<u>ME6406 HW4 Thanakorn Khamvilai Report</u>

<u>Problem 1: Artificial Neural Network (ANN)</u>

1-1) Derive the weight update rule for the ANN (using a **bipolar** sigmoid function for all the processing elements).

<u>Solution:</u> The **bipolar** sigmoid function

$$h(f) = -1 + \frac{2}{1+e^{-f}} = y$$

$$h'(f) = \frac{2}{(1+e^{-f})^2}\left\{\left[(1+e^{-f})\cdot\frac{d(1)}{df}\right] - \left[1\cdot\frac{d(1+e^{-f})}{df}\right]\right\}$$

$$h'(f) = \frac{2}{(1+e^{-f})^2}\{-[(-1)\cdot(e^{-f})]\}$$

$$h'(f) = \frac{2e^{-f}}{(1+e^{-f})^2}$$

$$h'(f) = \left(\frac{2}{1+e^{-f}}\right)\left(\frac{e^{-f}}{1+e^{-f}}\right)$$

$$h'(f) = \left(1 - 1 + \frac{2}{1+e^{-f}}\right)\left(\frac{-1+1+e^{-f}}{1+e^{-f}}\right)$$

$$h'(f) = \left[1 + \left(-1 + \frac{2}{1+e^{-f}}\right)\right]\left(\frac{1+e^{-f}}{1+e^{-f}} + \frac{-1}{1+e^{-f}}\right)$$

$$h'(f) = (1+y)\left(1 - \frac{1}{1+e^{-f}}\right)$$

$$h'(f) = \frac{1}{2}(1+y)\left(2 - \frac{2}{1+e^{-f}}\right)$$

$$h'(f) = \frac{1}{2}(1+y)\left(1 + 1 - \frac{2}{1+e^{-f}}\right)$$

$$h'(f) = \frac{1}{2}(1+y)\left[1 - \left(-1 + \frac{2}{1+e^{-f}}\right)\right]$$

$$h'(f) = \frac{1}{2}(1+y)(1-y)$$

$$h'(f) = \frac{1-y^2}{2}$$

$$h'(f) = \frac{1-h^2(f)}{2}$$

The cost function

$$E = \frac{1}{2}(r - z^{out})^2$$

The output

$$z^{in} = \sum_i y_i^{out} w_i$$

The weight from the hidden layer to the output layer

$$w_{qp}^{k+1} = w_{qp}^{k} - \Delta w_{qp}$$

$$\Delta w_{qp} = -\alpha \frac{\partial E}{\partial z^{out}} \frac{\partial z^{out}}{\partial z^{in}} \frac{\partial z^{in}}{\partial w_{qp}}$$

$$\Delta w_{qp} = -\alpha \cdot -(r - z^{out}) \cdot h'\left(z^{in}\right) \cdot y_p^{out}$$

$$\Delta w_{qp} = \alpha(r - z^{out})h'\left(z^{in}\right)y_p^{out}$$

Approximate Ep by E

The hidden node

$$y^{in} = \sum_j x_j^{out} w_j$$

The weight from the input layer to the hidden layer

$$w_{pj}^{k+1} = w_{pj}^{k} - \Delta w_{pj}$$

$$\Delta w_{pj} = -\alpha \frac{\partial E}{\partial w_{pj}}$$

$$\Delta w_{pj} = -\alpha \frac{\partial E}{\partial y^{out}} \frac{\partial y^{out}}{\partial y^{in}} \frac{\partial y^{in}}{\partial w_{pj}}$$

$$\Delta w_{pj} = -\alpha \frac{\partial E}{\partial z^{out}} \frac{\partial z^{out}}{\partial z^{in}} \frac{\partial z^{in}}{\partial y^{out}} \frac{\partial y^{out}}{\partial y^{in}} \frac{\partial y^{in}}{\partial w_{pj}}$$

$$\Delta w_{qp} = -\alpha \cdot -(r - z^{out}) \cdot h'\left(z^{in}\right) \cdot w_{qp} \cdot h'\left(y^{in}\right) \cdot x_j$$

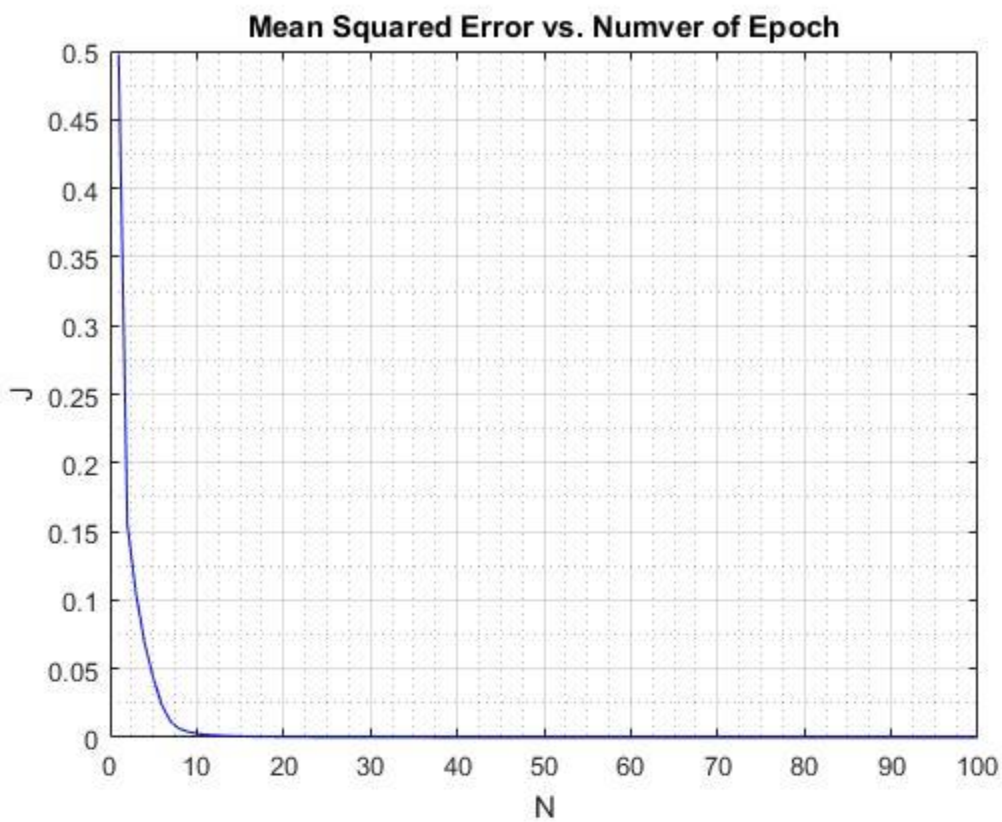
$$\Delta w_{qp} = \alpha(r - z^{out}) \cdot h'\left(z^{in}\right) w_{qp} h'\left(y^{in}\right) x_j$$

1-2) Solution: NN_training.m

For this neural network training, there are 3 layers, an input layer, one hidden layer, and an output layer. There are 49 inputs representing each pixel of the training images. There are 49 nodes in the hidden layer. There are 4 outputs representing each letter 'M', 'E', '1', and '7'.

The linear activation function h(f) = f is used for the input layer. The sigmoid function is used for the hidden layer and the output layer.

The cost function J = $\frac{1}{16}\sum(r - z_{out})^2$ (mean squared error) and is shown in the figure below.



To train the data set, both forward and backward propagation were performed.

The training model was run for 100 epoch, and the cost function is decreasing for each every epoch.

The resulted weights of each node were saved in the file NN_weights.mat.

1-3) Solution: NN_test.m

To test the data, only the forward propagation was performed. The resulted before rounding are shown in the table below.

	Output vector
M	[0.9967 0.0014 0.0008 0.0019]
E	[0.0047 0.9913 0.0009 0.0015]
1	[0.0004 0.0006 0.9988 0.0004]
7	[0.0092 0.0030 0.0004 0.9863]

After rounding,

	Output vector
M	[1 0 0 0]
E	[0 1 0 0]
1	[0 0 1 0]
7	[0 0 0 1]

Problem 2: Pose Estimation and Stereo Vision

2a) Camera Model Solution: CameraModel.m

From the translation, the location of both cameras is Z=8; hence, the depth of each pixel can be calculated from $|Z-Z_w|$

Since the cameras was rotated by R_x, translated by T, and rotated again by R_y, the feature points in the camera coordinate can be obtained from

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R_y]\left([R_x]\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + [T]\right)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R_y][R_x]\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + [R_y][T]$$

where

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\emptyset) & -sin(\emptyset) \\ 0 & sin(\emptyset) & cos(\emptyset) \end{bmatrix}$$

$$[R_y] = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix}$$

$$[T] = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

For the camera I, $\phi = 180^o$, $\Theta = -40^o$, $T = [2\ 1\ 8]^T$, and f = 1.3

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0.7660 & 0 & 0.6428 \\ 0 & -1 & 0 \\ 0.6428 & 0 & -0.7660 \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} -3.6102 \\ 1 \\ 7.4139 \end{bmatrix}$$

For the camera II, $\phi = 180^o$, $\Theta = 40^o$, $T = [-2\ 1\ 8]^T$, and f = 1.3.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0.7660 & 0 & -0.6428 \\ 0 & -1 & 0 \\ -0.6428 & 0 & -0.7660 \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} 3.6102 \\ 1 \\ 7.4139 \end{bmatrix}$$

Then, the feature points in the image plane can be obtained from

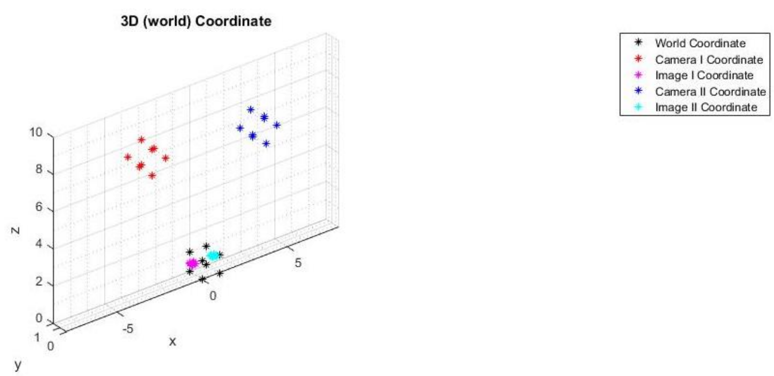
$$u = f\frac{x}{z}$$

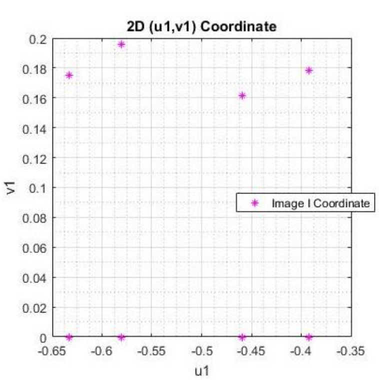
$$v = f\frac{y}{z}$$

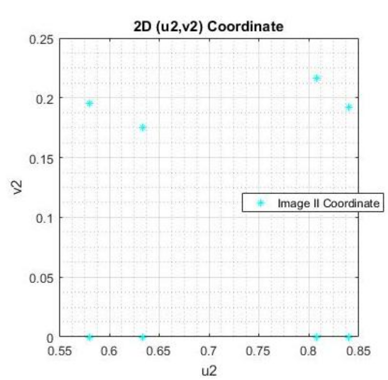
The results for each feature points in each coordinate are shown in the table below.

Feature Points								
X_w	0	1	1	0	0	1	1	0
Y_w	0	0	1	1	0	0	1	1
Z_w	0	0	0	0	1	1	1	1
Depth	8	8	8	8	7	7	7	7
x_1	-3.6102	-2.8441	-2.8441	-3.6102	-2.9674	-2.2013	-2.2013	-2.9674
y_1	1	1	0	0	1	1	0	0
z_1	7.4139	8.05674	8.0567	7.4139	6.6478	7.2906	7.2906	6.6478
x_2	3.6102	4.3762	4.3762	3.6102	2.9674	3.7334	3.7334	2.9674
y_2	1	1	0	0	1	1	0	0
z_2	7.4139	6.7711	6.7711	7.4139	6.6478	6.0050	6.0050	6.6478
u_1	-0.6330	-0.4589	-0.4589	-0.6330	-0.5802	-0.3925	-0.3925	-0.5802
v_1	0.1753	0.1613	0	0	0.1955	0.1783	0	0
u_2	0.6330	0.8402	0.8402	0.6330	0.5802	0.8082	0.8082	0.5802
v_2	0.1753	0.1919	0	0	0.1955	0.2164	0	0

The plots are shown here







2b) <u>Pose Estimation Solution:</u> PoseEstimation.m

Calculate the rotation matrix (**R**) and translation matrix (**T**) from world coordinate to Camera I

Construct the M^{16x12} matrix

$$M^{16x12} = \begin{bmatrix} fx_1 & fy_1 & fz_1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & f & 0 & -u_2 \\ 0 & 0 & 0 & fx_1 & fy_1 & fz_1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & 0 & f & -v_1 \\ & & & \vdots & & & & & \vdots & & & \\ fx_8 & fy_8 & fz_8 & 0 & 0 & 0 & -u_8x_8 & -u_8y_8 & -u_8z_8 & f & 0 & -u_8 \\ 0 & 0 & 0 & fx_8 & fy_8 & fz_8 & -v_8x_8 & -v_8y_8 & -v_8z_8 & 0 & f & -v_8 \end{bmatrix}$$

$$V^{12x1} = [r_{11} \quad r_{12} \quad r_{13} \quad r_{21} \quad r_{22} \quad r_{23} \quad r_{31} \quad r_{32} \quad r_{33} \quad t_x \quad t_y \quad t_z]^T$$

Then, solve the system of equation MV = 0 using the reduced row echelon form of the augmented matrix [M | 0]

rref([M|0])												
1	0	0	0	0	0	0	0	0	0	0	-0.1033	0
0	1	0	0	0	0	0	0	0	0	0	5.5511e-17	0
0	0	1	0	0	0	0	0	0	0	0	-0.0866	0
0	0	0	1	0	0	0	0	0	0	0	1.4974e-17	0
0	0	0	0	1	0	0	0	0	0	0	0.1348	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	-0.0866	0
0	0	0	0	0	0	0	1	0	0	0	-4.4408e-16	0
0	0	0	0	0	0	0	0	1	0	0	0.1033	0
0	0	0	0	0	0	0	0	0	1	0	0.4869	0
0	0	0	0	0	0	0	0	0	0	1	-0.1348	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Therefore, the elements of [**R**], and [**T**] are in the terms of t_z as

$$r_{11} = 0.1033t_z$$

$$r_{12} = 0$$

$$r_{13} = 0.0866t_z$$

$$r_{21} = 0$$

$$r_{22} = -0.1348t_z$$

$$r_{23} = 0$$

$$r_{31} = -0.1033t_z$$

$$r_{32} = 0$$

$$r_{33} = 0.1033t_z$$

$$t_x = -0.4869t_z$$

$$t_y = 0.1348t_z$$

t_z can be uniquely determined from the normality condition

$$r_{11}^2 + r_{12}^2 + r_{13}^2 = 1$$

$$(0.1033t_z)^2 + (0)^2 + (-0.1348t_z)^2 = 1$$

$$t_z = 7.4139$$

Substitute t_z back in, then obtained [**R**], and [**T**], which are the same as in problem 2a)

$$[R] = \begin{bmatrix} 0.7660 & 0 & 0.6428 \\ 0 & -1 & 0 \\ 0.6428 & 0 & -0.7660 \end{bmatrix}$$

$$[T] = \begin{bmatrix} -3.6102 \\ 1 \\ 7.4139 \end{bmatrix}$$

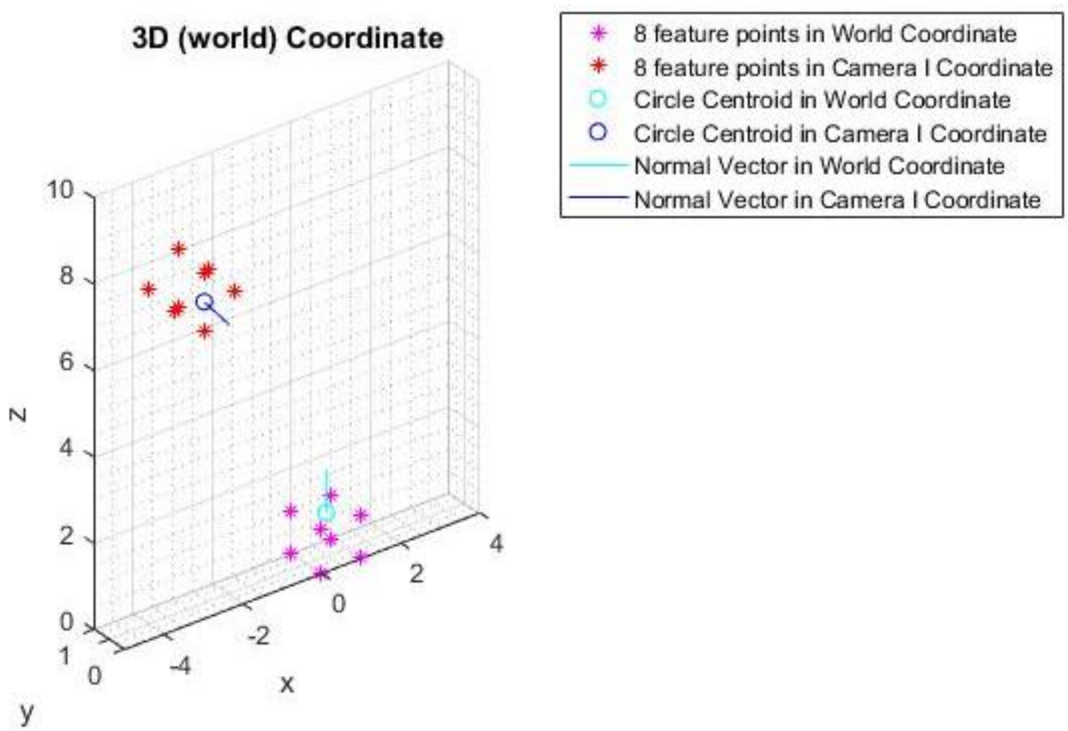
1) These 8 feature points and the centroid and normal vector of the circle in the camera I coordinate can be determined from the rotation and translation.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R]\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + [T],\ [O_c] = [R][O_o] + [T],\ [N_c] = [R][N_o]$$

The results are shown in the following table.

Feature Points								
X_w	0	1	1	0	0	1	1	0
Y_w	0	0	1	1	0	0	1	1
Z_w	0	0	0	0	1	1	1	1
x	-3.6102	-2.8441	-2.8441	-3.6102	-2.9674	-2.2013	-2.2013	-2.9674
y	1	1	0	0	1	1	0	0
z	7.4139	8.05674	8.0567	7.4139	6.6478	7.2906	7.2906	6.6478
O_{ox}	0.5	O_{cx}	-2.5844		N_{ox}	0.5	N_{cx}	0.6428
O_{oy}	0.5	O_{cy}	0.5		N_{oy}	0.5	N_{cy}	0
O_{oz}	1	O_{cz}	6.9692		N_{oz}	1	N_{cz}	-0.766

The plot is shown below.



2) Form the matrix **W** = [**M Q Q'**]^T and apply pseudo-inverse on **WV** = **b** to construct [**R**] and [**T**]

M and **V** are already defined above as

$$M^{16x12} = \begin{bmatrix} fx_1 & fy_1 & fz_1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & f & 0 & -u_2 \\ 0 & 0 & 0 & fx_1 & fy_1 & fz_1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & 0 & f & -v_1 \\ & & \vdots & & & & & & \vdots & & & \\ fx_8 & fy_8 & fz_8 & 0 & 0 & 0 & -u_8x_8 & -u_8y_8 & -u_8z_8 & f & 0 & -u_8 \\ 0 & 0 & 0 & fx_8 & fy_8 & fz_8 & -v_8x_8 & -v_8y_8 & -v_8z_8 & 0 & f & -v_8 \end{bmatrix}$$

$$V^{12x1} = [r_{11} \quad r_{12} \quad r_{13} \quad r_{21} \quad r_{22} \quad r_{23} \quad r_{31} \quad r_{32} \quad r_{33} \quad t_x \quad t_y \quad t_z]^T$$

The rest can be defined as following

$$Q^{6x12} = \begin{bmatrix} N_{ox} & N_{oy} & N_{oz} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & N_{ox} & N_{oy} & N_{oz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & N_{ox} & N_{oy} & N_{oz} & 0 & 0 & 0 \\ O_{ox} & O_{oy} & O_{oz} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & O_{ox} & O_{oy} & O_{oz} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & O_{ox} & O_{oy} & O_{oz} & 0 & 0 & 1 \end{bmatrix}$$

$$Q'^{3x12} = \begin{bmatrix} N_{cx} & 0 & 0 & N_{cx} & 0 & 0 & N_{cx} & 0 & 0 & 0 & 0 & 0 \\ 0 & N_{cy} & 0 & 0 & N_{cy} & 0 & 0 & N_{cy} & 0 & 0 & 0 & 0 \\ 0 & 0 & N_{cz} & 0 & 0 & N_{cz} & 0 & 0 & N_{cz} & 0 & 0 & 0 \end{bmatrix}$$

$$W^{25x12} = \begin{bmatrix} M \\ Q \\ Q' \end{bmatrix}$$

$$b^{25x1} = \begin{bmatrix} 0_{16x1} \\ N_{cx} \\ N_{cy} \\ N_{cz} \\ O_{cx} \\ O_{cy} \\ O_{cz} \\ N_{ox} \\ N_{oy} \\ N_{oz} \end{bmatrix}$$

Use pseudo-inverse method to obtained [**R**], and [**T**], which are the same as in problem 2a)

$$[R] = \begin{bmatrix} 0.7660 & 0 & 0.6428 \\ 0 & -1 & 0 \\ 0.6428 & 0 & -0.7660 \end{bmatrix}$$

$$[T] = \begin{bmatrix} -3.6102 \\ 1 \\ 7.4139 \end{bmatrix}$$

2c) Stereo Vision (Parallel) Solution: StereoVisionParallel.m

Since the cameras was rotated by R_x, and translated by T, the feature points in the camera coordinate can be obtained from

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R_x] \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + [T]$$

where

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\emptyset) & -sin(\emptyset) \\ 0 & sin(\emptyset) & cos(\emptyset) \end{bmatrix}$$

$$[T] = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

For the camera I, $\phi = 180^o$, $T = [2\ 1\ 8]^T$, and f = 1.3

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ 8 \end{bmatrix}$$

For the camera II, $\phi = 180^o$, $T = [-2\ 1\ 8]^T$, and f = 1.3.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} -2 \\ 1 \\ 8 \end{bmatrix}$$

Note that the focal length f of two cameras are equal to each other.

Then, the feature points in the image plane can be obtained from

$$u = f\frac{x}{z}$$

$$v = f\frac{y}{z}$$

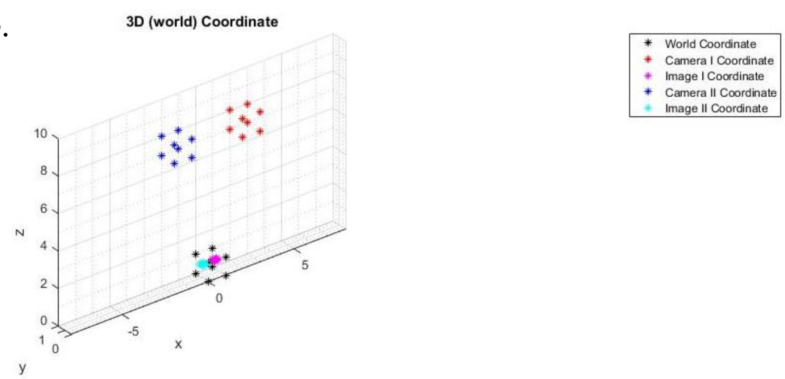
The distance between these two cameras is b = 4, and the depth of each point can be calculated from

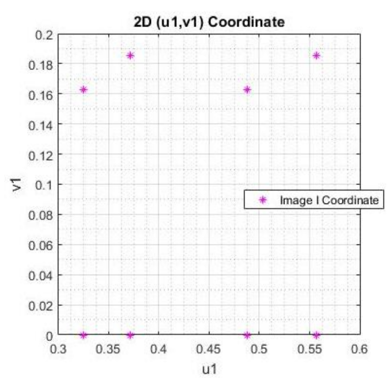
$$Z = \frac{bf}{u_1 - u_2}$$

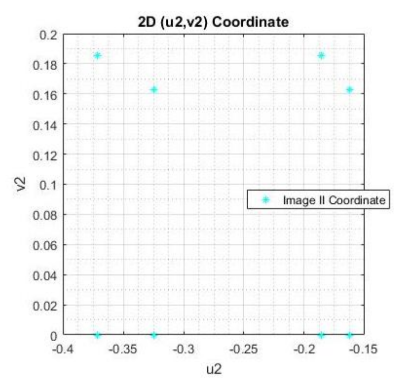
The results for each feature points in each coordinate are shown in the table below.

Feature Points								
X_w	0	1	1	0	0	1	1	0
Y_w	0	0	1	1	0	0	1	1
Z_w	0	0	0	0	1	1	1	1
Depth	8	8	8	8	7	7	7	7
x_1	2	3	3	2	2	3	3	2
y_1	1	1	0	0	1	1	0	0
z_1	8	8	8	8	7	7	7	7
x_2	-2	-1	-1	-2	-2	-1	-1	-2
y_2	1	1	0	0	1	1	0	0
z_2	8	8	8	8	7	7	7	7
u_1	0.3250	0.4875	0.4875	0.3250	0.3714	0.5571	0.5571	0.3714
v_1	0.1625	0.1625	0	0	0.1857	0.1857	0	0
u_2	-0.3250	-0.1625	-0.1625	-0.3250	-0.3714	-0.1857	-0.1857	-0.3714
v_2	0.1625	0.1625	0	0	0.1857	0.1857	0	0

The plots are shown here.







2d) Stereo Vision (General) Solution: StereoVisionGeneral.m

Use information from 2a) (u1, v1, u2, v2, Rx1, Ry1, T1, Rx2, Ry2, T2, f1, f2) to reconstruct these 8 feature points in the world coordinate.

Each of these points can be reconstructed from the following equations

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} x \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{y1}R_{x1} & R_{y1}T_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} x \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{y2}R_{x2} & R_{y2}T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = 0$$

Rewritten the cross product in the skew-matrix form

$$\begin{bmatrix} 0 & -1 & v_1 \\ 1 & 0 & -u_1 \\ -v_1 & u_1 & 0 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{y1}R_{x1} & R_{y1}T_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & -1 & v_2 \\ 1 & 0 & -u_2 \\ -v_2 & u_2 & 0 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{y2}R_{x2} & R_{y2}T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = 0$$

The product of the first to the fourth matrices will give the 3x4 matrix, and concatenate these 3x4 matrices from each equation, obtained

$$[V_1 \quad V_2 \quad V_3 \quad V_4]_{6x4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = 0$$

where each V_i has a dimension of 6x1. Performing the matrix multiplication,

$$V_1X_w + V_2Y_w + V_3Z_w + V_4 = 0$$

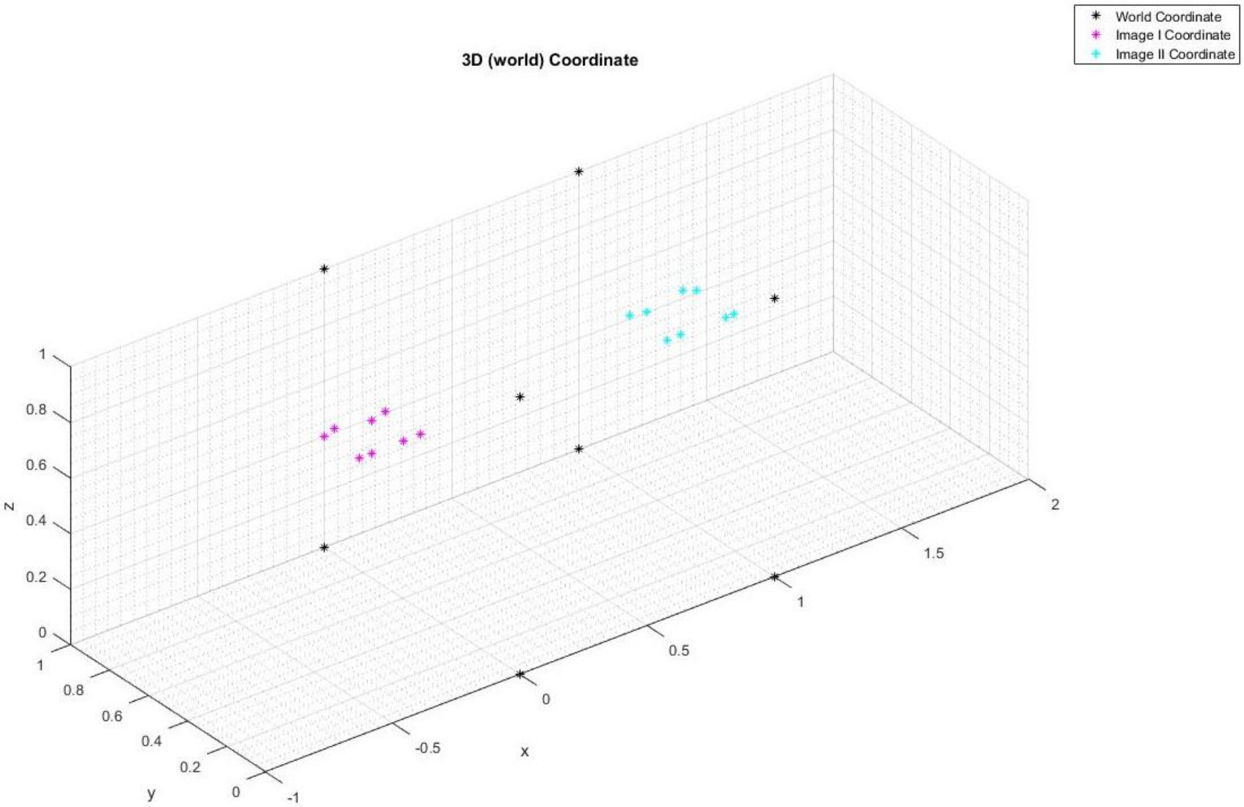
$$V_1X_w + V_2Y_w + V_3Z_w = -V_4$$

$$[V_1 \quad V_2 \quad V_3] \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = -V_4$$

Use the pseudo-inverse method, the object can be reconstructed and its result is shown in the following table

Feature Points								
X_w	0	1	1	0	0	1	1	0
Y_w	0	0	1	1	0	0	1	1
Z_w	0	0	0	0	1	1	1	1

The plot is shown below



Problem 3: Color

3a) Artificial Color Contrast (ACC) Solution: HW4_3a.m

The filtered image can be calculated from the following equations

$$h(x,y) = DoG * f_j(x,y) + G_{\sigma s} * (f_j(x,y) - f_k(x,y))$$

$$h(x,y) = (G_{\sigma c} - G_{\sigma s}) * R + G_{\sigma s} * (R - (-(R - G)))$$

$$h(x,y) = (G_{\sigma c} - G_{\sigma s}) * R + G_{\sigma s} * (R - (-R + G))$$

$$h(x,y) = (G_{\sigma c} - G_{\sigma s}) * R + G_{\sigma s} * (R + R - G)$$

$$h(x,y) = (G_{\sigma c} - G_{\sigma s}) * R + G_{\sigma s} * (2R - G)$$

Applying the distributivity properties of the convolution integral

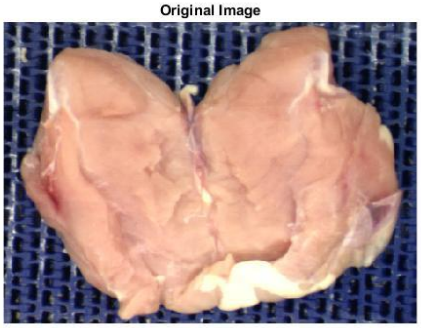
$$h(x,y) = G_{\sigma c} * R - G_{\sigma s} * R + G_{\sigma s} * 2R - G_{\sigma s} * G)$$

$$h(x,y) = G_{\sigma c} * R + G_{\sigma s} * R - G_{\sigma s} * G)$$

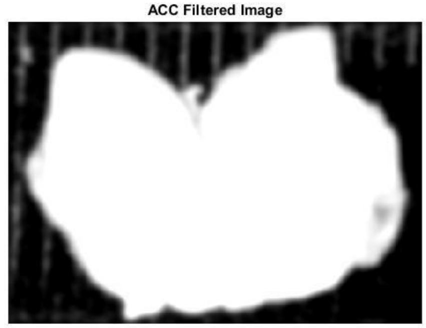
$$h(x,y) = G_{\sigma c} * R + G_{\sigma s} * (R - G)$$

The results are shown in the following figures

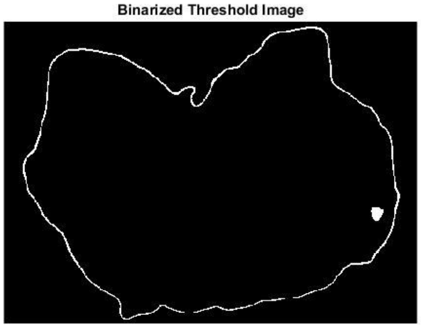
Original Image



ACC Filtered Image



Binarized Threshold Image



Binarized Threshold Image - Removed Boundary

3b) Principle Component Analysis (PCA) Solution: HW4_3b.m

The procedures were followed the example in class slides.

1) The covariance matrix of data is

$$C = \begin{bmatrix} 8099 & 5596.5 & 3166.9 \\ 5596.5 & 4123.1 & 2497.3 \\ 3166.9 & 2497.3 & 1756.4 \end{bmatrix}$$

2) The eigenvalues in descending order are

$$\lambda_1 = 13439.91$$

$$\lambda_2 = 494.3529$$

$$\lambda_3 = 45.2505$$

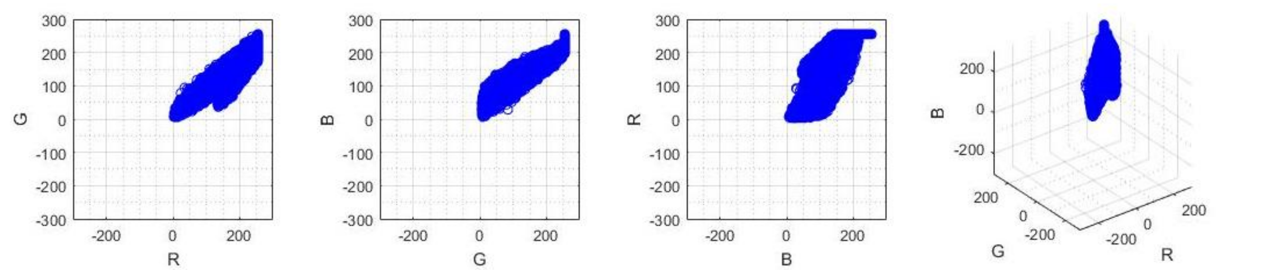
The eigenvectors correspond to each eigenvectors are

$$v_1 = \begin{bmatrix} 0.7693 \\ 0.5495 \\ 0.3260 \end{bmatrix}$$

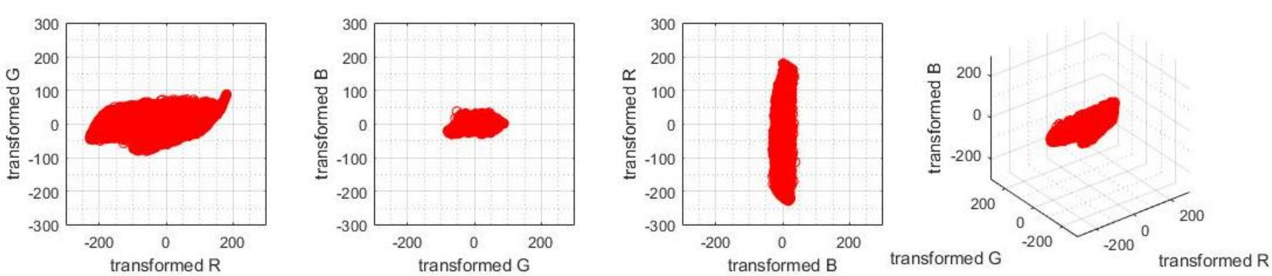
$$v_2 = \begin{bmatrix} -0.5338 \\ 0.2724 \\ 0.8005 \end{bmatrix}$$

$$v_3 = \begin{bmatrix} -0.3511 \\ 0.7898 \\ -0.5029 \end{bmatrix}$$

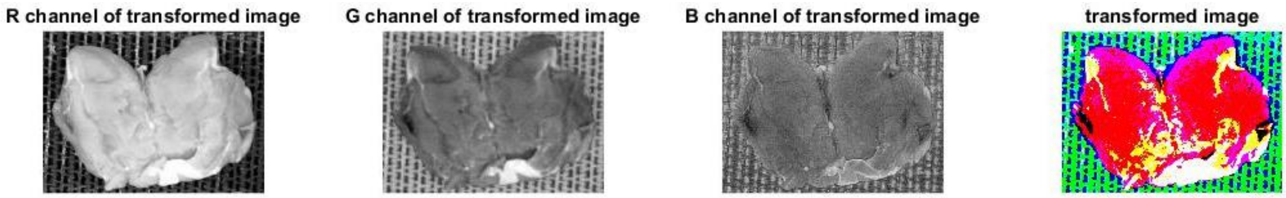
3) The following plots show the mapped R, G, B data on these three component axes



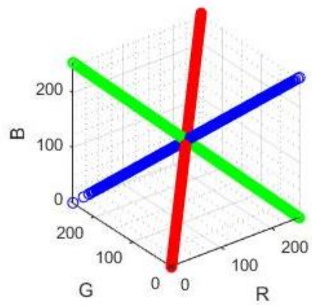
The following plots show the projected data along three component axes

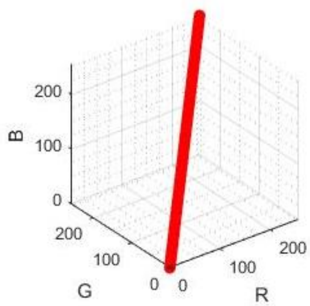


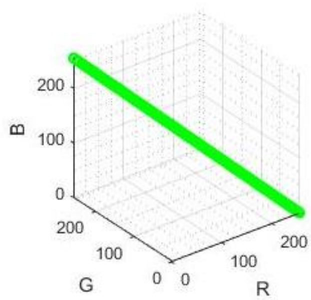
The following figures show the three single channel component image

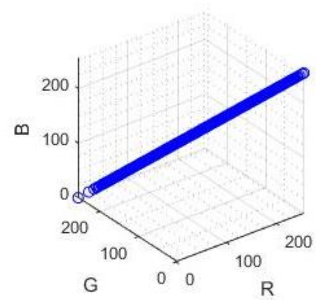


The following plots show the projected data by RGB coordinate

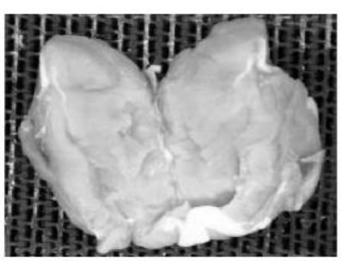


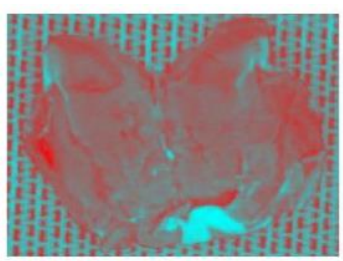


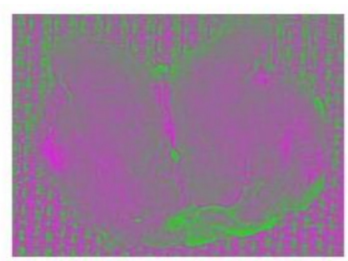




The following figures show the three color component images







<u>Compare and discuss these images</u>

It can be seen from figures above that transformed R component has more contrast than any of the origin bands because its gray level range is the largest.

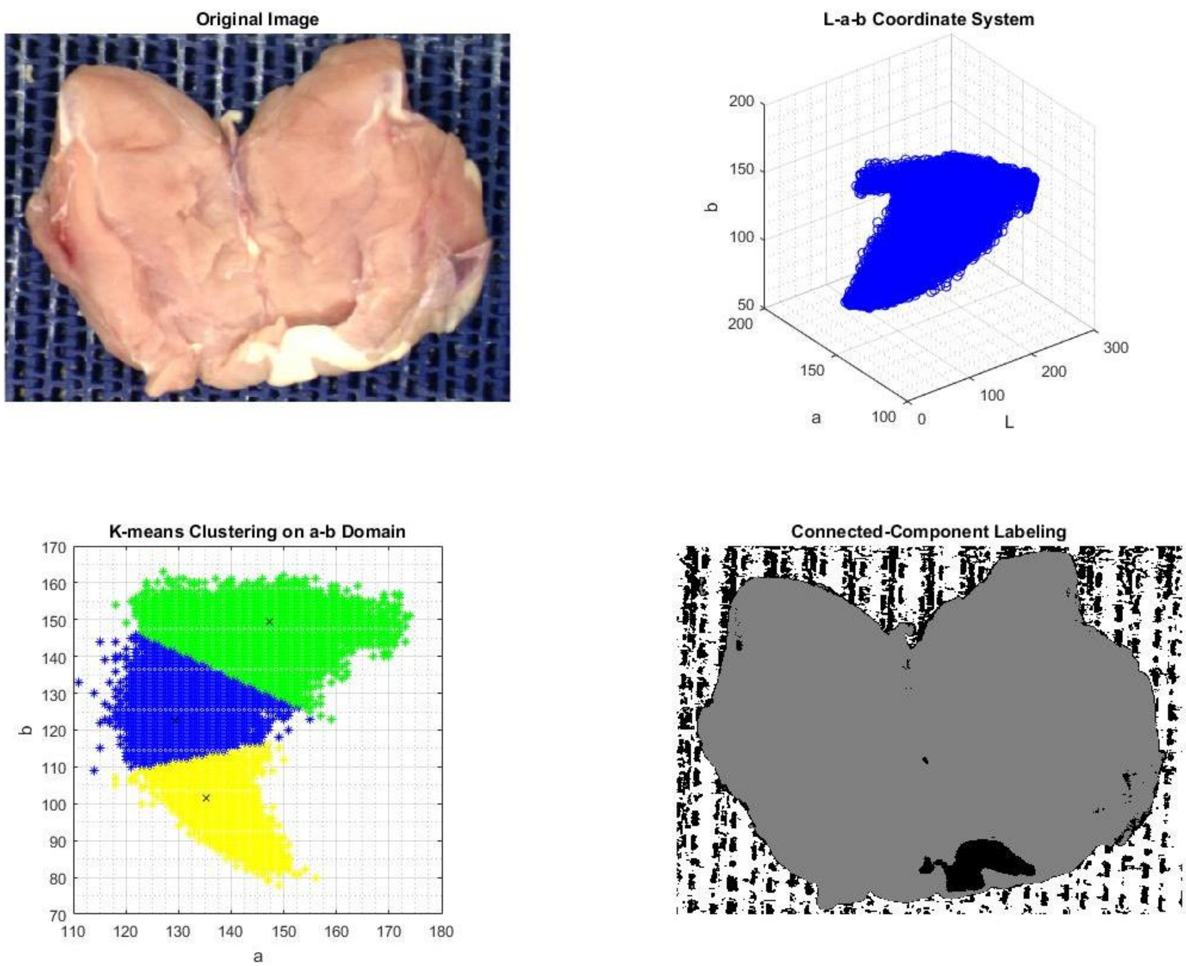
3c) Color-Based Image Segmentation Solution: HW4_3c.m

The given RGB picture was converted to Lab color system by MATLAB built-in function "`applycform()`" and each pixel was plotted on Lab coordinate.

Then, the K-mean clustering algorithm was applied to separate the data in a-b domain into 3 cluster (k = 3) for segmentation.

After that, each pixel was converted back to the image in order to visualize each cluster.

The results are shown in the following figure.



Note that the centroid of each cluster was randomly generated; therefore, the color assigned to each cluster may be different for each time the program is executed.