Homework 3

Meta information	
Name	Savan Kiran
Program	Masters in Computer Science
Questions skipped	N/A
Questions substituted	N/A
Extra credit questions	N/A

Part A1

We parse the provided data and draw a scatter plot to visualize the data points. By linear regression we now try to fit a line through the data points. We employ two ways to it, a) Homogeneous Linear Least Squares (shown by red line in Figure 1) and b) Non-homogeneous Linear Least Squares (shown by black line in Figure 1). (Note, LLS and Linear least squares will be used interchangeably)

Table 1 shows the slope, intercept and RMS error of the fit under both models.

	Slope	Intercept	Y-RMS error	D-RMS error
Non-homogeneous Linear Least Squares	-0.47724	1.9769	0.12503	0.11284
Homogeneous Linear Least Squares	-0.48404	1.9868	0.1252	0.11269

Below is the code snippet to display scatter plot of data points.

```
%A1
data=importdata('line_data.txt');
figure('Name','A1 line fitting','NumberTitle','off');
scatter(data(:,1),data(:,2),'b');
hold on
```

Below is the code snippet to fit line using Non-homogeneous linear least squares.

```
%Non-homogeneous Linear Least Squares
[nRows,nCols]=size(data);
U=[];
for r=1:nRows
    U=[U; data(r,1), 1];
end
A=pinv(U)*data(:,2);
m=A(1,1);
c=A(2,1);
lineX=[min(data(:,1)),max(data(:,1))];
lineY=m.*lineX+c;
plot(lineX,lineY,'k');
```

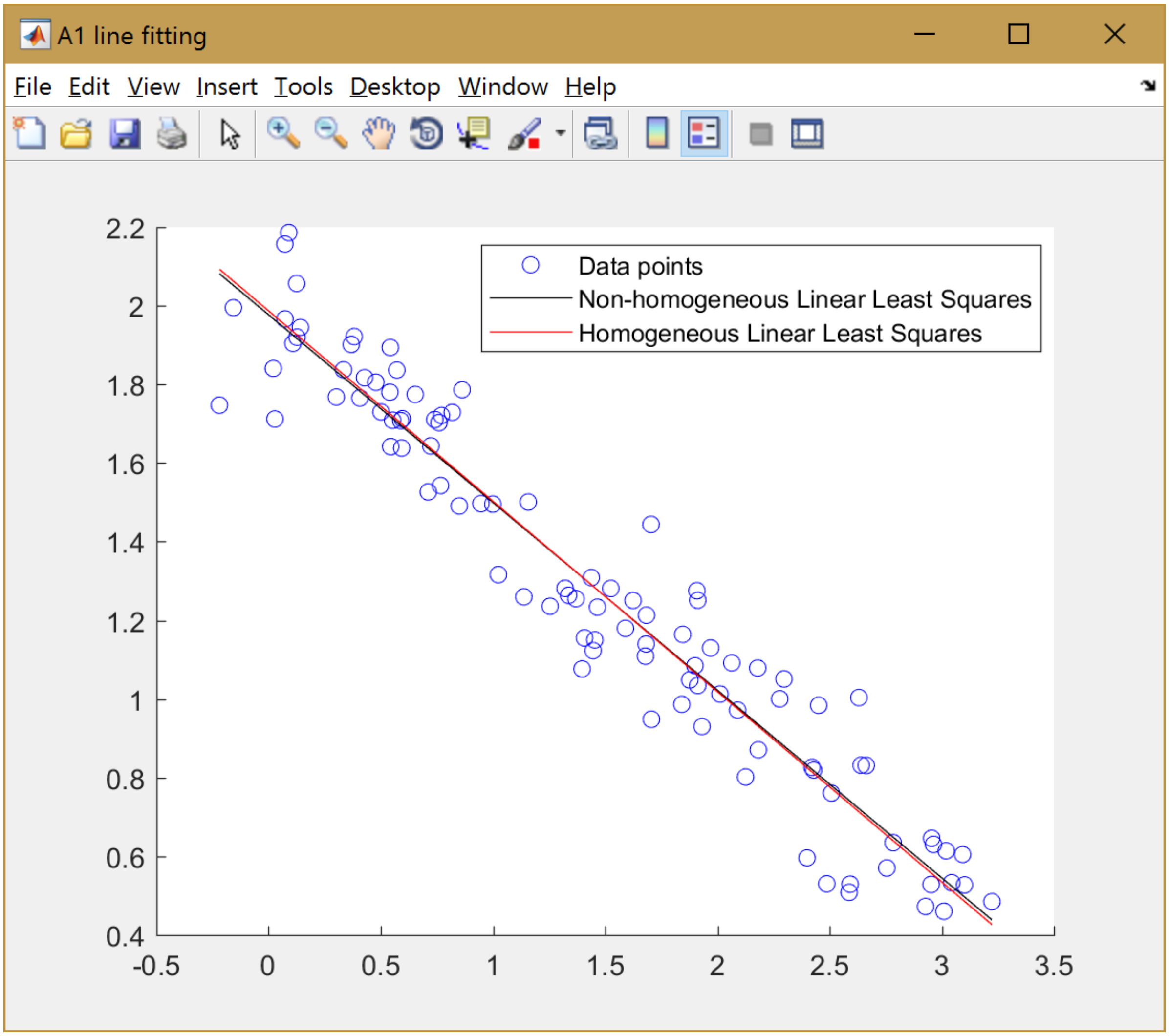

Figure 1. Scatter plot of data points and lines fitted using homogeneous and non-homogenous linear least squares

Below is the code snippet to fit line using Homogeneous linear least squares.

```
%Homogeneous Linear Least Squares
xMean=mean(data(:,1));
yMean=mean(data(:,2));
Ux=data(:,1)-xMean;
Uy=data(:,2)-yMean;
U2=[Ux Uy];
[eV,eD]=eig(U2'*U2);
a=eV(1,1);
b=eV(2,1);
d=a*xMean+b*yMean;
lineX=[min(data(:,1)),max(data(:,1))];
lineY=(-a/b).*lineX+(d/b);
plot(lineX,lineY,'r');
```

Below is the code snippet to calculate RMS error.

```
function [rms_error] = rms_error(A_line,A_points)
A_diff=(A_line-A_points).^2;
rms_error=sqrt(mean(A_diff));
end
```

Homogeneous linear least squares RMS error:

```
dLineVals=ones(nRows,1).*d;
dPointsVals=a*data(:,1)+b*data(:,2);
[rmsError_d]=rms_error(dLineVals,dPointsVals);
xLineVals=data(:,1);
m=-a/b;
c=d/b;
yLineVals=m*xLineVals+c;
[rmsError_y]=rms_error(yLineVals,data(:,2));

disp('Homogeneous Linear Least Squares -');
display((-a/b),(d/b),rmsError_y,rmsError_d);
```

Non-homogeneous linear least squares RMS error:

```
xLineVals=data(:,1);
yLineVals=m*xLineVals+c;
[rmsError_y]=rms_error(yLineVals,data(:,2));
q=sqrt(m^2+1);
a=-m/q;
b=1/q;
d=c/q;
dLineVals=ones(nRows,1).*d;
dPointsVals=a*data(:,1)+b*data(:,2);
[rmsError_d]=rms_error(dLineVals,dPointsVals);

disp('Non-homogeneous Linear Least Squares -');
display(m,c,rmsError_y,rmsError_d);
```

Comments on RMS errors:

As expected, non-homogeneous LLS has smaller 'Y' RMS error whereas homogeneous LLS has smaller 'D' RMS error as they're expected to minimize their respective errors. Non-homogeneous LLS minimizes the 'Y' or vertical distance of the points to the line whereas homogeneous LLS minimizes the 'D' or normal distance (hence considering both x and y components) of the points to the line. The former focuses solely on 'Y' and hence cannot guarantee smallest 'D' error i.e., normal distance and the latter, similarly cannot guarantee smallest 'Y' error i.e., vertical distance.

(Note: Part A2 is at the end)

Part B1

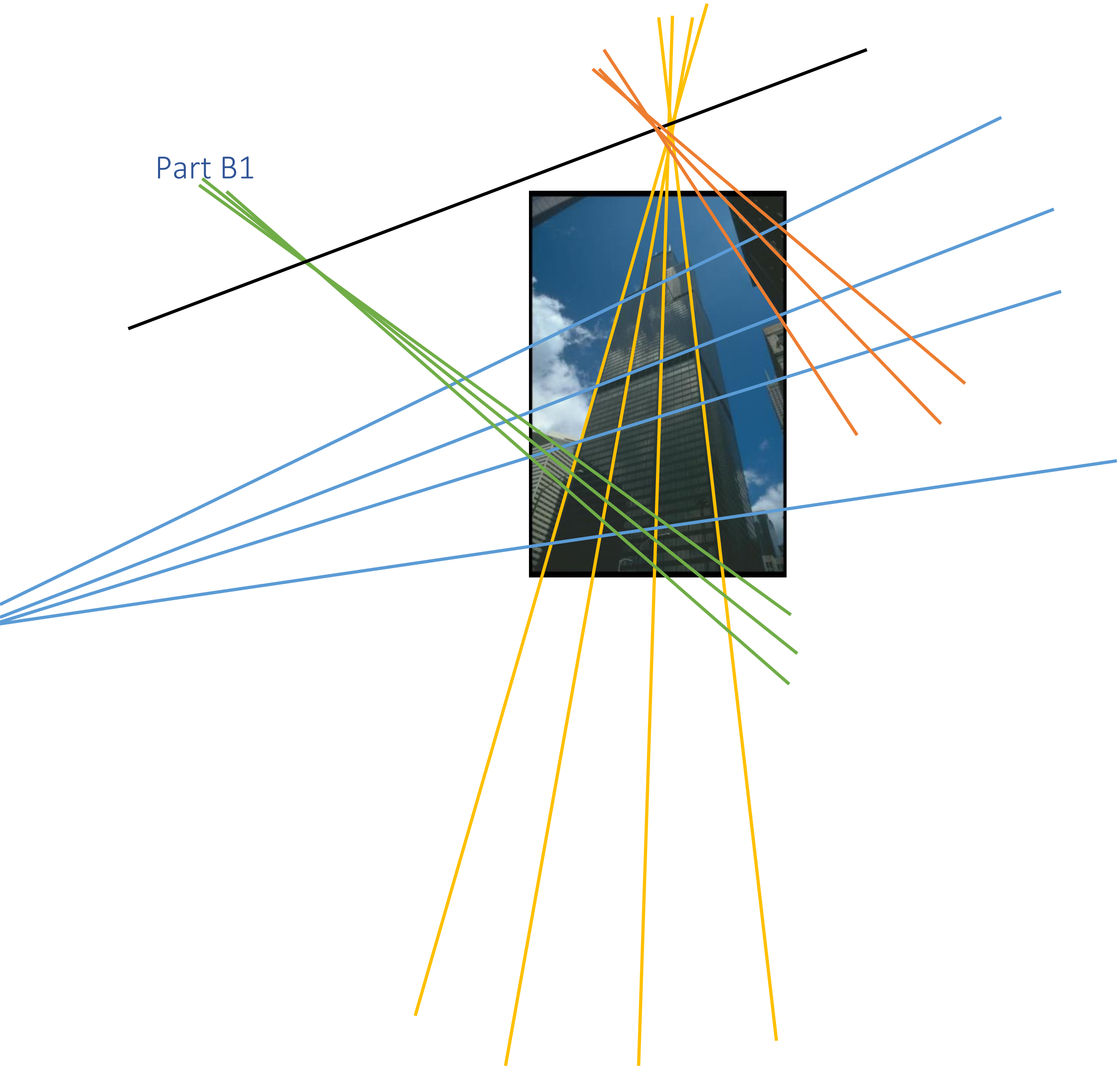

Figure 2. Image of a building with parallel lines, vanishing points and the horizon

The parallel lines drawn above do not exactly intersect at a point because of some properties (like, divergence) of the lens of the camera that was used to take the picture. We make an approximate judgement here. The horizon line also might not pass through all the vanishing points exactly and we make the same simplification as before.

1. Above is a set of parallel lines (each set encoded in a color – yellow, orange, blue and green) that meet at different vanishing points for each set.
2. The set of parallel lines indicated by orange, yellow and green are all on the same plane and their vanishing points are collinear. The line joining all 3 vanishing points forms one of the horizons for the image and is indicated in black in Figure 2.

The scale and perspective hold correct because of the satisfaction of above two conditions. Therefore, the image is real.

Part B2

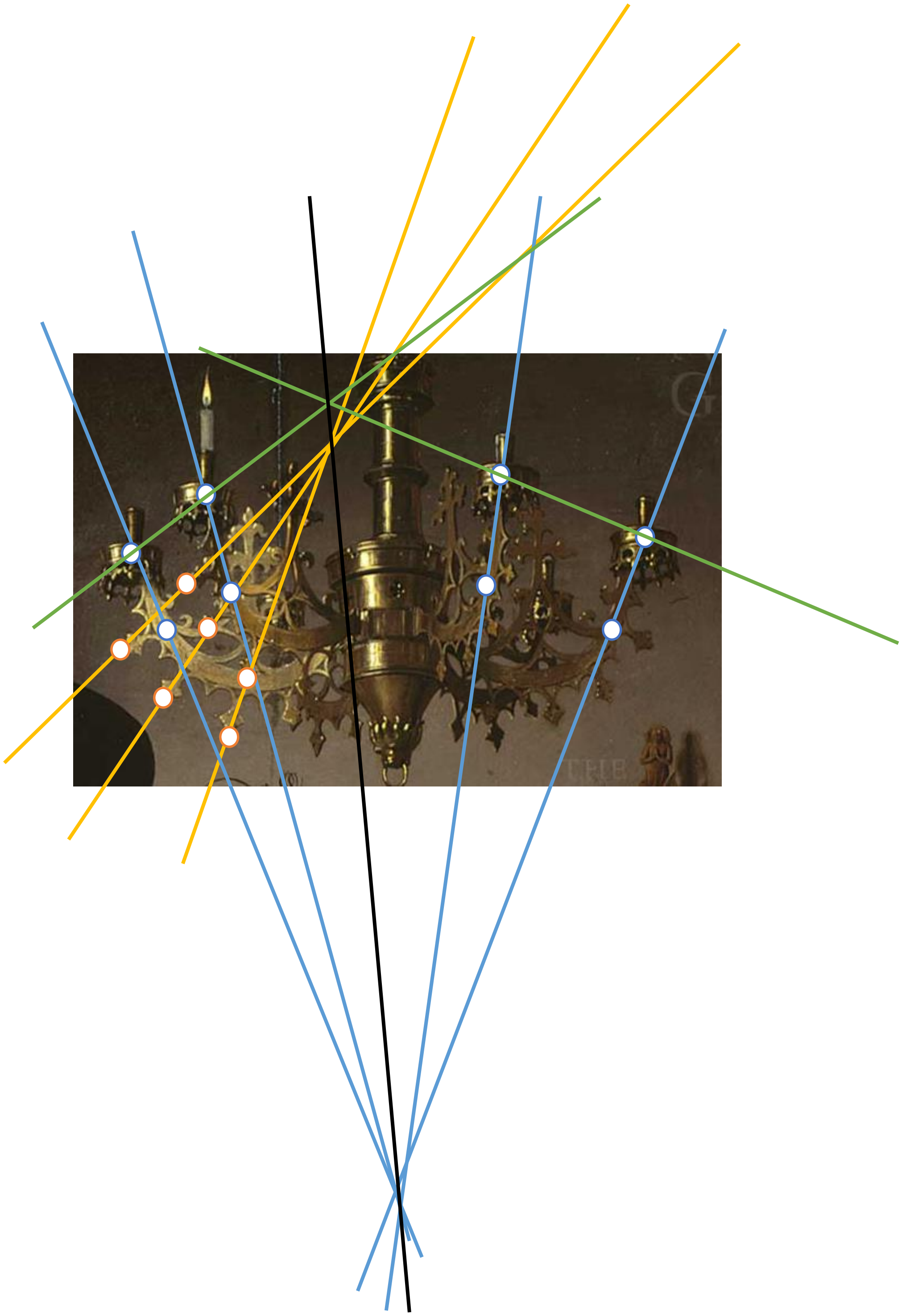

Figure 3. Image of a chandelier with parallel lines, vanishing points and the horizon

In Figure 3, we can see 3 sets of parallel lines marked in blue, green and yellow, all sets meeting at their vanishing points. The blue parallel lines connect the candle stand and arm below that is extended from the main tower, both of which are marked in blue circles. The yellow parallel lines connect the 3 artifacts under the arms adjacent to each other and on left side of the chandelier. The green parallel lines candle stands on either side of the chandelier's main tower.

The parallel lines drawn above do not exactly intersect at a point because of some properties (like, divergence) of the lens of the camera that was used to take the picture. We make an approximate judgement here. The horizon line also might not pass through all the vanishing points exactly and we make the same simplification as before.

1. Each set of parallel lines (yellow, blue and green) meets at different vanishing points.
2. The set of parallel lines indicated by blue, yellow and green are all on the same plane and their vanishing points are collinear. The line joining all 3 vanishing points forms one of the horizons for the image and is indicated in black in Figure 3.

The scale and perspective hold correct because of the satisfaction of above two conditions. Therefore, the image is real.

Part A2

What we found out was smallest 'Y' RMS error for non-homogeneous LLS and smallest 'D' RMS error for homogeneous LLS. This is in line to the expected result. As a corollary, non-homogeneous LLS can't guarantee smallest 'D' RMS error and homogeneous LLS can't guarantee smallest 'Y' RMS error.

Let's try to look at why that is the case.

For non-homogeneous LLS, the error is calculated as follows:

$$Non - homogeneous\ LLS\ error = \sum [y_i - (mx_i + b)]^2$$

We can observe that, Non-homogeneous LLS only considers the 'y' component for error calculation, therefor minimizing the global 'Y' error.

For homogeneous LLS, the error is calculated as follows:

$$Homogeneous\ LLS\ error = \sum (d - ax_i - by_i)^2$$

Here, we can observe that, Homogeneous LLS considers both 'x' and 'y' component for error calculation, and thus minimizing the (combined) normal distance of the points from the line. However, it cannot guarantee the global minimum for either of the component individually.

In the scenario where, both non-homogeneous and homogeneous LLS can only find local minimum for 'Y' RMS error and 'D' RMS error respectively, then the above argument doesn't hold.

If this is the case, it could so happen that, homogeneous LLS gives the smallest RMS error for both errors or vice-versa. It could also happen that, homogeneous LLS which was known to give global minimum for 'D' RMS error, might not anymore. Instead, non-homogeneous RMS error can give the smallest error for 'D' RMS error. The opposite case is also likely to happen.