installed two polarizers oriented at 0 in front of the sensor. Here we can observe how the intensity of the image decreases due to to the Malus's law. The first polarizer quells the unaligned component of the unpolarized light and outputs polarized light (with half the input's intensity). This polarized output has intensity I0. Of the polarized output from the first polarizer, the second polarizer lets through a fraction $(cos?)^2$ where ? is the angle between the axes of the polarizers. I0 is the intensity of the polarized input to the second polarizer, not the intensity of the unpolarized input to the system of two polarizers. With this proviso, the output intensity is $I0 * (cos?)^2$. This causes the decrease in the overall intensity of the picture.

2.1 Getting started

2.1.1 Polarizers oriented at 0°

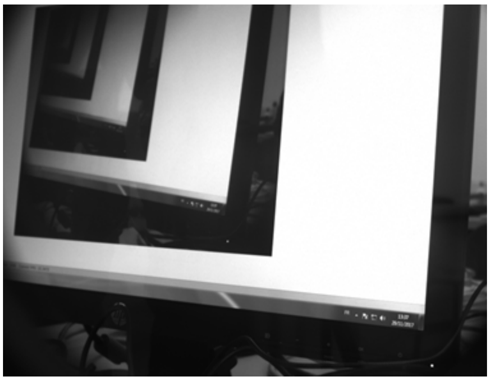


Figure 2.1: Without polarizer

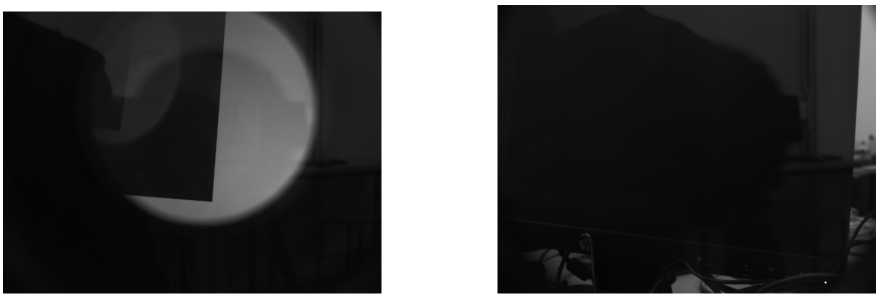
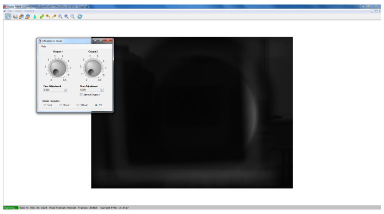


Figure 2.2: Two polarizer oriented at 0°

Analyze: Comparing to the first picture (figure 2.1) we can see by adding two the polarizer oriented at 0° the light decrease continuously this can be explained by the Malus's law. When completely plane polarized light is incident on the analyzer, the intensity I of the light transmitted by the anylizer is directly proportional to the square of the cosine of angle between the transmission axes of the analyzer and the polarizer.

2.1.2 Arcoptix switchable polarization rotator

For this part of the lab we add the arcoptix switchable polarization rotator between the 2 polarizers adn connect the Arcoptix usb LC Driver to the computer and start grabing some images below.



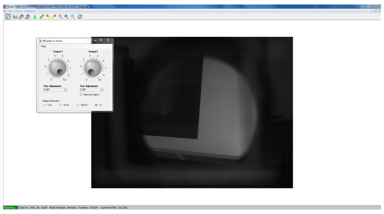


Figure 2.3: Arcoptix usb LC Driver images

Analyze: We can see by playing with the Arcoptix software when the value of the output 1 is equal to 0 (left image) the image produced is dark but when we increase the output 1 to 8.8 (right image) the image is more visible with more light. This can be due to the fact that we increase the intensity.

2.2 Diffuse Specular reflection

To remove reflections we used additional polarized lighting. Below you can observe two pictures with and without usage of the lighting

Figure 2.4: Test without light on the hand

Figure 2.5: Test using the light on the light

Chapter 3

Conclusion

During this work, we learned major basics of the polarization imaging course, including the methods to implement LMS method, which proves itself useful In imaging, and also it's particular example. Wolff's method, using the images obtained with the sensor and a set of polarizer. We mastered the basics of working with linear polarizers, and squinted ourselves with a switchable polarizer rotator based on the twisted nematic liquid crystal. This gave us overall impression, and strengthened our theoretic knowledge basis in the field of polarization imaging.

Appendix

- **Wolff method matlab code**

```
%% Lab 1 - Polarization Imaging
% Authors:Jaafer Al-Tuwayyij, Ivan Mikhailov and Selma BOUDISSA
% Wolff's Method

close all;
clear all;
clc;

% Loading the images
I0 = imread('Wolff_0.bmp');
I45 = imread('Wolff_45.bmp');
I90 = imread('Wolff_90.bmp');

%Compute the polarizer parameters
I=I0+I90;
P=(I-(2*I45))./(I-(2*I90));
phi=atan(double(P))/2;
ro=(sqrt(double(((I-2*I45).^2)+((I-2*I90).^2))))./double(I);

% plotting the figure
figure()

subplot(2,3,1)
imshow(I0);title('I0');

subplot(2,3,2)
imshow(I45);title('I45');

subplot(2,3,3)
imshow(I90);title('I90');

subplot(2,3,4)
imshow(I);title('I0+I90');

subplot(2,3,5)
imshow(phi);title('phi');

subplot(2,3,6)
imshow(ro);title('ro');
```

Lab_1_Wolff_method.m

- **Least Mean Square matlab code**

```
%% Lab 1 - Polarization Imaging
% Authors:Jaafer Al-Tuwayyij, Ivan Mikhailov and Selma BOUDISSA
% Least Mean Square
clc
close all;
clear all;

%load the images
LMS_0 = imread('LMS_0.bmp');
LMS_45 = imread('LMS_45.bmp');
LMS_90 = imread('LMS_90.bmp');
LMS_135 = imread('LMS_135.bmp');
LMS_180 = imread('LMS_180.bmp');
LMS_225 = imread('LMS_225.bmp');
LMS_270 = imread('LMS_270.bmp');
LMS_315 = imread('LMS_315.bmp');

% Calculate the mean
I=[mean(LMS_0(:))
mean(LMS_45(:))
mean(LMS_90(:))
mean(LMS_135(:))
mean(LMS_180(:))
mean(LMS_225(:))
mean(LMS_270(:))
mean(LMS_315(:))];

[rows,columns] = size(LMS_0);
M=rows*columns;
LMS_angles = [0 45 90 135 180  225 270 315];

    A= 0.5*[ 1 cos(2*LMS_angles(1)) sin(2*LMS_angles(1))
             1 cos(2*LMS_angles(2)) sin(2*LMS_angles(2))
             1 cos(2*LMS_angles(3)) sin(2*LMS_angles(3))
             1 cos(2*LMS_angles(4)) sin(2*LMS_angles(4))
             1 cos(2*LMS_angles(5)) sin(2*LMS_angles(5))
             1 cos(2*LMS_angles(6)) sin(2*LMS_angles(6))
             1 cos(2*LMS_angles(7)) sin(2*LMS_angles(7))
             1 cos(2*LMS_angles(8)) sin(2*LMS_angles(8))
        ];
  At=A.';

  s=((At*A).^(-1))*At*I;


  I=s(1);
  ro=sqrt(double((s(1).^2)+(s(2).^2)))./s(1);
  phi=atan(s(2)/s(1))/2;
```

Lab_1_Least_Mean_Square.m

References

- **Malus's law** `http://www.physicshandbook.com/laws/maluslaw.htm`