HACETTEPE UNIVERSITY
DEPARTMENT OF
GEOMATICS ENGINEERING

GMT 333

PHOTOGRAMMETRY

Homework -1

ABDULSAMET TOPTAŞ – 21905024

1. Question

The logic of the first problem is to create grid squares to scan the image, connect the necessary information and scan the image to obtain the perspective coordinates. The coordinates of the upper left and lower right corners of our image are given. First I found the Z coordinate of the flight altitude. I used the formula "(focal length x gsd / pixel size) + average terrain height" to find this out. This formula gives us the elevation (Z) of the image relative to sea level. After adjusting the bindings using the upper left coordinates of the first frame, I found the center coordinates (Figure 1). Then, based on the corner coordinates of the first grid I found, I created the 2nd, 3rd and 4th grids by making sufficient overlaps. Next, I found the lower left coordinates of the image for 8.Grid. Using these coordinates, I calculated the corner coordinates and center coordinates of the 8th grid. At the same time, I did not forget to make the necessary overlays in the y coordinates.

Note : I made the necessary overlays correctly by shifting +-10 meters from the given corner coordinates.

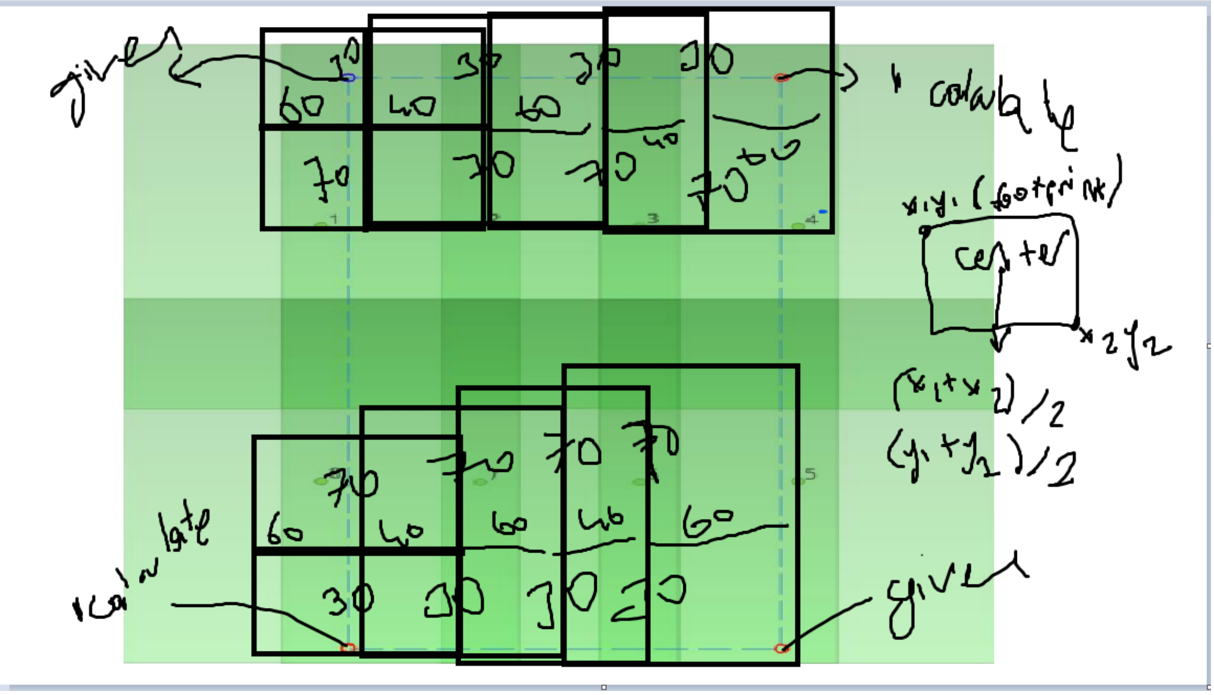
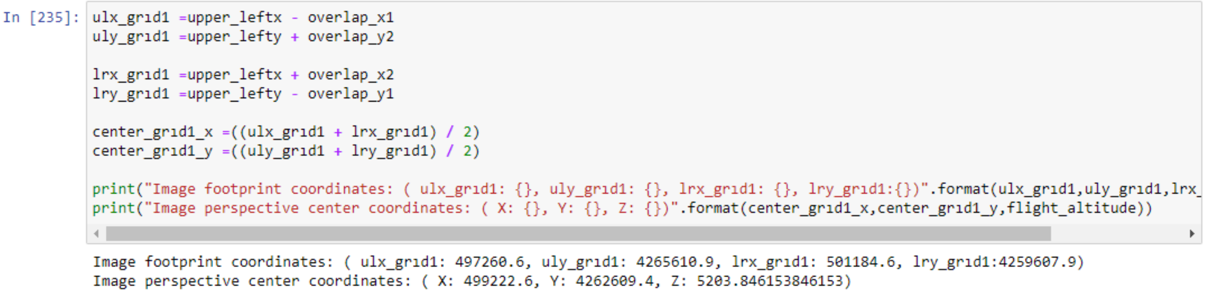


Figure 1

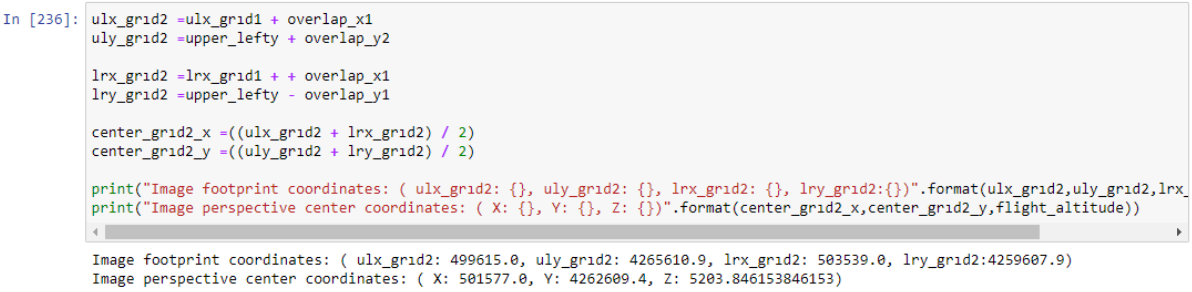
The logic in the image in Figure 2 gives the foot prints of the first grid we created by adding or subtracting the total amount of overlay on the coordinates given in the upper left of the image.



```
In [235]: ulx_grid1 =upper_leftx - overlap_x1
          uly_grid1 =upper_lefty + overlap_y2

          lrx_grid1 =upper_leftx + overlap_x2
          lry_grid1 =upper_lefty - overlap_y1

          center_grid1_x =((ulx_grid1 + lrx_grid1) / 2)
          center_grid1_y =((uly_grid1 + lry_grid1) / 2)

          print("Image footprint coordinates: ( ulx_grid1: {}, uly_grid1: {}, lrx_grid1: {}, lry_grid1:{})".format(ulx_grid1,uly_grid1,lrx_
          print("Image perspective center coordinates: ( X: {}, Y: {}, Z: {})".format(center_grid1_x,center_grid1_y,flight_altitude))

          Image footprint coordinates: ( ulx_grid1: 497260.6, uly_grid1: 4265610.9, lrx_grid1: 501184.6, lry_grid1:4259607.9)
          Image perspective center coordinates: ( X: 499222.6, Y: 4262609.4, Z: 5203.846153846153)
```

Figure 2

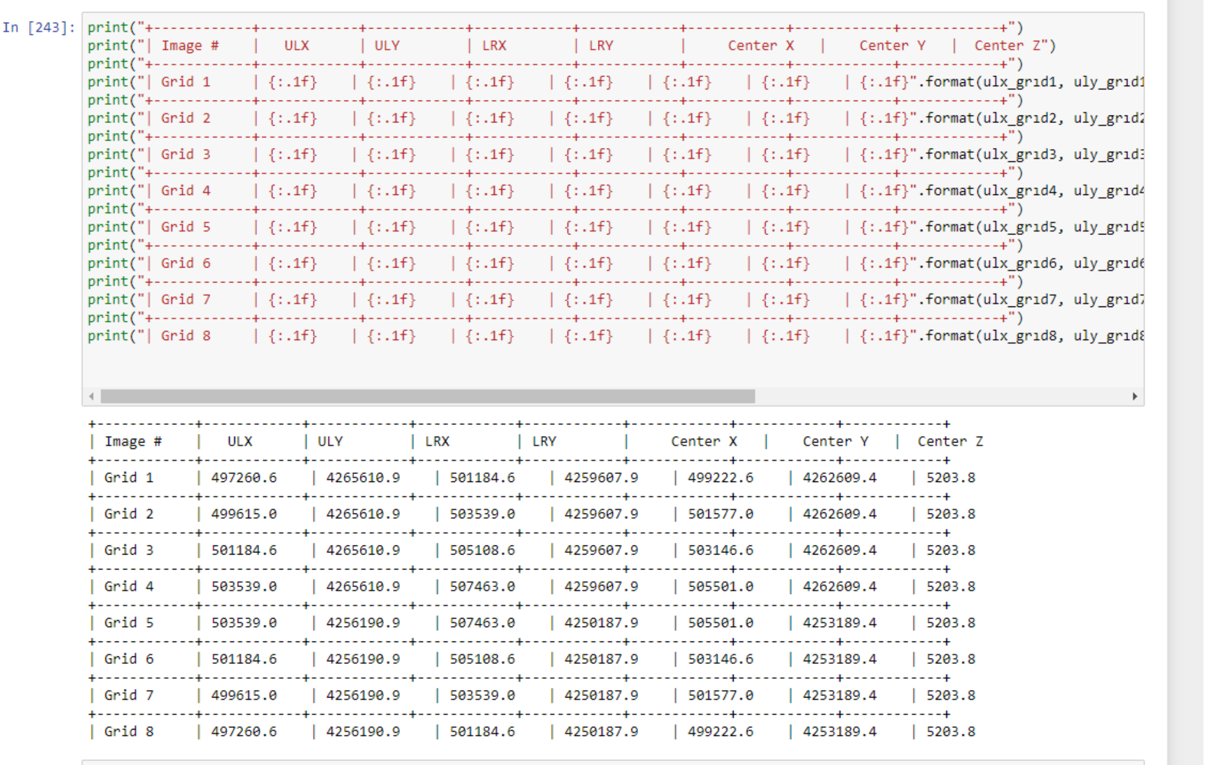
The logic in the image in Figure 3 can be a reference to all other images. The second grid foot prints are calculated by overlaying the corner coordinates of the first grid we found.



```
In [236]: ulx_grid2 =ulx_grid1 + overlap_x1
          uly_grid2 =upper_lefty + overlap_y2

          lrx_grid2 =lrx_grid1 + + overlap_x1
          lry_grid2 =upper_lefty - overlap_y1

          center_grid2_x =((ulx_grid2 + lrx_grid2) / 2)
          center_grid2_y =((uly_grid2 + lry_grid2) / 2)

          print("Image footprint coordinates: ( ulx_grid2: {}, uly_grid2: {}, lrx_grid2: {}, lry_grid2:{})".format(ulx_grid2,uly_grid2,lrx_
          print("Image perspective center coordinates: ( X: {}, Y: {}, Z: {})".format(center_grid2_x,center_grid2_y,flight_altitude))

          Image footprint coordinates: ( ulx_grid2: 499615.0, uly_grid2: 4265610.9, lrx_grid2: 503539.0, lry_grid2:4259607.9)
          Image perspective center coordinates: ( X: 501577.0, Y: 4262609.4, Z: 5203.846153846153)
```

Figure 3

First Question Results ;

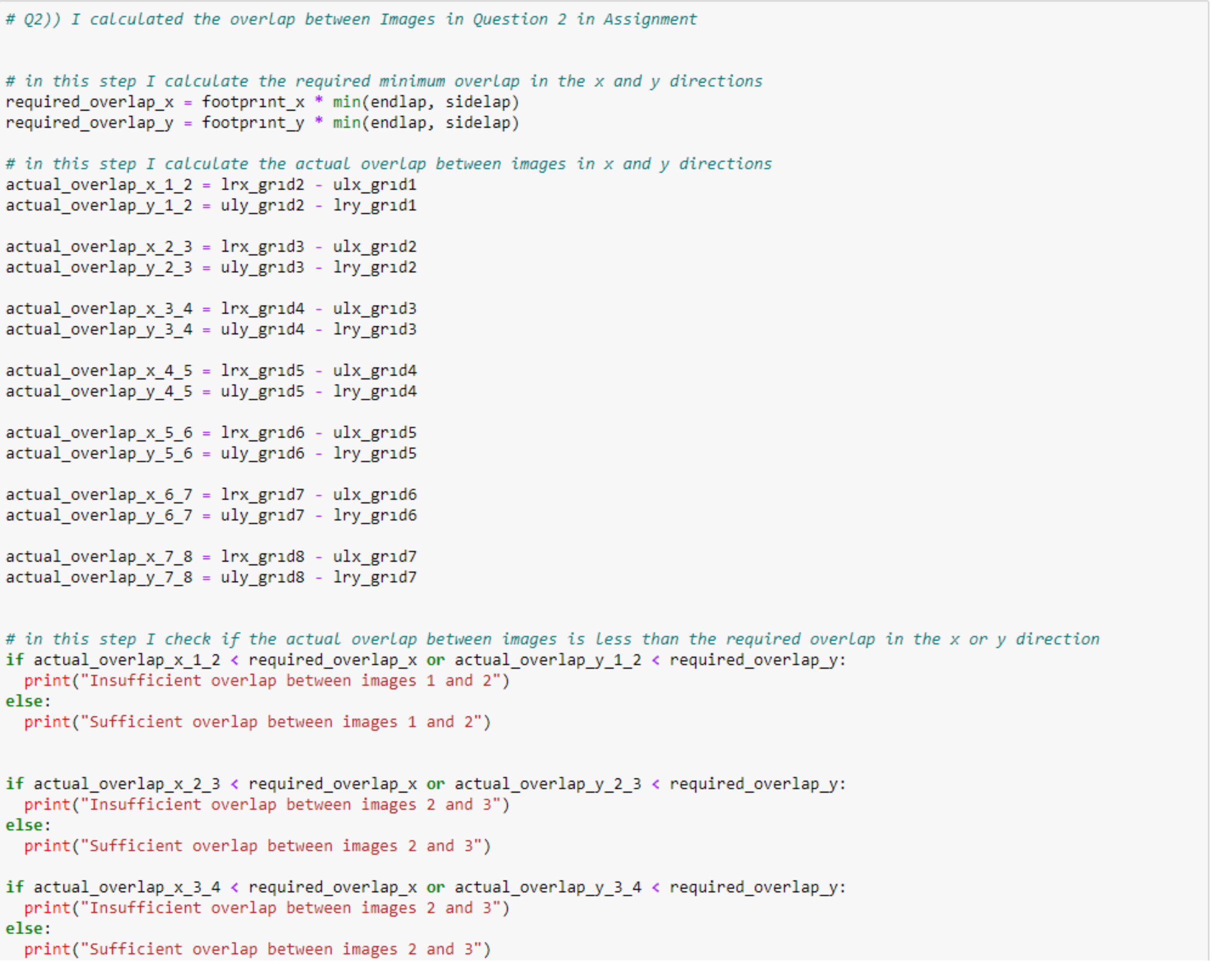


```
In [243]: print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Image #    |    ULX     | ULY        | LRX        | LRY        |      Center X  |     Center Y   |  Center Z")
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 1     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid1, uly_grid1
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 2     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid2, uly_grid2
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 3     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid3, uly_grid3
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 4     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid4, uly_grid4
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 5     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid5, uly_grid5
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 6     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid6, uly_grid6
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 7     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid7, uly_grid7
          print("+------------+------------+------------+------------+------------+------------+------------+------------+")
          print("| Grid 8     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}     | {:.1f}".format(ulx_grid8, uly_grid8

          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Image #    |    ULX     | ULY        | LRX        | LRY        |      Center X  |     Center Y   |  Center Z
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 1     | 497260.6     | 4265610.9     | 501184.6     | 4259607.9     | 499222.6     | 4262609.4     | 5203.8
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 2     | 499615.0     | 4265610.9     | 503539.0     | 4259607.9     | 501577.0     | 4262609.4     | 5203.8
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 3     | 501184.6     | 4265610.9     | 505108.6     | 4259607.9     | 503146.6     | 4262609.4     | 5203.8
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 4     | 503539.0     | 4265610.9     | 507463.0     | 4259607.9     | 505501.0     | 4262609.4     | 5203.8
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 5     | 503539.0     | 4256190.9     | 507463.0     | 4250187.9     | 505501.0     | 4253189.4     | 5203.8
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 6     | 501184.6     | 4256190.9     | 505108.6     | 4250187.9     | 503146.6     | 4253189.4     | 5203.8
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 7     | 499615.0     | 4256190.9     | 503539.0     | 4250187.9     | 501577.0     | 4253189.4     | 5203.8
          +------------+------------+------------+------------+------------+------------+------------+------------+
          | Grid 8     | 497260.6     | 4256190.9     | 501184.6     | 4250187.9     | 499222.6     | 4253189.4     | 5203.8
```

2. Question

I determined if there is insufficient overlap between images based on footprint coordinates.

In this code, let's calculate the minimum trap required in the x and y directions, based on the smaller of the end trap and side overlap. I then calculated the actual overlap between the first and second images in the x and y directions by taking the difference between the bottom right x coordinate of the next image and the top left x coordinate and top left y coordinate of the previous image. Finally, I checked if the actual overlap between each pair of images was less than the required overlap in the x or y direction and printed a message accordingly



```
# Q2)) I calculated the overlap between Images in Question 2 in Assignment


# in this step I calculate the required minimum overlap in the x and y directions
required_overlap_x = footprint_x * min(endlap, sidelap)
required_overlap_y = footprint_y * min(endlap, sidelap)

# in this step I calculate the actual overlap between images in x and y directions
actual_overlap_x_1_2 = lrx_grid2 - ulx_grid1
actual_overlap_y_1_2 = uly_grid2 - lry_grid1

actual_overlap_x_2_3 = lrx_grid3 - ulx_grid2
actual_overlap_y_2_3 = uly_grid3 - lry_grid2

actual_overlap_x_3_4 = lrx_grid4 - ulx_grid3
actual_overlap_y_3_4 = uly_grid4 - lry_grid3

actual_overlap_x_4_5 = lrx_grid5 - ulx_grid4
actual_overlap_y_4_5 = uly_grid5 - lry_grid4

actual_overlap_x_5_6 = lrx_grid6 - ulx_grid5
actual_overlap_y_5_6 = uly_grid6 - lry_grid5

actual_overlap_x_6_7 = lrx_grid7 - ulx_grid6
actual_overlap_y_6_7 = uly_grid7 - lry_grid6

actual_overlap_x_7_8 = lrx_grid8 - ulx_grid7
actual_overlap_y_7_8 = uly_grid8 - lry_grid7


# in this step I check if the actual overlap between images is less than the required overlap in the x or y direction
if actual_overlap_x_1_2 < required_overlap_x or actual_overlap_y_1_2 < required_overlap_y:
  print("Insufficient overlap between images 1 and 2")
else:
  print("Sufficient overlap between images 1 and 2")


if actual_overlap_x_2_3 < required_overlap_x or actual_overlap_y_2_3 < required_overlap_y:
  print("Insufficient overlap between images 2 and 3")
else:
  print("Sufficient overlap between images 2 and 3")

if actual_overlap_x_3_4 < required_overlap_x or actual_overlap_y_3_4 < required_overlap_y:
  print("Insufficient overlap between images 2 and 3")
else:
  print("Sufficient overlap between images 2 and 3")
```

```
if actual_overlap_x_4_5 < required_overlap_x or actual_overlap_y_4_5 < required_overlap_y:
  print("Insufficient overlap between images 2 and 3")
else:
  print("Sufficient overlap between images 2 and 3")

if actual_overlap_x_5_6 < required_overlap_x or actual_overlap_y_5_6 < required_overlap_y:
  print("Insufficient overlap between images 2 and 3")
else:
  print("Sufficient overlap between images 2 and 3")

if actual_overlap_x_6_7 < required_overlap_x or actual_overlap_y_6_7 < required_overlap_y:
  print("Insufficient overlap between images 2 and 3")
else:
  print("Sufficient overlap between images 2 and 3")

if actual_overlap_x_7_8 < required_overlap_x or actual_overlap_y_7_8 < required_overlap_y:
  print("Insufficient overlap between images 2 and 3")
else:
  print("Sufficient overlap between images 2 and 3")
```

```
Sufficient overlap between images 1 and 2
Sufficient overlap between images 2 and 3
Sufficient overlap between images 2 and 3
Insufficient overlap between images 2 and 3
Sufficient overlap between images 2 and 3
Sufficient overlap between images 2 and 3
Sufficient overlap between images 2 and 3
```