Dual-Angle Imaging Camera for Determination of Dielectric Thin Film Thickness
Design Description Document

Yang Deng (Project Coordinator)
Diana Magana (Document)
Zheng Tan (Customer Liaison)

Customer/Advisor: Professor. Jennifer Kruschwitz
Advisor: Eastman Clarke

Document Number 00007

Revisions Level	Date
G	05-05-2018

Authentication Block

Dual-Angle Imaging Camera for Determination of Dielectric Thin Film Thickness
Design Description Document

Rev	Description	Date	Authorization
A	Initial PRD	01-21-2018	DM, YD
B	Calibration/Set up/ Part List updates	02-04-2018	DM, YD, ZT
C	Algorithm updates	02-19-2018	DM, YD
D	Midterm DDD	02-25-2018	DM, YD, ZT
E	Updated Set Up	04-01-2018	DM, YD
F	Final Review	04-19-2018	DM, YD, ZT
G	Final DDD	05-05-2018	DM, YD, ZT

Dual-Angle Imaging Camera for Determination of Dielectric Thin Film Thickness
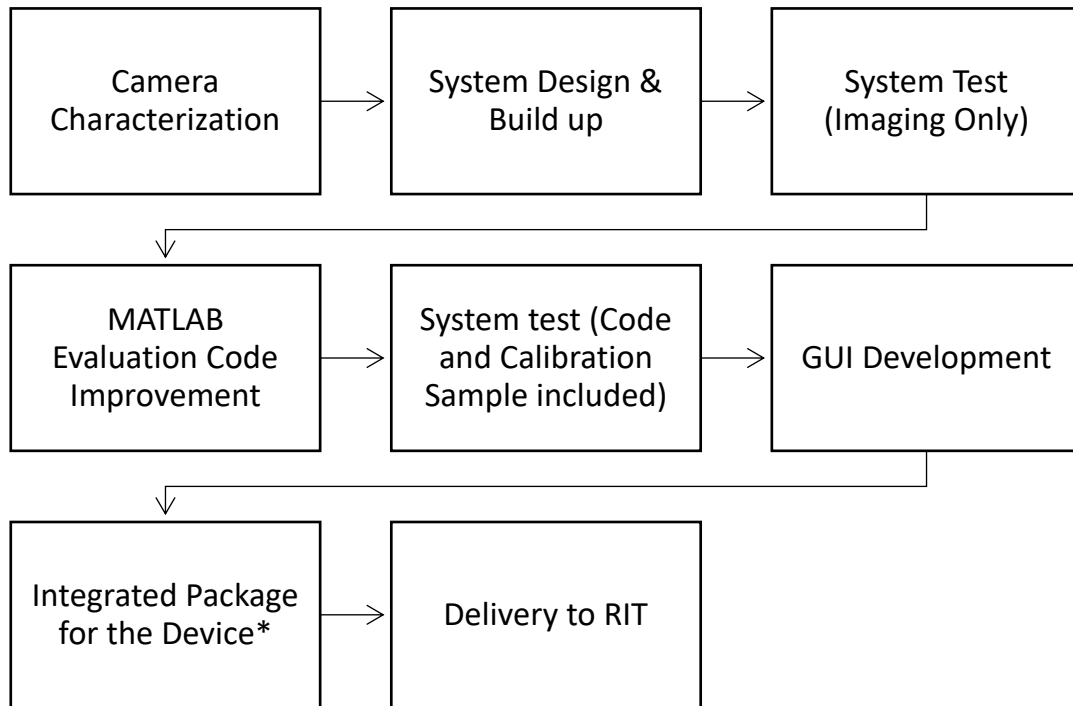Design Description Document

Contents

Dual-Angle Imaging Camera for Determination of Dielectric Thin Film Thickness
Design Description Document

I.   Vision Statement

The Dual-Angle Imaging Camera is currently being developed as a spectrophotometer to determine dielectric thin film thickness. Our goal is developing and producing a dual-angle camera imaging system for determining dielectric thin film thickness in the RIT Microelectronics Cleanroom Lab. The product should be ready to use at the end of this project.

II.   Project Scope

We are responsible for the design and development of the dual-angle imaging camera including imaging test with developed MATLAB code and calibrated samples. If time permits, we are also responsible for the user-friendly GUI development and device package design.

```
┌──────────────────┐      ┌──────────────────┐      ┌──────────────────┐
│     Camera       │ ───> │  System Design & │ ───> │   System Test    │
│ Characterization │      │     Build up     │      │  (Imaging Only)  │
└──────────────────┘      └──────────────────┘      └──────────────────┘
         │                                                     │
         ▼                                                     │
┌──────────────────┐      ┌──────────────────┐      ┌──────────────────┐
│     MATLAB       │      │ System test (Code│      │                  │
│ Evaluation Code  │ ───> │  and Calibration │ ───> │ GUI Development  │
│  Improvement     │      │ Sample included) │      │                  │
└──────────────────┘      └──────────────────┘      └──────────────────┘
         │                                                     │
         ▼                                                     │
┌──────────────────┐      ┌──────────────────┐
│ Integrated Package│ ───> │                  │
│ for the Device*  │      │  Delivery to RIT │
└──────────────────┘      └──────────────────┘
```

*Package design will only be performed if time allows.

III. Background

The thickness of optical thin films, such as dielectric thin films can be measured by several instruments such as a profilometer, spectrophotometer, or interferometer. However, these methods have limitations including surface damage and size limitation. Also, all the above techniques use costly instruments to measure optical thin film thickness. Therefore, a cheaper and practical method is required for optical thin film thickness measurements, especially for an organization with a limited budget such as an undergraduate teaching lab.

Indeed, Professor Kruschwitz has determined a simple method for determining the physical thickness of a dielectric film on a silicon wafer using two images from a digital camera. From Professor Kruschwitz's paper, as a dielectric film increases in thickness on a substrate of higher refractive index $(n_s > n_f)$, a maximum in reflectance will fall at a given wavelength, λ_1, for a film with refractive index n_f and physical thickness t when [1]

$$\frac{2\pi n_f t}{\lambda_1} = m\pi \tag{1}$$

A reflectance minimum will fall at λ_2 when

$$\frac{2\pi n_f t}{\lambda_2} = \left(\frac{2m+1}{2}\right)\pi \tag{2}$$

Combining both Eqs. (1) and (2), and solving for physical thickness, t, produces

$$t = \frac{\lambda_1 \lambda_2}{4 n_f (\lambda_1 - \lambda_2)} \tag{2}$$

Therefore, a digital color camera with two incident angles on the sample silicon wafer will be a straightforward way to measure the thin film thickness. The key for the dual-angle camera imaging system to be functional is for the system to be able to capture color change on the silicon wafer that is capable of calculating the thickness of a thin film. The dual-angle imaging camera will only need a training target, a light source, and camera imaging system to reconstruct the spectral reflectance from RGB data. With the reconstructed reflectance for the training sample, the system should be able to match the reflectance data of any unknown thin film thickness to the legacy database that has been generated before this project.

IV.   Dual-Angle Imaging System Overview

I.   *Design Performance/Requirement*
- i.   Desired angle difference between center and edge rays ~ 10 degrees.
- ii.   Measurement of thin film thickness ranges from 0 – 600 nm with 10nm interval.
- iii.   The sample (silicon dioxide wafer) should be horizontally placed on the table while the measurements are performed by the dual-angle camera imaging system.
- iv.   Real time results feedback.

II.   *Optical System Design*
Our group has developed various optical designs for the dual-angle camera imaging system and reached a final design that will be used as our prototype. The first design was developed to achieve collimated illumination on the silicon wafer sample while meeting design requirements. However, the second design was updated from the first design due to several factors which included the customer's needs, easiness of operations, and optical performance. After testing both systems, it was found that the second system, with only one lens, actually provides a better optical performance. After knowing single lens system meets the customer's requirements, the group moved forward to find an optimal design form that is our current final design.

- i.   Initial Design
As shown in figure 1 and figure 2, our group set up our first optical design with two lenses to achieve collimated illumination of the silicon wafer sample. Originally, we were thinking the collimated illumination would provide us with more accurate results for better images.
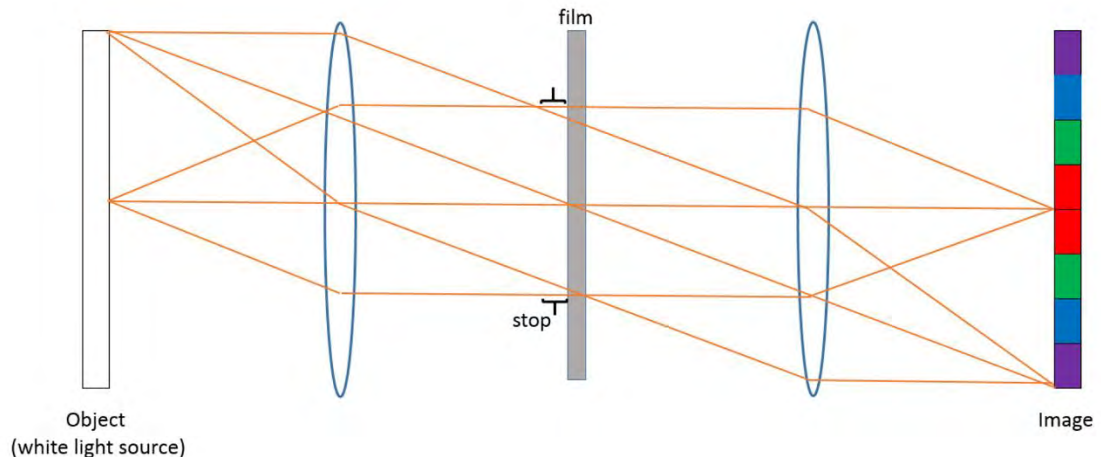

Figure 1. The simple optical layout of our first optical system design. Two lenses are used to achieve collimated illumination.
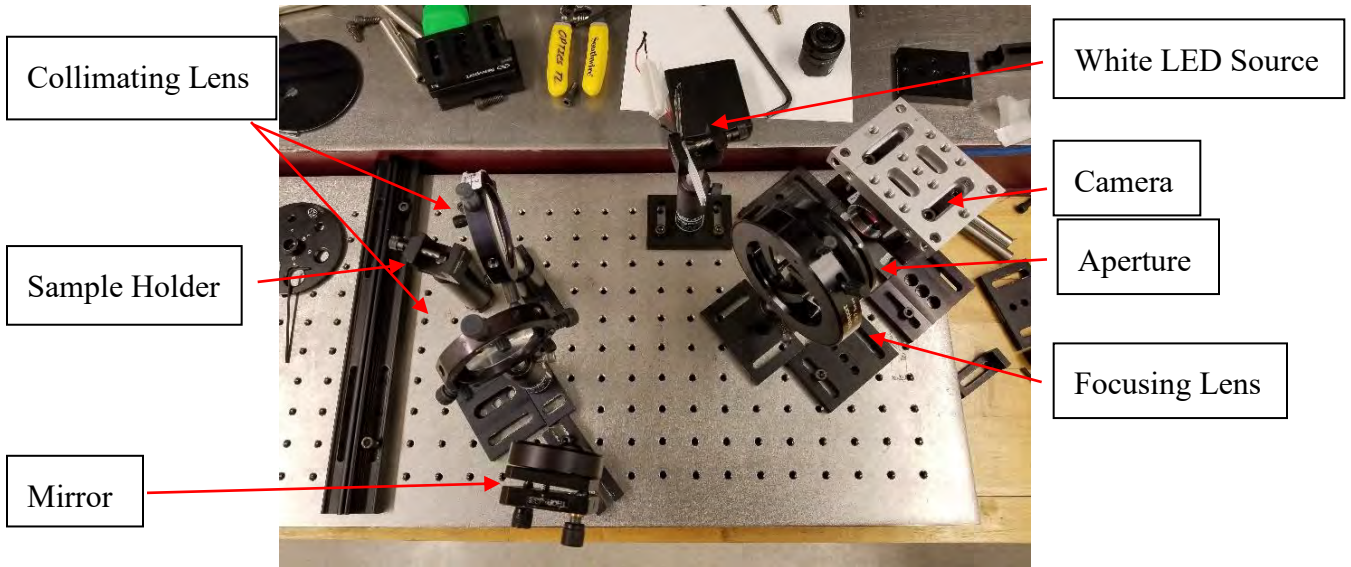
Figure 2. Image of the real optical setup for the first optical design. One mirror and camera focusing lens were added for camera performance and system size control.

Figure 2 clearly shows that we might have used too many optical elements for a simple camera system. After discussing with our customer, we realized that we did not need to consider image quality as our priority as the camera will be trained by the training sample to recognize reflectance from RGB data. Thus, our lens design expert, Zheng Tan, proposed a new design to have a simpler system with less components making it cheaper, more user friendly, and a more reliable system.

ii. The Second design
The second design was built last week in the lab and some initial measurements were performed to test its performance and to compare it to the first optical design.
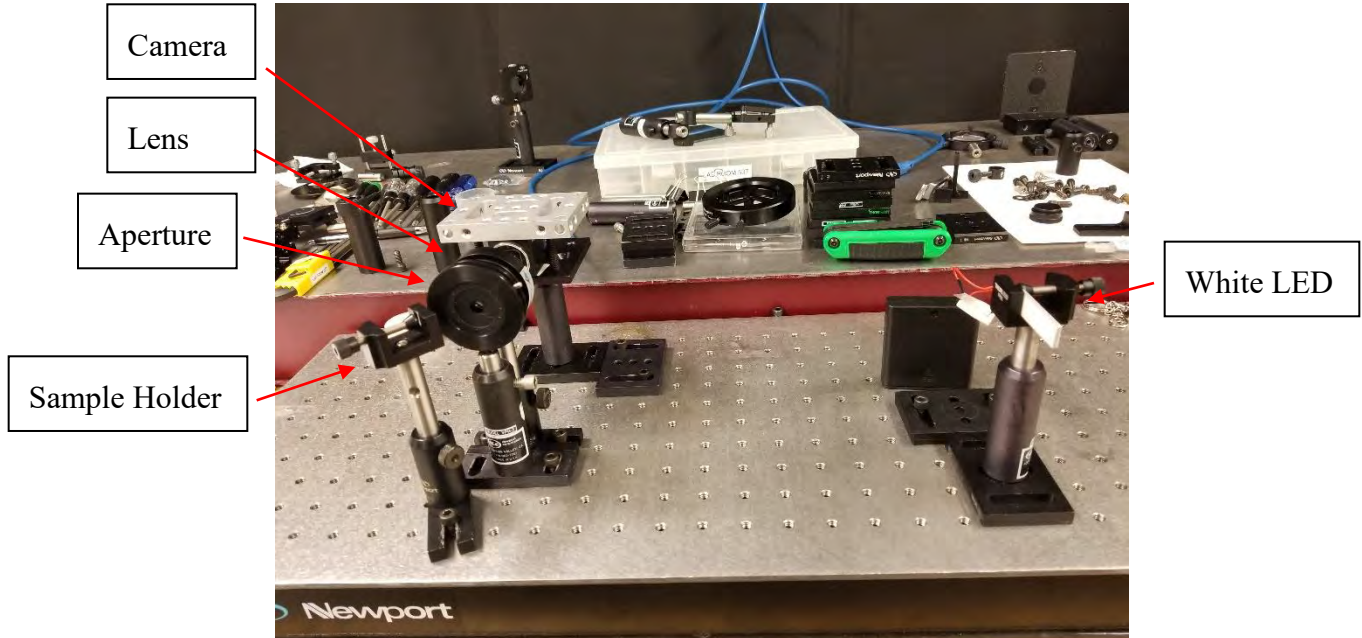
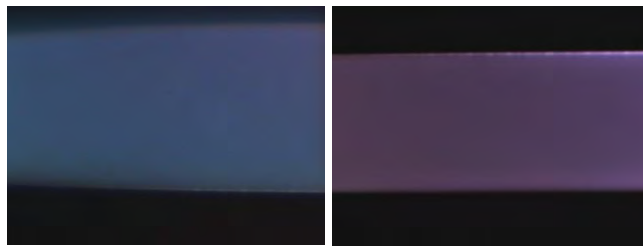**Figure 3.** Image of the real set up of the second optical design.

Figure 4. Left: Image of the same thickness taken by the first optical design. **Right:** Image of the same thickness taken by the second optical design.

It turned out that the second system actually has better optical performance since the system only has one lens in the system, which means less aberrations due to lenses. Also, the simpler optical system design will allow us to convert the setup to a vertical setup more easily.

iii. The Final Design

After having the experience with the first two designs, the group decided to transform the set up into a vertical testing system. Since there were several intermediate steps during the transformation, photos were not taken, as the design was improved rather quickly on a daily basis. The final design was determined based on size, package, and budget.
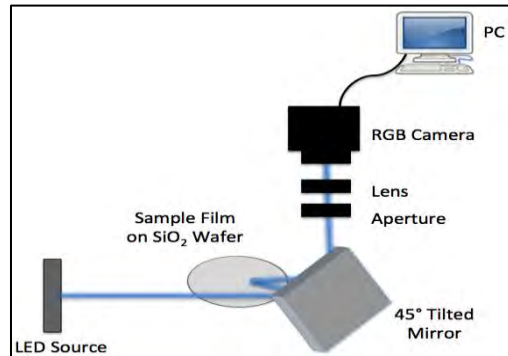

Figure 5. Schematic of the final system setup.

In the final system set-up, a white LED source is shined to a mirror that is titled 45° from the z-axis. The light then reflects to the sample, a dielectric thin film on a SiO_2 wafer, then back to the mirror. Finally, the light is reflected to an aperture, through a lens and into an RBG camera. The RGB camera is connected to a computer so that its RGB information can be analyzed.
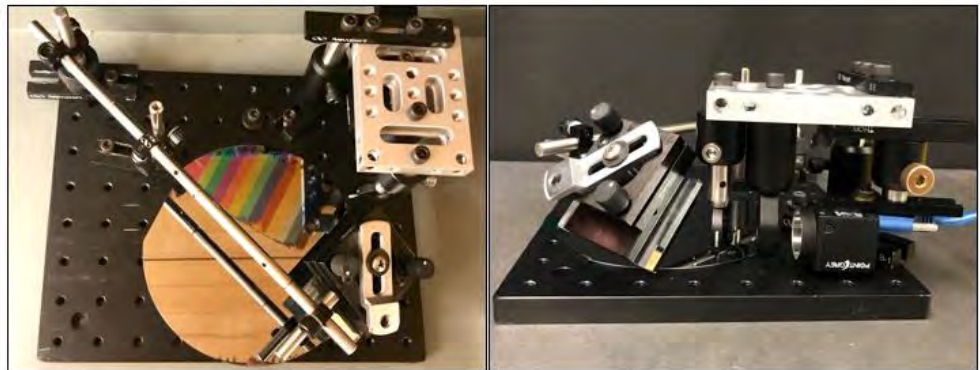

Figure 6. Image of the real set up of the final optical design.
Left: Top view of the system. **Right:** Side view of the system.
The source is missing in these images.

With the fact that the final design is our most compact and simple design with only one lens, the group is able to control the total cost of the system to be in the $1000 range. The performance of the final system will also be illustrated in the performance section.

iv. Considerations on Effect of Aberrations:
Our system uses lenses to form an image of the source to see the color change, which is our primary goal. Imaging systems' standards don't apply the same way to our system, for example, MTF is not an essential factor for us. We can tolerate aberrations as long as they don't change the color on each pixel. One way to model how aberrations effect detected color is to use spot diagrams. A point source would create a set of spots on the image plane and point sources close enough will have overlapping sets of spots. We plan to pick some source points at the object plane so that the resulting spot diagram densely fills a few pixels (current plan is 4 pixels) of our CCD camera. The spot diagram is generated by ray tracing at the d.F.c spectrum. Then the number of rays falling in each pixel corresponding to each wavelength will be counted, and the ratio between them will be calculated. We will verify, first, if this ratio stays the same across different pixels, and second if the ratio at the detector plane is the same as the source plane (which is assumed to be 1:1:1). The simulation is carried out in Matlab and below are some results:
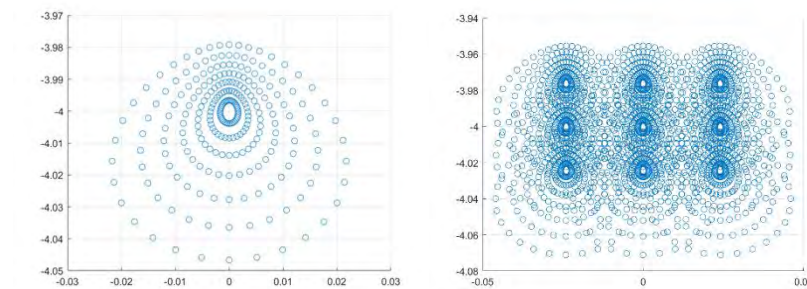

Figure 7. Left: spot diagram from a single point source and Right: overlapping diagrams from multiple point sources. Axis units are millimeters.

	486nm	589nm	656nm
Pixel 1	0.996478647800756	0.995390229848262	1
Pixel 2	1.00051209832288	0.999039815644604	1
Pixel 3	0.996579321027495	0.99528498773719	1
Pixel 4	1.00032260145816	0.99909671591716	1

Table 1. 3 selected wavelengths' ray number ratios calculated by dividing 656nm ray number at four pixels at maximum field

We can see from table 1 that the ratio remains close to 1:1:1, which is the ratio the source has. This implies that the spectral power distribution would remain almost unchanged after light goes through the lens and reaches the detector. Since spectral power distribution is

directly related to color detected by the camera, we see that aberration from a singlet would not affect the color detection of the source. From this, we conclude that aberrations would not be a primary concern for the design of the system.

Single lens setup:
Considering that our reflectance algorithm functions are based on calibrated data and in a way adaptive, we think it has the potential to function without a collimated light beam being incident on the film. Shown below is the single lens system setup we are currently experimenting with. The stop is set to be small so that the ray bundle stays close to being collimated. A smaller stop also decreases aberration of the system. We will need more data to see if this setup is possible.

Figure 8. The single lens set up. In actual setup the film acts as a 45-degree horizontal mirror.

Dual-Angle Imaging Camera for Determination of Dielectric Thin Film Thickness
Design Description Document

III.   *Final Design Performance and Guideline*
        i.   Final Design Performance
             Although the final design only has one lens, the results from the final
             design show us that the single lens set-up generally provides the best
             performance and accuracy. Figure 9 shows the promising results we
             have from the final design. As the error bar is too small to show in
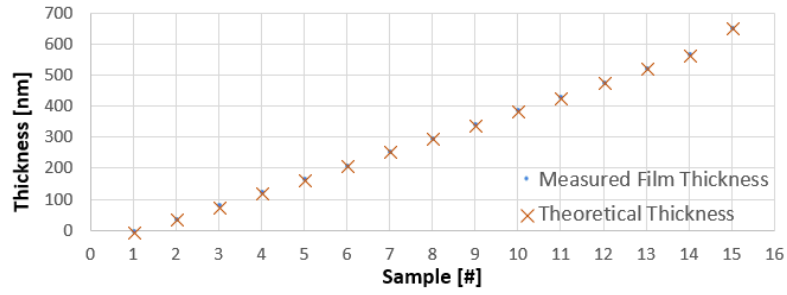   figure 9, figure 10 is was made to show the errors.

Figure 9. Plot of the measured thickness compared to the theoretical
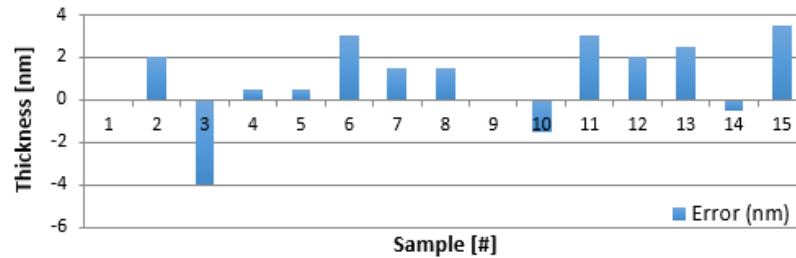thickness for each training samples.

Figure 10. Plot of the measured thickness compared to the theoretical
thickness for each training sample.

However, all the results are measure in a dark room. The group is still
working on a package that can provide a relatively dark measurement
environment. It is also possible that the group would like to improve
the final design's performance in a room with the lights on. Some
problems with the prototype have been shown in testing and on design
day, which will be addressed in a future section.
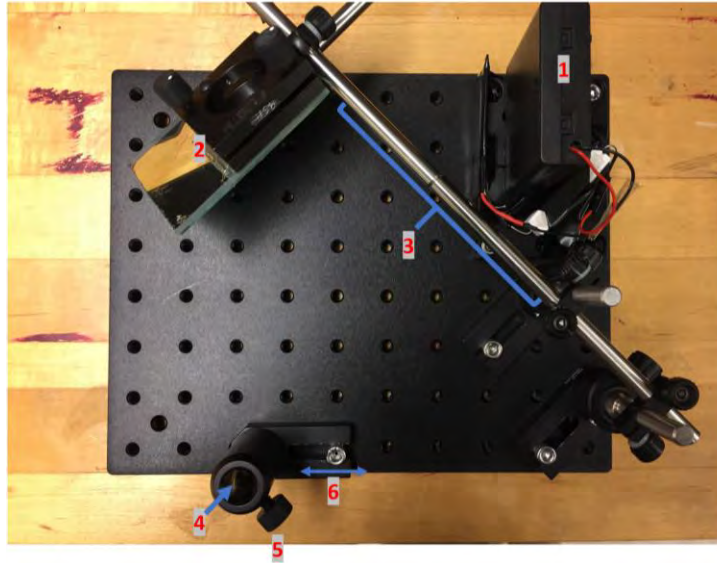
ii. Final Design Alignment Guideline

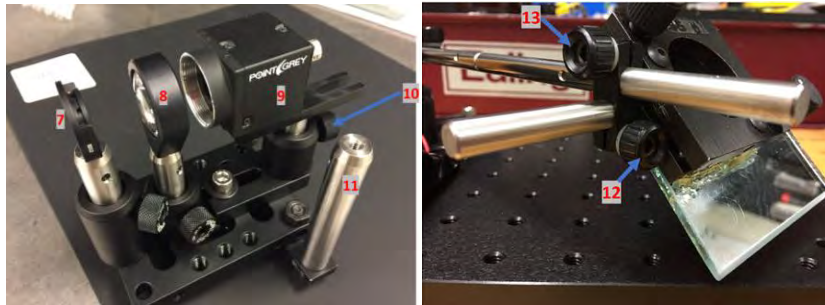**Figure 11.** Setup with imaging board removed.

Figure 12. Left: imaging board containing the aperture, lens and camera. Right: two knobs next to the mirror.

Below are steps to align the system, some steps can be skipped depending on the situation. Numbers in parentheses refers to the corresponding parts labeled in Figures 11 and 12.

1. Fix the LED source (1) at the top right corner (as oriented in figure 11).

2. Make sure the mirror support (3) is at 45 degrees.

3. Remove camera (9) from the imaging board and adjust the aperture (7) and lens (8) so that they share the same optical axis. Then put the camera back in and make sure it is horizontally in line with the lens, it's OK if the camera departs a little vertically.

4. Connect the camera to the computer and run the program to start the camera for the preview.

5. Insert the imaging board support (11) in the holder (4) with the aperture facing the mirror. Adjust the height so that the center of the aperture is at the same height as the center of the LED source. Do not change the height of the imaging board in later steps. The aperture and lens should always be at the same height as the LED source.

6. Place the bare silicon wafer beneath the mirror.

7. Gently loosen (12) and rotate the mirror (2) so that it's 45 degrees from the vertical (z direction). If done properly the image (a long thin bar) viewed from the camera would be horizontal.

   The camera view can be very helpful. It's OK if the image is departed from the center of the view for now. If the image goes outside of the camera's view, adjust the height of the camera through (10). Fix the mirror in place after this step.

8. Adjust the base (6) to left/right and/or rotate (11) so that horizontally the center of the image is at the center of the view. Height of the imaging board should not be changed.

9. Adjust the camera position through (10) so that both horizontally and vertically the center of image match center of camera view.

Note: Loosening the knob (13) will allow you to slide the mirror (2) along the support (3), this will change the position of the optical axis relative to the baseboard and hence the source and imaging board positions. This is not recommended unless necessary.

IV. *General Design Process*

The first step of the overall system setup is gamma calibration of the camera. We were hoping to have the ideal case in which the camera is linear. If the camera is not linear then an algorithm needs to be written to compensate for the nonlinear gamma. A calibrated camera appears as a linear plot of gamma as a function of digital counts for the R, G, and B channels. After calibrating the camera, we needed to write an algorithm to calculate the spectral reflectance of any dielectric thin film and then match the reconstructed reflectance for the 15 films on the training sample and 61 thickness samples to a database of film thicknesses

I. Controlling camera with Matlab
- Wrote MATLAB code to establish communication with the camera. The group purchased a particular Pointgrey camera that has an interface compatible with MATLAB.
- All camera settings were manually set to optimized values for image capturing, which includes gamma, shutter speed, and white balance.

II. Camera calibration
- We took reflection data of the gray scale squares on a camera calibration checkerboard (the bottom six squares on the color checkerboard. See figure 8 in appendix) with a spectrophotometer.
- The gamma of the camera was calculated on MATLAB using the reflectance data, color matching functions, and luminance (see appendix for MATLAB code).
- We took an image of the gray scale but because the lighting was not uniform, a white paper was placed in front of the checkerboard (see figure 9 in appendix) and an image was taken. Then the image of the checkerboard was divided by the image of the white paper. This got rid of the cosine fall off from the light source and any shadows.
- Then the digital counts for R, G, and B of the gray scale were plotted vs the gamma.
- Gamma turned out to be approximately linear (see plot below). After consulting with our customer and faculty advisor, the group decided that we will assume the camera to be linear in the later design and testing process.
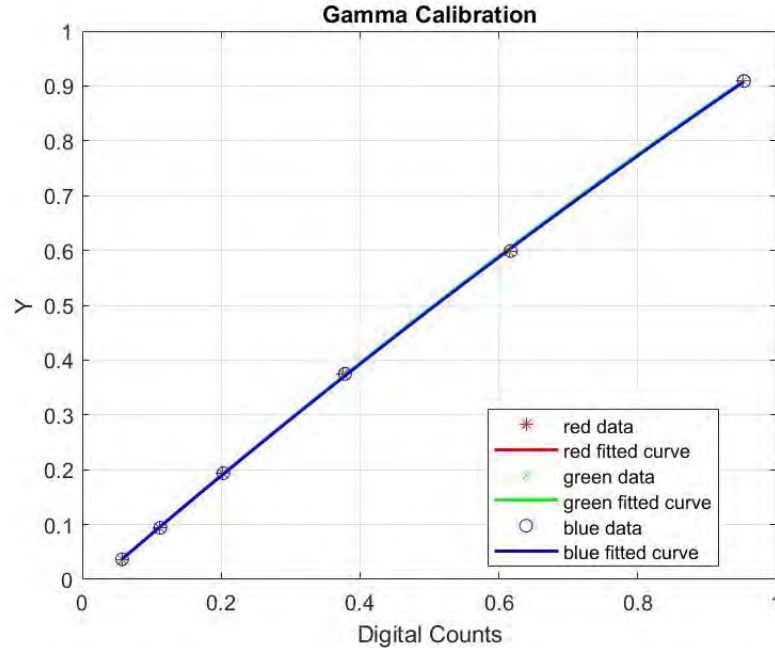
Figure 13. The Y axis has value Y which is gamma
of the camera. Both gamma and digital counts have
been normalized to see if the camera is linear. The
curves for the R and G channels cannot be seen
because they are overlapping with the B data.

III. A complete MATLAB algorithm has been developed to calculate
sample thin film thickness. The code covers the following steps: 1.
Camera initialization. 2. Data saving in folders setup. 2. Thickness
measurements with instructions (Code will be further developed to test
any user defined numbers of thickness) 3. All the measurements will
be converted into an RGB matrix that will be used in spectral
reflectance conversion.

IV. Wrote MATLAB code to calculate the spectral reflectance of any
dielectric thin film thickness, $R_{\lambda,1-q}$, using the equation below:

$$R_{\lambda,1-q} = \bar{R}_\lambda + e_{\lambda,m} * M * \left(\begin{bmatrix} r_{\theta_1,1} & \cdots & r_{\theta_1,q} \\ g_{\theta_1,1} & \cdots & g_{\theta_1,q} \\ b_{\theta_1,1} & \cdots & b_{\theta_1,q} \\ r_{\theta_2,1} & \cdots & r_{\theta_2,q} \\ g_{\theta_2,1} & \cdots & g_{\theta_2,q} \\ b_{\theta_2,1} & \cdots & b_{\theta_2,q} \end{bmatrix} - \begin{bmatrix} \bar{r}_{\theta_1} \\ \bar{g}_{\theta_1} \\ \bar{b}_{\theta_1} \\ \bar{r}_{\theta_2} \\ \bar{g}_{\theta_2} \\ \bar{b}_{\theta_2} \end{bmatrix} \right)$$

Figure 14. Equation 5 from Reference#1

where R_λ is the mean spectral reflectance of the reflectance dataset,
$e_{\lambda,m}$ are the eigenvector scalars derived from the measured color

standard spectral reflectance, M is the transformation matrix that transforms spectral reflectance data to RGB color space, $r_{\theta1,q}$ are the digital counts for the q(=15) film thicknesses for the image at the first angle, $r_{\theta2,q}$ are the digital counts for the q(=15) film thicknesses for the image at the second angle, and $r_{\theta1}$ are the average digital counts of the image at each angle. In order to have M for the equation above, it needed to be calculated from a known set of reflectance measurements. So we wrote MATLAB code to solve for M using known spectral reflectance data collected with a spectrophotometer for the 15 different film thicknesses on the training target.
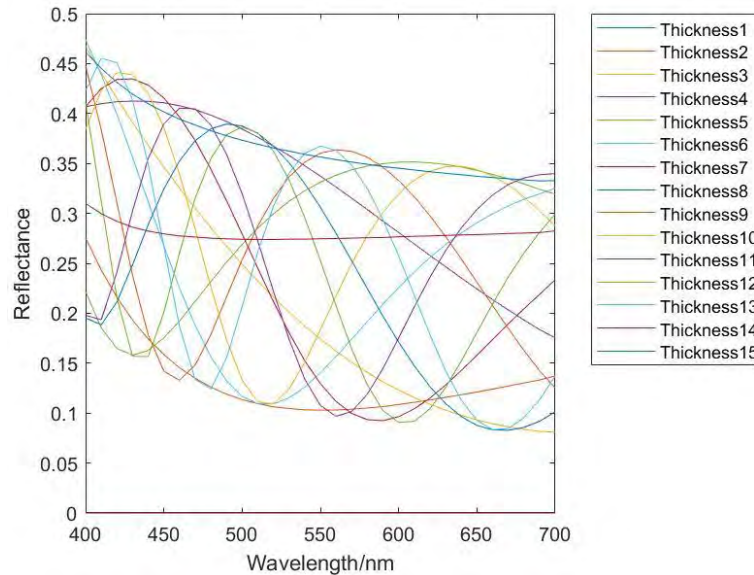

Figure 15. The results from the MATLAB code using
the reflectance calculation algorithm.

- All the MATLAB codes have been combined into a single file so that the entire process from measurements to data analysis will be performed in a single file and results will be real-time feedback. MATLAB code was developed to match the reconstructed reflectance for the 15 films on the training sample and 61 thickness samples to a database of film thicknesses.
- A basic error analysis was carried out on the results by comparing the measurements to results from the spectrophotometer.
- A simple GUI has been designed for freshman usage at RIT Microelectronics Lab.

V. Cost Analysis

With all the final components confirmed for the final design, the group submitted the following list for the final product/prototype build up. The total cost is $0.94 over our $1000 budget, but the purchases were approved by the customer.

Item	Quantity	Description	Cost	Link
3" post	3		24	https://www.thorlabs.com/thorproduct.cfm?partnumber=TR3
2"post	5		35.45	https://www.thorlabs.com/thorproduct.cfm?partnumber=TR2
Post Holder Bases	4		20.8	https://www.thorlabs.com/thorproduct.cfm?partnumber=BA1S
Newport 1" post holders	2		30	https://www.newport.com/p/VPH-1
Right Angle Post Clamp	3		51	https://www.thorlabs.com/thorproduct.cfm?partnumber=MSRA90
Lens	1	LB1761, f=25.4mm	26	https://www.thorlabs.com/thorproduct.cfm?partnumber=LB1761
Base plate	1		98	https://www.newport.com/p/CB-2
Newport M-BC-15	1		5.5	http://www.lightglassoptics.com/Newport-M-BC-15-Metric-Adjustable-Height-Base-Clamp_p_278.html
2"post holder	1		7.7	https://www.thorlabs.com/thorproduct.cfm?partnumber=PH2#ad-image-0
1.5"post holder	1		7.22	https://www.thorlabs.com/thorproduct.cfm?partnumber=PH1.5#ad-image-0
1"post holder	2		7.03	https://www.thorlabs.com/thorproduct.cfm?partnumber=PH1#ad-image-0
2" post	1		5.19	https://www.thorlabs.com/thorproduct.cfm?partnumber=TR2
1.5"post	1		4.97	https://www.thorlabs.com/thorproduct.cfm?partnumber=TR1.5
1" post	2		9.48	https://www.thorlabs.com/thorproduct.cfm?partnumber=TR1
Lens holder	1	LMR1	16	https://www.thorlabs.com/thorproduct.cfm?partnumber=LMR1
Screws	4	5 pack	48.6	https://www.thorlabs.com/thorproduct.cfm?partnumber=SH25S063V
Small aperture	1	12mm	48	https://www.thorlabs.com/thorproduct.cfm?partnumber=ID12
Camera Holder	1		25	https://www.edmundoptics.com/cameras/gigabit-ethernet-cameras/Blackflyreg-frac14-20-Tripod-Adapter/
Camera	1		395	https://www.edmundoptics.com/cameras/usb-cameras/bfly-u3-03s2c-cs-usb-3.0-blackfly-colorcamera/
Source	1	White Light LED includi	20	https://www.adafruit.com/product/1626
Breadboard	1	8x10'	116	https://www.thorlabs.com/thorproduct.cfm?partnumber=MB810
TOTAL			1000.94	

Table 2. The final list of all components needed for the final design build up.

VI. Plans for next step

1. There are still lots of improvements we can work on for the final design. Zheng will be staying this summer to continue optimizing the prototype, and he will also try to develop new designs for more complicated usage. Diana is also staying this summer at LLE, so she will also help Zheng on the system build up. Yang will not be in Rochester this summer, but he will take any MATLAB, software, and camera control problems remotely.

2. One of the design requirements comes from our chemical engineering department. A professor would like our design in his lab to measure polymer thin film thickness during the deposition process. A telescope system might be needed for this design requirement. This plan can be combined with our design day feedback to provide a new senior design project for the class of 2019.

3. The room light problem will also need to be solved. The ideal solution is by optimizing our algorithm that way we can solve this problem without physically changing anything. Otherwise, a designed package might be needed for the prototype.

Schedule

Week 1 Feb 4-10:
Finish camera gamma calibration and Current design testing. Read Professor Kruschwitz's paper and discuss with Professor. Kruschwitz on the algorithm to develop the MATLAB code.

Week 2 Feb 11-17:
Thicknesses measurement on sample silicon wafers for initial system testing without algorithm analysis.

Week 3 Feb 18-24:
The part list should be ready to purchase all setup components; Collect basis eigenvectors from Professor. Kruschwitz to carry out algorithm calculations.

Week 4 Feb 25-Mar 3:
Have characterization samples ready. Finish algorithm MATLAB code.

Week 5 Mar 4-Mar10:
Have measurements from calibration samples and testing the algorithm MATLAB code on the calibration samples.

Week 6 Mar 18-24:
GUI development and system testing.

Week 7 Mar 25-Mar31:
Set up Improvement, GUI development

Week 8 Apr 1- Apr 7:
GUI development

Week 9 Apr 8-14:
Final system build up and testing

Week 10 Apr 15-21:
Optimization of system and error anlysis

Week 11 Apr 22-28:
Complete and print poster

Week 9 Apr 29-May 5:
More error analysis

00007   Rev G

Appendix

Appendix A: Pointgrey Camera Specification

Resolution	808 x 608
Frame Rate	50 FPS
Megapixels	0.5 MP
Chroma	Color
Sensor Name	Sony ICX693
Sensor Type	CCD
Readout Method	Global shutter
Sensor Format	1/3"
Pixel Size	6.0 μm
Lens Mount	CS-mount (5 mm C-mount adapter not included)
ADC	12-bit
Quantum Efficiency Blue (% at 470 nm)	58
Quantum Efficiency Green (% at 525 nm)	60
Quantum Efficiency Red (% at 640 nm)	50
Temporal Dark Noise (e-)	9.20
Absolute Sensitivity Threshold (γ)	17.13
Saturation Capacity (e-)	21516
Dynamic Range (dB)	66.92
Gain Range	0 dB to 23.991 dB
Exposure Range	0.031 ms to 31.9 seconds
Trigger Modes	Standard, bulb, low smear, overlapped, multi-shot
Partial Image Modes	Pixel binning, decimation, ROI
Image Processing	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	16 MB
User Sets	2 user configuration sets for custom camera settings
Flash Memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output
Auxiliary Output	3.3 V, 120 mA maximum
Interface	USB 3.1 Gen 1
Power Requirements	5-24 V via GPIO or 5 V via USB3
Dimensions	29 mm x 29 mm x 30 mm
Mass	36 grams
Machine Vision Standard	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature (Operating)	0° to 45°C
Temperature (Storage)	-30° to 60°C
Humidity (Operating)	20 to 80% (no condensation)
Humidity (Storage)	20 to 95% (no condensation)
Warranty	3 years

00007 Rev G

Appendix B: Camera Calibration
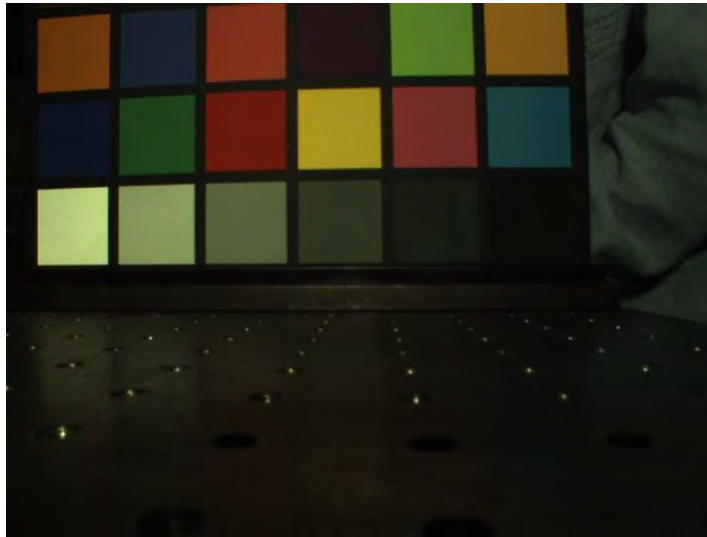
Figure 16: Camera Calibration Color Checkerboard.
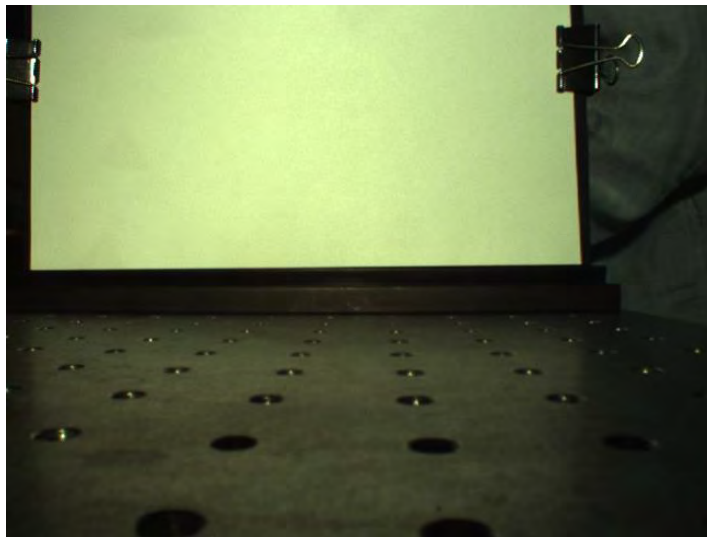
Figure 17: White paper placed in front of color
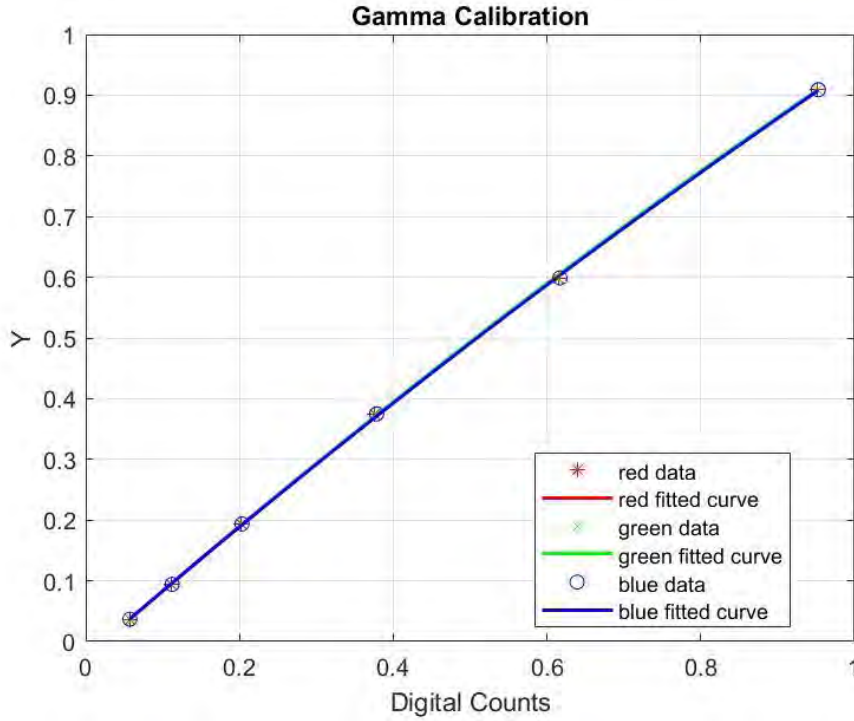checkerboard.

Figure 18: The Y axis has value Y which is gamma of the camera. Both gamma and digital counts have been normalized to see if the camera is linear. The curves for the R and G channels cannot be seen because they are overlapping with the B data.
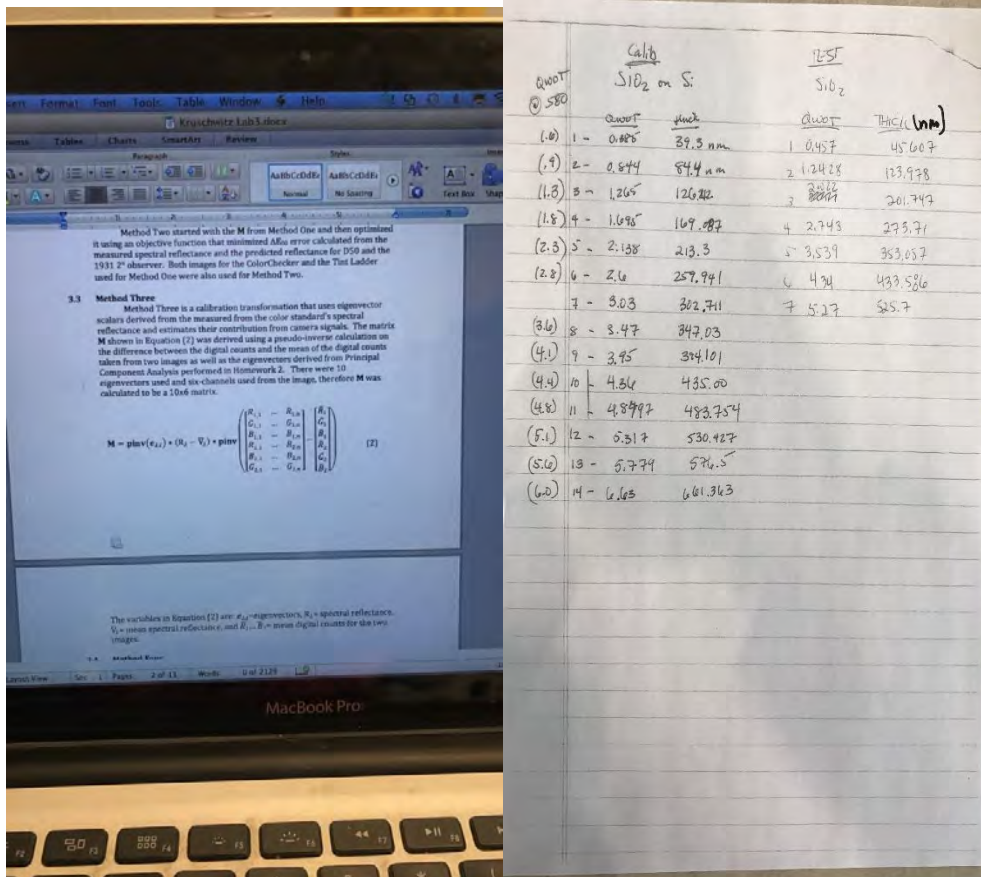
Appendix C: Algorithm Reference

Figure 19: Left: Equation used to solve for the transformation matrix, M.
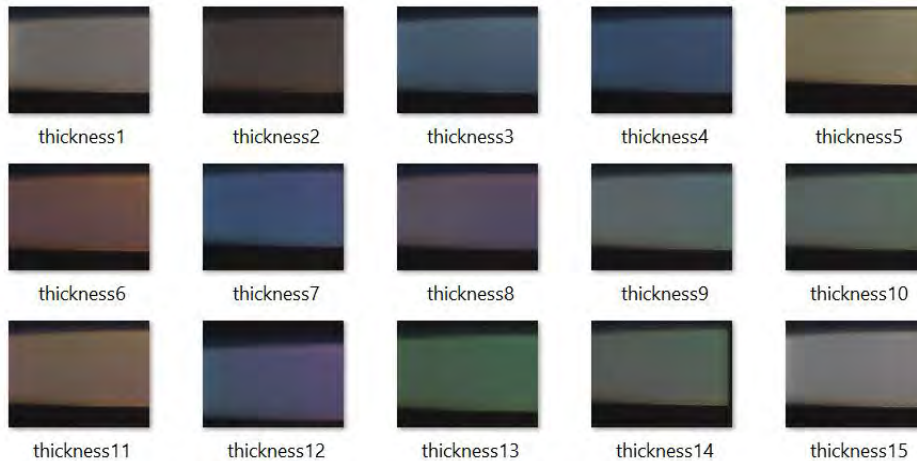**Right:** The known quarter wave optical thicknesses on the test targets.

Figure 20: Images of films taken with the first optical system.

Appendix D: MATLAB Code

Gamma calibration

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Optical Engineering Class 2018 Team Films Senior Desgin
%
%   This is used for camera calibration by fitting curve on gamma for
%   future measurement references.
%
%   cmf2D.mat and ref2XYZ.m are the intellectual property of
%   Dr. Kruschwitz, the group used these functions with permission from
%   Dr. Kruschwitz.
%
%   Editor: Yang Deng
%   Last Update: 01/26/2018 by Yang
%   Update: using images from new directory - 01/21
%           new images at differnt settings
%           all camera settings at manual - 01/26
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load('ref_37.mat')          % Measured reflectance data after spline fit
load('cmf2D.mat')           % Transformation

M_color = imread('image1_color.tiff');          % read captured images
A_color = double(M_color);

M_white = imread('image1_white.tiff');
A_white = double(M_white);

A = (A_color./A_white).*0.92;           % eliminate oversaturated pixels

figure
image(A);

I1 = imcrop(A,[45, 190, 36, 36]);               % crop first 6th color from color checker
I1_R = I1(:,1);
I1_G = I1(:,2);
I1_B = I1(:,3);
I2 = imcrop(A,[120, 190, 36, 36]);
I2_R = I2(:,1);
I2_G = I2(:,2);
I2_B = I2(:,3);
I3 = imcrop(A,[200, 190, 36, 36]);
I3_R = I3(:,1);
I3_G = I3(:,2);
I3_B = I3(:,3);
I4 = imcrop(A,[288, 190, 36, 36]);
I4_R = I4(:,1);
I4_G = I4(:,2);
I4_B = I4(:,3);
I5 = imcrop(A,[369, 190, 36, 36]);
I5_R = I5(:,1);
I5_G = I5(:,2);
I5_B = I5(:,3);
I6 = imcrop(A,[458, 190, 36, 36]);
I6_R = I6(:,1);
I6_G = I6(:,2);
I6_B = I6(:,3);

R = transpose(ref_37);          % using ref2XYZ to convert data into Y
E = ones(37,1);
out = ref2XYZ(R,Cmf2D,E);
out1 = out(2,:);
Y1 = out1.';
Y = Y1./10000;
```

00007   Rev G

```matlab
X_R = zeros(6,1);            % define digital counts for RGB channels
X_R(1,1) = mean(I1_R);
X_R(2,1) = mean(I2_R);
X_R(3,1) = mean(I3_R);
X_R(4,1) = mean(I4_R);
X_R(5,1) = mean(I5_R);
X_R(6,1) = mean(I6_R);

X_G = zeros(6,1);
X_G(1,1) = mean(I1_G);
X_G(2,1) = mean(I2_G);
X_G(3,1) = mean(I3_G);
X_G(4,1) = mean(I4_G);
X_G(5,1) = mean(I5_G);
X_G(6,1) = mean(I6_G);

X_B = zeros(6,1);
X_B(1,1) = mean(I1_B);
X_B(2,1) = mean(I2_B);
X_B(3,1) = mean(I3_B);
X_B(4,1) = mean(I4_B);
X_B(5,1) = mean(I5_B);
X_B(6,1) = mean(I6_B);

f_R = fit(X_R,Y,'poly2');            %fit
f_G = fit(X_G,Y,'poly2');
f_B = fit(X_B,Y,'poly2');

r = plot(f_R,'r',X_R,Y,'*r');            %plot
hold on
g = plot(f_G,'g',X_G,Y,'xg');
hold on
b = plot(f_B,'b',X_B,Y,'ob');

set(r(2),'Linewidth',1.5);
set(g(2),'Linewidth',1.5);
set(b(2),'Linewidth',1.5);
title('Gamma Calibaration');
xlabel('Digital Counts');
ylabel('Y');
legend('red data','red fitted curve','green data','green fitted curve','blue data','blue
fitted curve','Location','best');
grid on
```

System measurements

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Optical Engineering Class 2018 Team Films Senior Desgin
%
%   This is used for thin film thickness measuremnts and analysis.
%
%   Algorithm and formulas used in this code are the intellectual property of
%   Dr. Kruschwitz, the group used these with permission from
%   Dr. Kruschwitz.
%
%   Editor: Yang Deng, Diana Magana
%   Last Update: 02/25/2018 by Yang
%   Update: All measurements and algorithm combined - 02/25
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vid = videoinput('pointgrey', 1, 'F7_BayerBG16_648x488_Mode0');     %adjust camera
settings to have LED images in white color.
src = getselectedsource(vid);
vid.FramesPerTrigger = 1.5;
src.Brightness = 7.421875;
src.ExposureMode = 'Manual';
src.Exposure = 2;
src.FrameRateMode = 'Manual';
src.GainMode = 'Manual';
src.Gain = 0;
src.ShutterMode = 'Manual';
src.Shutter = 11;
src.WhiteBalanceRBMode = 'Manual';
src.WhiteBalanceRB = [749 505];
preview(vid);

prompt = {'Please check preview and Click YES'};              %Initintion Check
dlg_title = 'Initiation';
answer = questdlg(prompt,dlg_title);
close all

t = datetime('now','Format','yyyy-MM-dd''__''HH-mm');
T = char(t);
fnam = ['C:\Users\yangd\Documents\UR\Academic\Spring 2018\OPT311\images\' T
'\'];        %Setting storage folder route
mkdir (fnam);

for i =
1:15                                                            %Actual
different thin film thicknesses measurements
    prompt = ['Please Move to thickness' num2str(i) ' and Click Yes'];
    dlg_title = 'Change thickness';
    answer = questdlg(prompt,dlg_title);
    start(vid);
    data = getdata(vid);
    image = getsnapshot(vid);
    fnam1 = [fnam 'thickness' num2str(i) '.tiff'];
    imwrite(image,fnam1);
end

M = cell(1,15);                                          %Algorithm to achieve RGB
matrixs from the measurements
A = cell(1,15);
T = cell(1,15);
for i = 1:15                                             %Read saved images
    fnam1 = [fnam 'thickness' num2str(i) '.tiff'];
    A{i} = double(imread(fnam1));
end
```

00007   Rev G

```matlab
ref = A{i};                                        %Set reference matrix as
bare silicon
RGB = zeros(6,15);
RGB_mean = zeros(6,15);
r = cell(1,15);
g = cell(1,15);
b = cell(1,15);

for i = 1:15                                       %Input data into already
built matrix for RGB
    T{i} = (A{i}./ref)*0.5;
    r{i} = T{i}(:,:,1);
    g{i} = T{i}(:,:,2);
    b{i} = T{i}(:,:,3);
    RGB(1,i)= r{i}(220,30);
    RGB(2,i)= g{i}(220,30);
    RGB(3,i)= b{i}(220,30);
    RGB(4,i)= r{i}(220,630);
    RGB(5,i)= g{i}(220,630);
    RGB(6,i)= b{i}(220,630);
    for n = 1:6
    RGB_mean(n,i) = mean(RGB(:,i));
    end
end

load('eig_v.mat');                                 %loads given
eigenvectors%
load('mean_61.mat');                               %loads the mean
wavelengths%
load('workspace_variables.mat')
e_lambda=transpose(eig_v);                         %eigenvectors%
R_lambda=SiO2calib45deg/100;                       %Reflectance of
target films divided by 100 to have values ranging from 0 to 1%
Mean_61=repmat(m,15,1);                            %repeating mean
reflectance values in order to have same size matrix as reflectance matrix%
v_lambda=transpose(Mean_61);                       %mean reflectance
values%
M1=pinv(eig_v)*(R_lambda-v_lambda)*pinv(RGB-RGB_mean);    %matrix still missing last
part%

R = Mean_61+transpose((eig_v)*M1*(RGB-RGB_mean));

for i = 1:15
    x = 400:10:700;
    plot(x, R(i,:));
    xlabel('Wavelength');
    ylabel('Reflectivity');
    hold on
end

delete(vid);
clear
```

00007   Rev G

GUI with all the MATLAB codes for all measurements


```matlab
function varargout = test(varargin)
% TEST MATLAB code for test.fig
%      TEST, by itself, creates a new TEST or raises the existing
%      singleton*.
%
%      H = TEST returns the handle to a new TEST or the handle to
%      the existing singleton*.
%
%      TEST('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in TEST.M with the given input arguments.
%
%      TEST('Property','Value',...) creates a new TEST or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before test_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to test_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help test

% Last Modified by GUIDE v2.5 22-Apr-2018 14:46:39

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @test_OpeningFcn, ...
                   'gui_OutputFcn',  @test_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

00007   Rev G

```matlab
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before test is made visible.
function test_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to test (see VARARGIN)


% Choose default command line output for test
handles.output = hObject;
setappdata(handles.axes1,'vid',0);
setappdata(handles.axes1,'tmp',1);

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes test wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = test_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
imaqreset
vid = videoinput('pointgrey', 1, 'F7_BayerBG16_648x488_Mode0');     %adjust camera
settings to have LED images in white color.
assignin('base', 'vid', vid);
src = getselectedsource(vid);
vid.FramesPerTrigger = 1.5;
src.Brightness = 7.421875;
src.ExposureMode = 'Manual';
src.Exposure = 2;
src.FrameRateMode = 'Manual';
src.GainMode = 'Manual';
src.Gain = 0;
src.ShutterMode = 'Manual';
src.Shutter = 11;
src.WhiteBalanceRBMode = 'Manual';
src.WhiteBalanceRB = [749 505];
im = image(zeros(648, 488, 3),'parent',handles.axes1);
preview(vid,im);
setappdata(handles.axes1,'vid',vid);
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

00007   Rev G

```matlab
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
vid = getappdata(handles.axes1,'vid');
start(vid);
tmp = getsnapshot(vid);
t = datetime('now','Format','yyyy-MM-dd-ss''__''HH-mm');
T = char(t);
fnam = ['C:\images\' T '\'];        %Setting storage folder route
mkdir (fnam);
fnam1 = [fnam '.tiff'];
%imwrite(image,fnam1);
%tmp = imread('C:\images\Zheng pic 8\measurementsthickness4.tiff');
setappdata(handles.axes1,'tmp',tmp);
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% --- Executes on button press in Analyze.
function pushbutton7_Callback(hObject, eventdata, handles)
load('SI')
load('SIO')

load('t_range')
load('RGBu_simTN')
load('RGBu_sim45ang')

load('pointers')
load('white_eg')
load('white_md')


samp_num = 1;  %%% specify number of samples intended to be predicted here

%%% write path to folder of prediction samples here
%addpath('')

for i = 1:samp_num

    tmp = getappdata(handles.axes1,'tmp');

    sz = size(tmp);

    lf_eg = tmp(:,round(sz(2)*0.3),2);
    rt_eg = tmp(:,round(sz(2)*0.7),2);
    avg = (mean(lf_eg) + mean(rt_eg)) /2;
    lf_idx = find(lf_eg >= 1.5*avg);
    rt_idx = find(rt_eg >= 1.5*avg);
    lf_idx = [lf_idx(1),lf_idx(end)];
    rt_idx = [rt_idx(1),rt_idx(end)];

    up_idx = min(lf_idx(1),rt_idx(1)) - 5;
    dn_idx = max(lf_idx(2),rt_idx(2)) + 5;
    mid_ver    = round((dn_idx - up_idx)/2);

    horiz   = tmp(mid_ver + up_idx,:,2);
    hor_idx = find(horiz >= 1.5*avg);
    h_lf    = hor_idx(1);
    h_rt    = hor_idx(end);
```

00007   Rev G

```matlab
    h_len   = h_rt - h_lf;

    crop2 = tmp(up_idx:dn_idx,h_lf:h_rt,:);
%     figure          %% can check how accurate are images cropped
%     imshow(crop2);
    lf_pos = round(h_len*pointers(1));
    rt_pos = round(h_len*pointers(2));
    md_pos = round(h_len*pointers(3));

    G = 4;

    rgb0_eg(1,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, lf_pos-G : lf_pos+G, 1)));
    rgb0_eg(2,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, lf_pos-G : lf_pos+G, 2)));
    rgb0_eg(3,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, lf_pos-G : lf_pos+G, 3)));
    rgb0_eg(4,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, rt_pos-G : rt_pos+G, 1)));
    rgb0_eg(5,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, rt_pos-G : rt_pos+G, 2)));
    rgb0_eg(6,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, rt_pos-G : rt_pos+G, 3)));

    rgb0_md(1,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, md_pos-G : md_pos+G, 1)));
    rgb0_md(2,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, md_pos-G : md_pos+G, 2)));
    rgb0_md(3,i) = mean(mean(crop2(mid_ver-G : mid_ver+G, md_pos-G : md_pos+G, 3)));
end

%%% divided by white
rgbu_eg   = rgb0_eg ./ white_eg;
rgbu_md   = rgb0_md ./ white_md;


tck_pre = zeros(1,samp_num);

for i = 1:samp_num
%%% normalize off axis data
rgbu_egN   = rgbu_eg(:,i) ./norm(rgbu_eg(:,i));

dotP     = RGBu_simTN' * rgbu_egN;
diff     = sum(abs(RGBu_sim45ang - rgbu_md(:,i)));
diff     = diff';

%%% find min
[vv, pp]    = min(abs(dotP - 1) + 0.01 * diff);

%%% find corresponding thickness and store
tck_pre(i) = t_range(pp);   %%% predicted thickness stored here

end


%%%%%%% Generate reflectance curves
lambda  = linspace(400,700,31);
inc_ang = 0;    %% specify incidence in degrees

R_predicted = zeros(31,samp_num);

for i = 1:samp_num

%%% R curves stored here
R_predicted(:,i) = R_generate(1,SIO,SI,lambda,inc_ang,tck_pre(i));


plot(lambda, R_predicted(:,i),'Linewidth', 1.5,'parent',handles.axes2);
grid on
xlabel('Wavelength');
ylabel('Reflectance');
end
set(handles.text1,'String',tck_pre);
clear
```
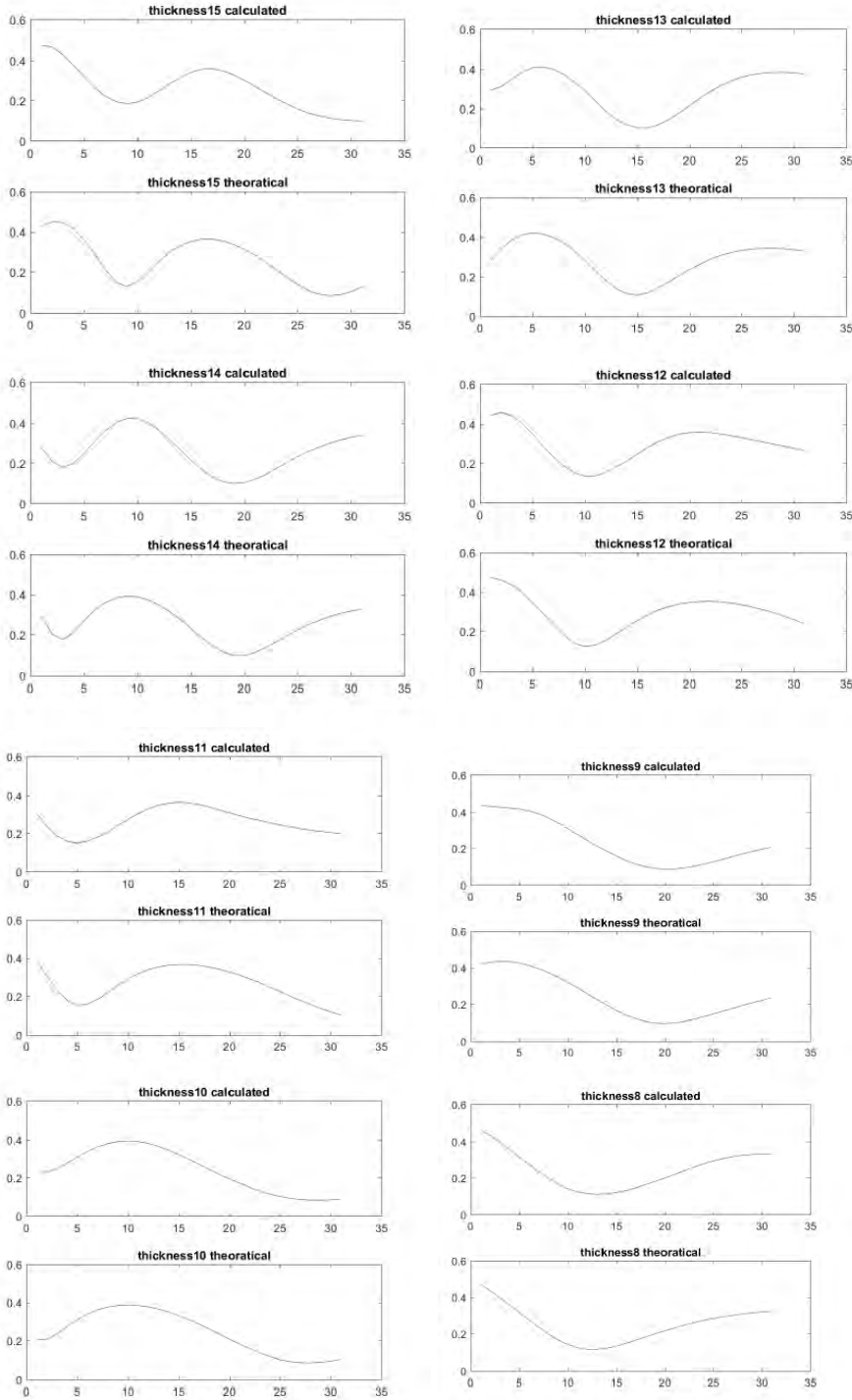
00007   Rev G

```
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Appendix E: Reflectance Curve Comparison with Spectrophotometer

Appendix F: Design Day Demo Plan

- Have the poster ready to explain how the system work.
- The system will be portable by one person based on current design. We are planning to bring the physical system setup with us on the design day.
- The group will also bring all the different film samples on the silicon wafers we have in the lab, so we can demonstrate the working ability of the dual-angle camera imaging system.
- The MATLAB GUI has been coded with the assumption that even freshman with zero coding or MATLAB knowledge will be able to use it. Therefore, the group will encourage any audience on the design day to try out our system by themselves.

Appendix G: Design Day Feedback

- Professor Tom Brown was very interested in our set up, and he suggested we build one set up for the Hopkins Center. This will be discussed, and might be included during our summer work or for next year's senior project team.
- Mr. Rick Plympton also raised a good question that if our design can measure spherical surface such as a lens from Optimax. Unfortunately, our design cannot fulfill that requirement as of yet. This question will be discussed too in our future work or in next year's senior design group.

Reference:

[1] Kruschwitz, Jennifer DT, and Roy S. Berns. "Dual-angle, spectral reconstruction imaging method for the determination of dielectric thin-film thickness." Optics letters 42.15 (2017): 3032-3035.