

PROGRAMMABLE TIME DIVISION MULTIPLE POLARIZATION GRAPHIC PROCESSING SYSTEM

*NSF Summer Undergraduate Fellowship in Sensor Technologies (SUNFEST)
Carlos Biauou (Electrical Engineering)- Prince George's Community College
Advisors: Dr. Jan Van der Spiegel & Dr. Nader Engheta
Mentor: Dr. Milin Zhang*

ABSTRACT

Polarization is an intrinsic and fundamental property of light, as are intensity and wavelength. It is determined by the direction of the electric field found in light. The orientation of the electric field changes as it interacts with objects, including particles in the air. Image quality can be improved by introducing the “degree of polarization” into standard imaging processing. In this project, in order to derive the degree of polarization from standard intensity image, a programmable time-division multiple polarization graphic processing system is proposed. Frames are captured at different polarization angles by rotating a polarizer placed in front of a camera. Post data processing is performed in real-time to extract polarization information. A precise but bulky optical system has been proposed in literature to realize the detection of polarization. In order to improve the flexibility of such a polarization detection application, we introduce electric mechanical devices into the system for the control of the polarizer. The system is tested to showcase its versatility and possible application in material detection, object recognition and fingerprint identification.

TABLE OF CONTENTS

1. INTRODUCTION
2. BACKGROUND: POLARIZATION IMAGING SYSTEM
 - 2.1 “Degree-of-polarization” Imaging
 - 2.2 Servomotors and Field Programmable Gate Arrays (FPGA)
3. EXPERIMENTAL SETUP
 - 3.1 Hardware
 - 3.2 Software
4. EXPERIMENTAL RESULTS
5. DISCUSSION
6. RECOMMENDATIONS
7. CONCLUSION
8. ACKNOWLEDGEMENTS
9. REFERENCES
10. APPENDIX 1
11. APPENDIX 2

1. INTRODUCTION

Each vibrating atom at the source of any light beam creates a wave (the electric field) that is oriented in a particular direction, which is known as the direction of polarization. Light is polarized if the electric field is vibrating in the same direction at any time [1]. Sometimes, the source light is not polarized, but once it is interacting with the surface of an object or small particles, the reflection becomes partially polarized. An ordinary beam of light is made of electromagnetic waves emitted by vibrating atoms at the source. Each wave corresponds to a particular orientation of the electric field vector. The direction of polarization is therefore defined as the direction in which the electric field is vibrating. Light sources, such as light coming from the sun, are unpolarized, because the electric field is vibrating in many different directions. Light is polarized if at any given time, the electric field is always oscillating in the same direction. When light interacts with a surface and is scattered, the specular component of the reflected light is partially polarized, according to Fresnel's analysis and equations [2]. Also, according to the theory of Rayleigh [2], the atmosphere's scattering of light is also a source of polarization due to air particles that are electric dipoles. Since oscillating dipoles do not radiate in their direction of oscillation, a polarization-sensitive observer will see the dome of the sky to exhibit a polarization pattern that depends on the location of the sun [3]. Polarization-sensitive imaging can therefore be employed to reveal details that color and intensity images cannot represent.

Labhart [4] has demonstrated that bees and crickets are able to detect polarized light through their visual system. Wehner further has proven that insects, such as bees and ants are not only able to detect polarized light, but they actually are able to navigate by using the polarization information [5].

Inspired by this biological phenomenon, Rowe *et al.* successfully identified a target in a scattering medium by the means of polarization-difference imaging [6]. Another research group was able to enhance images to reveal details that wouldn't normally be seen by applying the principle of polarization of light [3].

Polarization is an intrinsic and fundamental property of light, as are intensity and wavelength. To record important data for polarization imaging, polarization difference imaging (PDI) is an effective method to use. It consists of recording the intensity of two images at two orthogonal directions of polarization and calculating the difference of intensity pixel-by-pixel. Tyo *et al.* conclusively proved that the resultant image quality in a scattering medium is better than an image recorded by a polarization-blind camera in the same conditions [7]. Tyo *et al.* mechanically changed the position of their polarizer to record the data, which could be optimized by implementing a control system to automatically change the position of the polarizer and compute the data.

Considering the importance of polarization of light, building a detection of polarized light system will be useful for material identification, fingerprint detection in forensic science, among other applications. Therefore, we propose to build a compact and portable polarization detection system.

2. BACKGROUND: POLARIZATION IMAGING SYSTEM

2.1 “Degree-of-polarization” Imaging

Considering the images in grayscale, the captured frame represents the level of intensity of each pixel. The image is pretty accurate, but when there is a very intense light source in the environment of the scene to be captured, it tends to hide details that are near that light source. The result is a white frame at the position where the light is coming from. Ordinary cameras do not take into account the phase angle of a polarized wave of light, necessary to find the degree of polarization of light. However, the intensity I recorded by a camera equipped with a polarizer is given by [3] as follows:

$$I(\phi) = I_u[1 + p \cos(2(\theta - \phi))] \quad (1)$$

where I_u is half intensity at each pixel, θ is the orientation angle of the major axis of the polarization ellipse, ϕ is the angle of linear polarization with respect to an arbitrary axis, and p is defined as the degree of linear polarization. Using p , it is possible to extract images that reflect the intensity of the polarization of light. The formula above suggests that the intensity information should be captured from a polarizer oriented at three different and independent angles, such as 0° , 45° and 90° . Thus, we obtain the following expressions for I_u , θ , and p :

$$I_u = \frac{I_0 + I_{90}}{2}$$

$$I_u = (I_0 + I_{90})/2$$

$$p = 2 \frac{\sqrt{(I_{45} - I_u)^2 + (I_{90} - I_u)^2}}{I_0 + I_{90}}$$

$$\theta = \frac{1}{2} \times \arctan \left[\frac{I_{45} - I_u}{I_{90} - I_u} \right]$$

$$p = 2 \frac{\sqrt{(I_{45} - I_u)^2 + (I_{90} - I_u)^2}}{I_0 + I_{90}}$$

$$\theta = \frac{1}{2} \arctan \left[\frac{I_{45} - I_u}{I_{90} - I_u} \right]$$

The indices 0, 45, and 90 indicate the orientation of the polarizer (ϕ).

2.2 Servomotors and Field Programmable Gate Arrays (FPGAs)

To control the angle of a polarizer, a servomotor is implemented and controlled by an FPGA. A servomotor uses negative feedback to monitor its movement. Thus, after each of its rotation the motor receives a feedback to indicate whether it has reached the desired position. Once the desired position is reached, a positive feedback signal is sent to the motor, which will stop rotating and maintain that position due to a magnetic field created inside of the rotor. Servomotors are particularly useful in applications such as positioning in which precision is vital. In this project, it will be used to position the polarizer at three different polarization angles using feedback sent from a FPGA.

A Field Programmable Gate Array, or FPGA, is an integrated circuit that can be programmed to suit the user's application. It is capable of controlling other devices as well as performing computations, which is highly desirable in this project, because we will be able to control the servomotor and potentially process the data recorded from the polarization-sensitive camera at a single location. These capabilities are highly desirable for building a portable system.

3. EXPERIMENTAL SETUP

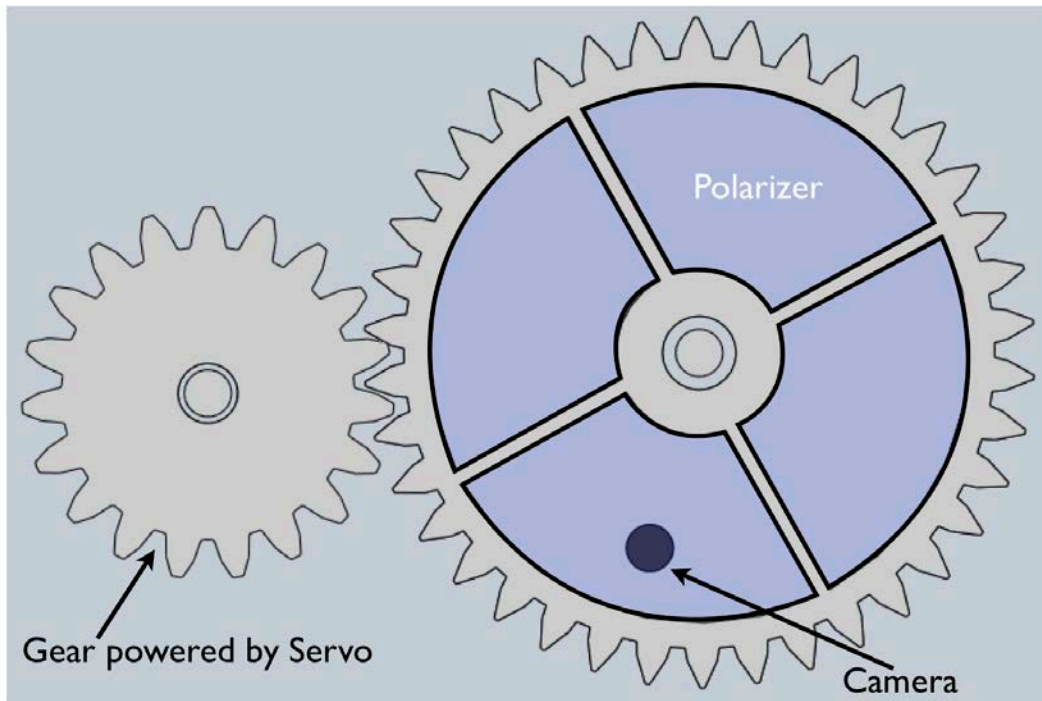
The system consisted of assembling hardware development, software development, and interface design.

3.1 Hardware

The components of the experimental setup are as follows:

- Two spur gears of respective diameters 12 cm and another of 6 cm;
- Polarization sheet;
- Futaba S3003 Servo motor;
- Opal Kelly XEM3010 Xilinx Spartan-3 FPGA;
- Apple iSight camera, 640x480 resolution.

The polarization sheet, which filters specific angles of polarization, is attached to the larger spur gear. It allows the camera to capture frames through the polarization sheet at various angles. The smaller spur gear is directly attached to the servomotor. The motor and the spur gear rotate at the same angular speed; thus, a same angular position is achieved. The teeth of the two gears mesh. The 12cm-diameter spur gear has 38 teeth, whereas the 6cm-diameter spur gear has 19 teeth. The gear ratio from the motor to the polarizing sheet is 2:1. The gears are arranged as shown in Figure 1 (a).



(a)



(b)

Figure 1: (a) Schematic of the front view of the experimental setup showing the arrangements of the gears. (b) Picture of the system implemented.

The servomotor is controlled by pulse width modulation (PWM). Further details will be given in the software section. The FPGA itself is powered by a USB cable supplying 3.3V, which is enough to power the servomotor since it would function for any voltage in the range 3V to 6V.

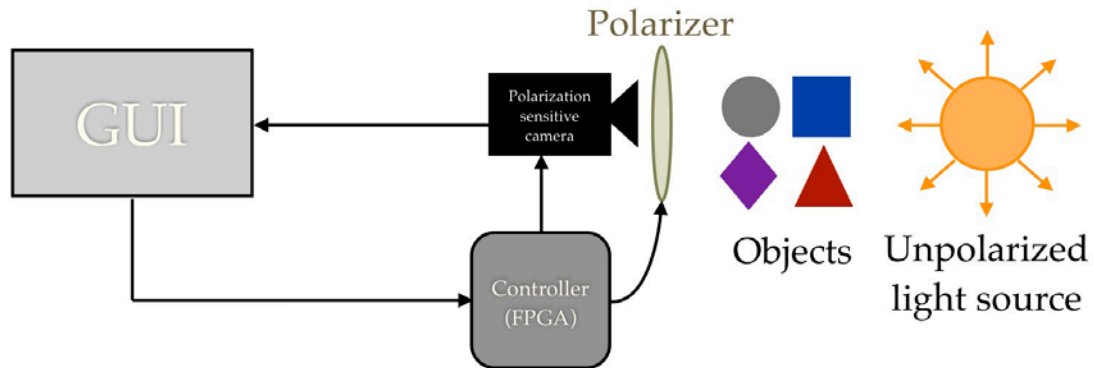


Figure 2: Schematic of the overview of the experimental setup.

3.2 Software

Programming the FPGA to control the servomotor is vital. Indeed, the operation of the servo depends on the duty cycle of the incoming signal voltage, also known as pulse width modulation (PWM). The servo must receive a pulse every 20ms of width varying between $400\mu\text{s}$ and $2400\mu\text{s}$. A VHDL code was implemented to control the duty cycle of the servomotor and is available in Appendix 1.

Processing the frames is important to represent the degree of polarization (See 2.1). The OpenCV (**Open Source Computer Vision**) [8] library is used. It is compatible with many popular programming languages (C, C++, Python...) widely used for computer vision and image processing. Using a C++ compiler, the algorithm described in 2.4 was implemented to obtain polarization sensitive images. The resulting code is available in Appendix 2.

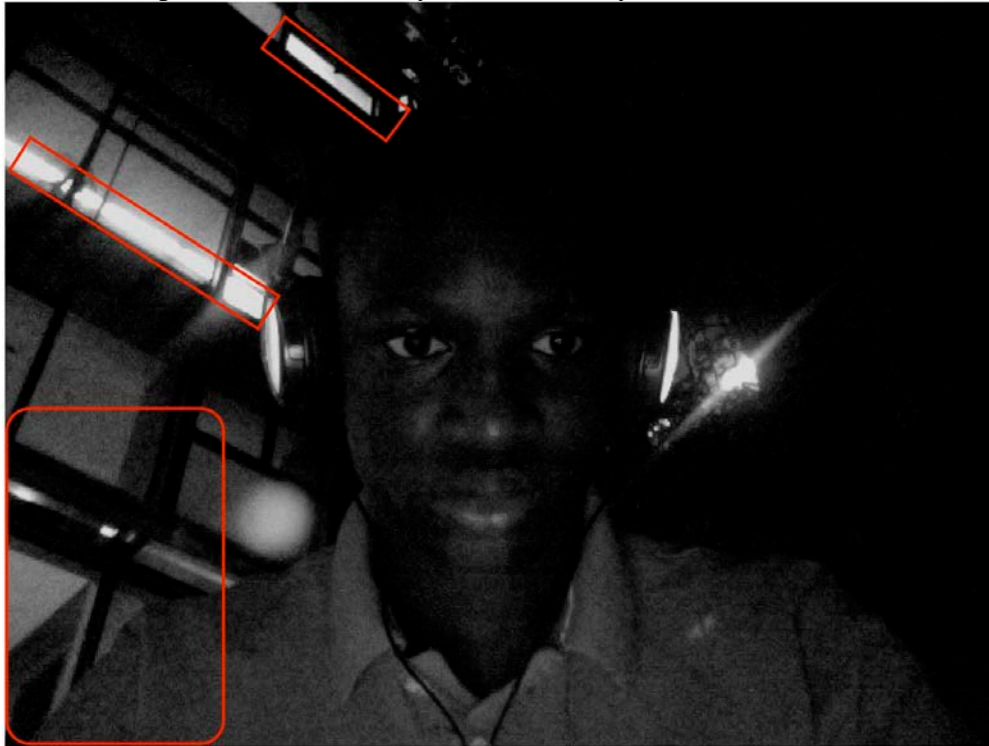
4. EXPERIMENTAL RESULTS

Testing of the system follows these steps:

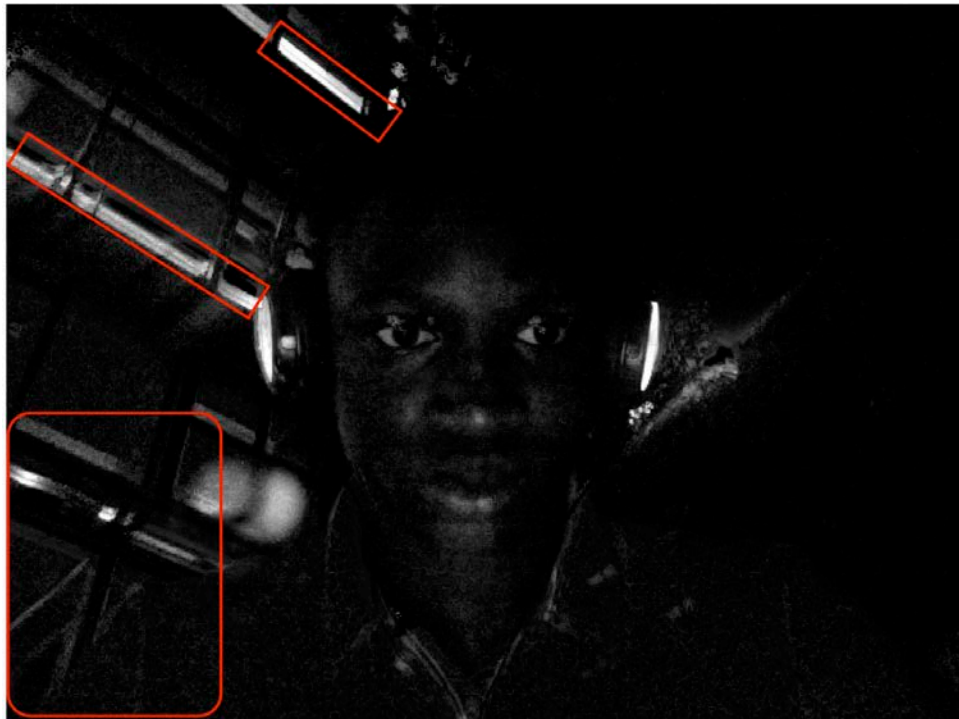
- 1) Set up the polarizer system in front of the camera
- 2) Run C++ code for capturing frames and process “degree of polarization”
- 3) Run VHDL code on the FPGA to control angle of polarization
- 4) Capture grayscale frame at 0°
- 5) Rotate polarizer at 45° with FPGA
- 6) Capture grayscale frame at 45°
- 7) Rotate polarizer at 90° with FPGA
- 8) Capture grayscale frame at 90°

The C++ code using the OpenCV library will consequently output an image reflecting information from the degree of polarization.

The first example is an outdoor scene captured at night to demonstrate that the degree of polarization could reveal details previously invisible. The second example is a demonstration for possible use of the system to identify materials.

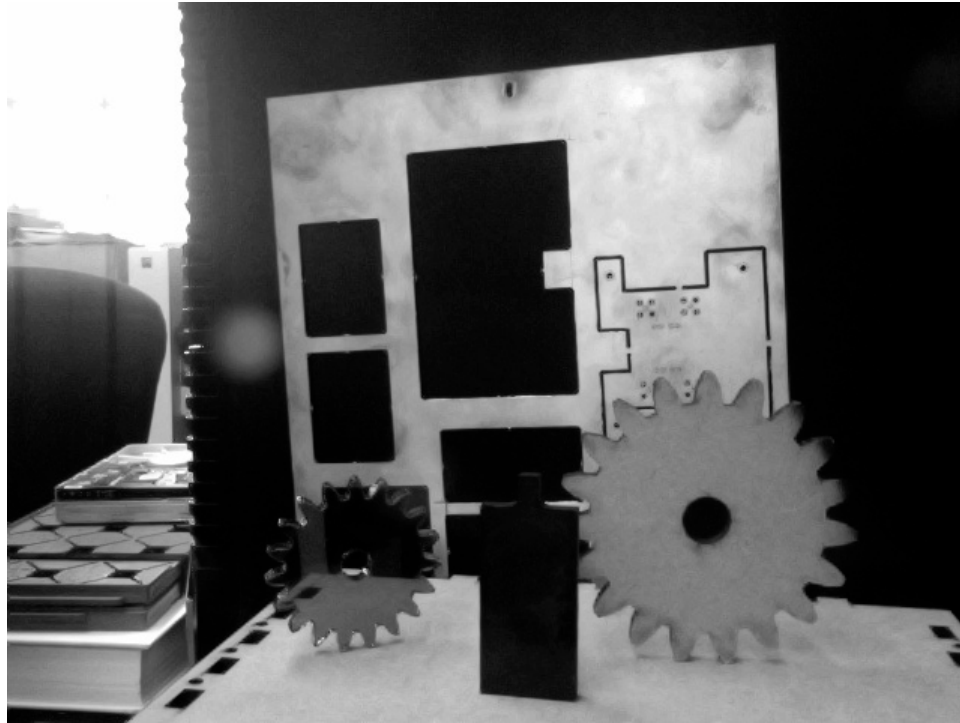


(a)

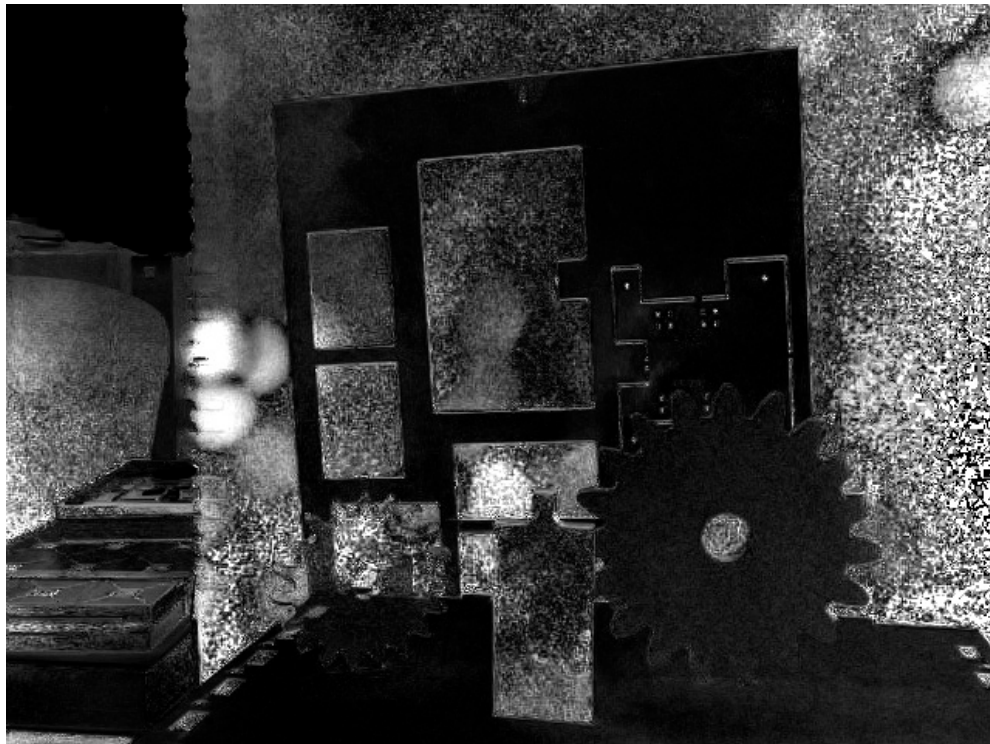


(b)

Figure 3: (a) Grayscale image of an outdoor scene with a building with glass windows. (b) Frame of the same scene processed with the degree of polarization information. In the grayscale image, wherever there's a light source, most of the surrounding details are covered in white. The degree of polarization frame however reveals, though slightly, lines that weren't previously perceptible (see red boxes).



(a)



(b)

Figure 4: (a) Grayscale image of objects made out of various materials (b) Resulting frame after processing with the “degree of polarization” is extracted. Notice that the appearance of each object from (a) is different in (b). The small gear is made out of transparent acrylic, the second gear is wood, the rectangular shape made out of plastic and the shape in the background is made out of copper. The grayscale image makes it impossible to identify the material that the different objects are made of. However, the “degree of polarization” image outputs different p-values for each of the objects made out of different materials. Thus, each of them has a specific appearance. Further work must be done to identify an average p-value for each material and conclusively prove that degree of polarization

5. DISCUSSION

The experiments illustrate various applications of a programmable time division multiple polarization graphic processing system. Concerning image enhancement, it is possible to reveal details that weren't previously visible. However, the improvements are limited due to the limitation of the quality of the polarizer as well as the image sensor. The resolution of the image sensor is only 640x480, which doesn't allow capturing images of high quality, and the dynamic range of the image sensor is not very high as well. In addition, the uncontrollable of the illumination enhances the complexity of post data processing. Furthermore, the accuracy of the rotation of the polarizer effects on the quality of the post-data processing results. There is no question of how powerful the degree of polarization algorithm is. An improvement of performance of the proposed system will increase the polarization image quality.

The second application that the experiments were presenting was the ability to identify materials. This application is quite promising because the results obtained were more conclusive. All the objects in the second example were made of different materials. A "degree of polarization" image made it possible to distinguish the target objects from one another. However, many questions remain to be answered before this technique could find practical application. First, is the p-value (degree of polarization) unique for every material? Second, does the color of the object affect the resulting p-value? Third, does the shape of the object affect the p-value? Finally, are there other elements that could affect the p-value and that are unknown thus far? These are open questions that need to and can be answered. The uniqueness of the p-value for different materials is particularly important because it would mean that if we extract the p-value from an unknown object and compare it to the average p-value of a known one, we could have in our possession a technology that is very useful for fire fighters for example because they can identify the materials burning in a rescue situation and wear the gear to protect themselves and save lives.

6. RECOMMENDATIONS

Future works should be focused on answering the questions that were raised in the discussion. Other routes to explore should include but not limit themselves to:

- Using higher resolution cameras
- Applying the algorithm with other image enhancement techniques
- Applying the algorithm on color images as well
- Demonstrating other applications such as fingerprint detection
- Testing in different environments (foggy, underwater...)

7. CONCLUSION

Polarization sensitive imaging is inspired by biological observations. Previous studies have explored means to exploit polarization, an intrinsic property of light, to capture frames in various environments. This project aimed to build a compact portable, programmable time division multiple polarization graphic processing system. The system was built and tested. Possible applications have been realized using the proposed device. The results obtained were promising, yet could be improved. Recommendations were offered to guide future works in possible directions of interest such as determining the uniqueness or not of the degree of polarization for different materials.

8. ACKNOWLEDGEMENT

I would like to thank Dr. Milin Zhang, Prof. Jan Van der Spiegel, and Prof. Nader Engheta for their guidance as well as Gu Hao, Bo Yuan, Matt Zhu, and Xilin Liu for their technical support. Thanks also go the Summer Undergraduate Fellowship in Sensor Technologies (SUNFEST) at the University of Pennsylvania and the National Science Foundation (NSF) for the research opportunity and the financial support.

9. REFERENCES

- [1] R. A. Serway and J. W. Jewett, *Physics for Scientists and Engineers with Modern Physics*, Eighth Ed. Brooks/Cole, pp. 1228, 2010.
- [2] E. Hecht, *Optics*, Addison Wesley Longman, Inc., Reading, MA, USA 1998.
- [3] S.S. Lin, K. M. Yemelyanov, E. N. Pugh, and N. Engheta, "Separation and contrast enhancement of overlapping cast shadow components using polarization." *Optics express*, vol. 14, no. 16, pp. 7099-108, Aug. 2006.
- [4] T. Labhart, "Polarization-opponent interneurons in the insect visual system." *Nature*, vol. 331, pp. 435-437, 1988.
- [5] R. Wehner, "Neurobiology of polarization vision." *Trends in neurosciences*, vol. 12, no. 9, pp. 353-9, Sep. 1989.
- [6] M. P. Rowe, E. N. Pugh, J. S. Tyo, and N. Engheta, "Polarization-difference imaging: a biologically inspired technique for observation through scattering media." *Optics letters*, vol. 20, no. 6, pp. 608-10, Mar. 1995.
- [7] J. S. Tyo, M. P. Rowe, E. N. Pugh, and N. Engheta, "Target detection in optically scattering media by polarization-difference imaging." *Applied optics*, vol. 35, no. 11, pp. 1855-70, Apr. 1996.
- [8] G. Bradski, "OpenCVWiki." Internet: opencv.willowgarage.com/wiki/, May 8, 2012 [Aug. 1, 2012]

APPENDIX 1: VHDL Code for polarization angle control

```
-----
-- Company: Upenn
-- Engineer: Carlos Biauou (summer intern)// biauoucar@gmail.com--
-- Create Date: July 2012
-- Module Name: control_motor
-- Target Devices: Opal Kelly XEM3010
-- Description: This module is used to control the motor rotation angle
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity control_motor is
port (
    clk : in std_logic;
    rst : in std_logic; -- resets and initializes the motor
    button: in std_logic_vector (1 downto 0);
    duty_cycle: out std_logic
);
end control_motor;

architecture Behavioral of control_motor is
    signal count : integer :=1;
    signal pulse_width: integer :=1;
    signal sys_clk: std_logic :='0';
begin
    process (clk)
        begin
            if (button="11") then
                pulse_width<= 19919; --- 0 degree position
            elsif (button="10") then
                pulse_width<= 64738; --- 90 degree for motor,
45 degree for polarization angle
            elsif (button="01") then
                pulse_width<= 119520; --- 180 degree for motor,
90 degree for polarization angle
            end if;
            if(clk'event and clk='1') then
                count <= count + 1;
                if (count=995970) then --- 20ms period
                    count<=1;
                end if;
                if (count>=pulse_width) then
                    sys_clk<='0';
                else
                    sys_clk<='1';
                end if;
            end if;
        end process;
        duty_cycle<=sys_clk;
    end Behavioral;
```

APPENDIX 2: C++ code for extracting "degree of polarization" image

```
#include <QtGui/QApplication>
#include <opencv/cv.h>
#include <opencv/highgui.h>
#include <opencv/cxcore.h>
#include <math.h>
#include "mainwindow.h"
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    CvCapture *capture = 0;
    IplImage* frame = 0;
    IplImage* grayframe = 0;
    IplImage* grayframe_0 = 0;
    IplImage* grayframe_45 = 0;
    IplImage* grayframe_90 = 0;
    int key = 0;
    /* initialize camera */
    capture = cvCaptureFromCAM( 0 );
    cvSetCaptureProperty(capture,CV_CAP_PROP_FRAME_WIDTH,640);
    cvSetCaptureProperty(capture,CV_CAP_PROP_FRAME_HEIGHT,480);
    /* always check */
    if ( !capture ) {
        fprintf( stderr, "Cannot open initialize webcam!\n" );
        return 1;
    }
    /* create a window for the video */
    cvNamedWindow("Live Webcam Feed", CV_WINDOW_KEEPRATIO );
    /* create a loop to capture x images before quitting */
    for (int x=0; x<3; x++) {
        while( key != 'c' ) {
            /* get a frame */
            frame = cvQueryFrame( capture );
            //cvSaveImage("frame.jpg", frame);
            //grayframe=cvCloneImage(cvLoadImage("frame.jpg",0));
            /* always check */
            if( !frame ) break;
            /* display current frame */
            cvShowImage( "Live Webcam Feed", frame );
            //cvShowImage( "Gray Live Feed", grayframe);
            /* capture if user press 'c' */
            key = cvWaitKey(1);
        }
        /* exit if user press 'q' and save image*/
        while( key != 'q'){
            key = cvWaitKey(1);
            string s1 =
"/Users/Carlos/Documents/SUNFEST/Projects_with_OPENCV/Webcam_Feed_Image
s/frame_.jpg";
```

```

        string s2 =
"/Users/Carlos/Documents/SUNFEST/Projects_with_OPENCV/Webcam_Feed_Image
s/frame_.jpg";
        string s3 = "gray_";
        string s = static_cast<ostream*>(&(ostream()<<
x ))->str();
        s1.insert(78, s);
        s2.insert(78, s);
        s2.insert(78, s3);
        const char* s4 = s1.c_str();
        const char* s5 = s2.c_str();
        cvSaveImage(s4, frame);
        grayframe=cvCloneImage(cvLoadImage(s4,0));
        cvSaveImage(s5, grayframe);
    }
}
/*process image for polarization sensitive information */
grayframe_0 =
cvCloneImage(cvLoadImage("/Users/Carlos/Documents/SUNFEST/Projects_with
_OPENCV/Webcam_Feed_Images/frame_gray_0.jpg",-1));
grayframe_45=
cvCloneImage(cvLoadImage("/Users/Carlos/Documents/SUNFEST/Projects_with
_OPENCV/Webcam_Feed_Images/frame_gray_1.jpg",-1));
grayframe_90=
cvCloneImage(cvLoadImage("/Users/Carlos/Documents/SUNFEST/Projects_with
_OPENCV/Webcam_Feed_Images/frame_gray_2.jpg",-1));
for (int i=0; i<640; i++)
{
    for (int j=0; j<480; j++)
    {
        CvScalar I_0,I_45,I_90;
        double p,I;
        I_0=cvGet2D(grayframe_0,j,i);
        I_45=cvGet2D(grayframe_45,j,i);
        I_90=cvGet2D(grayframe_90,j,i);
        I=(I_0.val[0]+I_90.val[0])/2;
        p=(2*sqrt(pow(I_45.val[0]-I,2)+pow(I_90.val[0]-
I,2)))/(I_0.val[0]+I_90.val[0]);
        //printf("Intensity p=%f\n",p);
        //printf("Intensity I_0=%f\n",I_0.val[0]);
        I_0.val[0]= 270 * p; //Should multiply the degree of
polarization by the original intensity?
        cvSet2D(grayframe_0,j,i,I_0);
    }
}
/*Saving processed frame*/

cvSaveImage("/Users/Carlos/Documents/SUNFEST/Projects_with_OPENCV/Webca
m_Feed_Images/polarized_frame.jpg", grayframe_0);
/* free memory */
cvDestroyWindow( "Live Webcam Feed" );
//cvDestroyWindow( "Gray Live Feed" );
cvReleaseCapture( &capture );
return 0;

return a.exec();
}

```