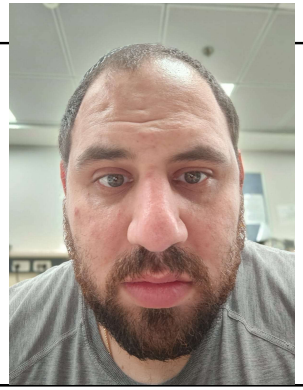
Course: Image Processing 31651
Assignment #21
Pixel to Pixel Operations (Part 1)

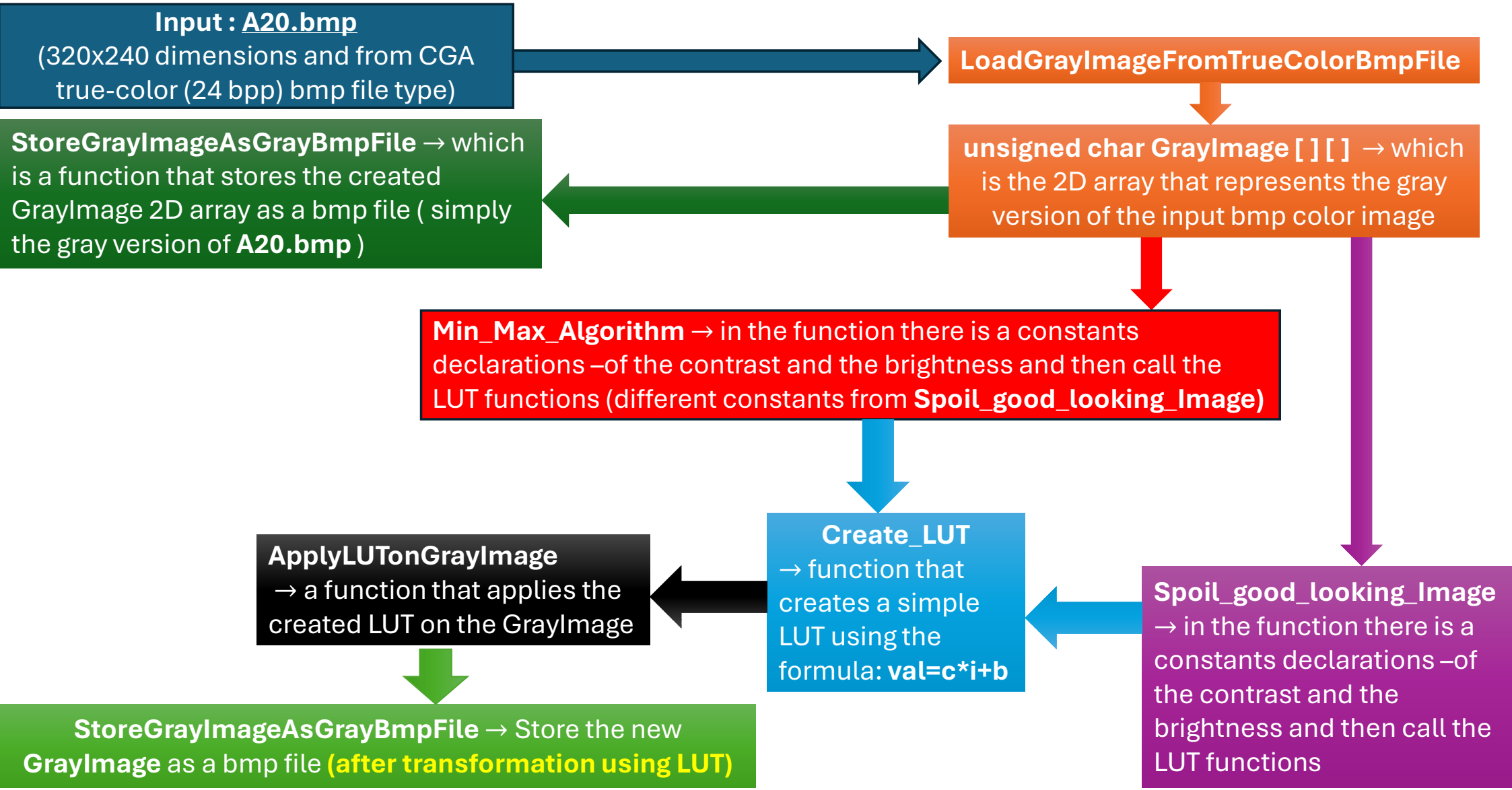
	ID (4 last digits)	Shorten Name	Photo of the student
Student #1	**1950**	shienfeld	
Student #2	**2210**	pony	
Student #3	**7939**	akimov	

Assignment #21: Process set of Gray Images

In the file ImProcInPlainC.h defined numerical values of:
NUMBER_OF_ROWS and NUMBER_OF COLUMNS

#	Image Description
1	Find in the Internet Color PHOTO of a set of beautiful fruits and vegetables (at least 6 different types). *(Joke about Vovochka – 5 animals at Africa)* ***Photo must have natural and non-trivial background*** Resize the image as CGA true color (24 bpp) bmp file. A20.bmp Convert color image A20.bmp to gray image A20gray.bmp Spoil "good looking image": Calculate LUT for the **contrast** = 0.22+0.023*B+0.003*C ; **brightness** = (23 + F*3 +G) (B, C ..– from short ID) Apply LUT to A20gray.BMP. **Pointers must be used to access the pixels**. Store processed Image as A211.bmp
2	Apply MIN-MAX algorithm to A211.bmp. **The algorithm must use pointers to access the pixels**. Store processed Image as A212.bmp
	For report use YOUR OWN STYLE. From YOUR Power Point report customer must understand what was done, how and what are the results (Lecturer may later ask to add slides) **Done: 21.1 21.2** **REMOVE ALL RED when the items are ready** **Important hint: Fill report pages while working.** **Do not complain "not enough time" in case you did not follow THIS rule.**

21.0 - flow chart of the code



21.1 – “Spoil_good_looking_Image” function

```
43  void Spoil_good_looking_Image(unsigned char img[][NUMBER_OF_COLUMNS])
44  {
45      // from short ID => ABCD=1950 (BARAK'S ID) and EFGH=2210 (OMRI'S ID)
46
47      double c = 0.442;  //  c=0.22+0.023*B+0.003*C => ABCD=1950 ==> B=9,C=5 ==> c=0.442
48      double b = 30; //23+3*F+G => EFGH=2210 ==> F=2,G=1 ==> b=30
49
50      Create_LUT(LUT, c, b);
51      ApplyLUTonGrayImage(GrayImage, LUT);
52  }
```

After ‘c’ and ‘b’ declarations – create using them a simple LUT and then apply the LUP on the gray image (img [ ] [ ])

Pay attention to ‘c’ and ‘b’ values – corresponding the IDs of the group

21.2 – “Create_LUT” and “ApplyLUTonGrrayImage” functions

```
21  void Create_LUT(unsigned char LUT[], double c, double b)
22  {
23      for (unsigned i = 0; i <= PIXEL_MAXVAL; i++)
24      {
25          double val = c * i + b;
26          if (val > PIXEL_MAXVAL)
27              LUT[i] = PIXEL_MAXVAL;
28          else if (val < PIXEL_MINVAL)
29              LUT[i] = PIXEL_MINVAL;
30          else
31              LUT[i] = val;
32      }
33  }
```

Create_LUT is a very simple function for creating a LUT using the contrast (c) and brightness (b) constants.

```
35  void ApplyLUTonGrayImage(unsigned char img[][NUMBER_OF_COLUMNS], unsigned char LUT[])
36  {
37      for (int row = 0; row < NUMBER_OF_ROWS; row++)
38      {
39          for (int col = 0; col < NUMBER_OF_COLUMNS; col++)
40          {
41              unsigned char* pixel = &img[row][col];
42              *pixel = LUT[*pixel];
43          }
44      }
45  }
```

After defining the LUT – we need to apply the LUT of the original gray image (img [ ] [ ])

21.3 – "CreateMinMaxLUT" and "Min_Max_Algorithm" functions

```
59  void CreateMinMaxLUT(unsigned char LUT[], unsigned char min, unsigned max)
60  {
61      double c = 255./ (max - min);
62      cout << "c" << c << endl;
63      double b = -c * min;
64      Create_LUT(LUT, c, b);
65  }
```

Another LUT creation function – now for the min_max_algorithm.
The max value is 255 and the min value is 0 (in grayscale). Now, the 'c' and the 'b' are defined differently then they were defined for the Spoil_good_looking_Image algorithm

```
67  void Min_Max_Algorithm(unsigned char img[][NUMBER_OF_COLUMNS])
68  {
69      unsigned char min, max;
70      min = PIXEL_MAXVAL;
71      max = PIXEL_MINVAL;
72      for (int row = 0; row < NUMBER_OF_ROWS; row++)
73      {
74          for (int col = 0; col < NUMBER_OF_COLUMNS; col++)
75          {
76              unsigned char* pixel = &img[row][col];
77              if (*pixel < min)
78                  min = *pixel;
79              if (*pixel > max)
80                  max = *pixel;
81          }
82      }
83      CreateMinMaxLUT(LUT, min, max);
84      cout << "max " << (int)max << endl;
85      cout << "min " << int(min) << endl;
86      cout << "pixel_maxval " << int(PIXEL_MAXVAL) << endl;
87      cout << "pixel minval " << int(PIXEL_MINVAL) << endl;
88      ApplyLUTonGrayImage(GrayImage, LUT);
89  }
```

The first part of the function finds the minimum and maximum pixel values in the image.

min is initially set to the maximum possible pixel value (255), and max is set to the minimum possible pixel value (0).

The function then iterates through each pixel in the image. For each pixel : If the pixel value is less than min, min is updated to this pixel value. If the pixel value is greater than max, max is updated to this pixel value.

By the end of these loops, min holds the smallest pixel value in the image, and max holds the largest pixel value.

After finding the minimum and maximum pixel values, the function creates a LUT to normalize the pixel values.

Finally, the LUT is applied to the image to adjust the pixel values.

Introduction to profiles part of the report (parts 21.4 – 21.12)

In the next slides we will present the analysis of the results using the profiler tool – rows and columns.

We decided to present two sets of points with their profiles

Parts **21.4 - 21.7** will deal with point **(152,157)** to present the profiles of the 4 versions of the image

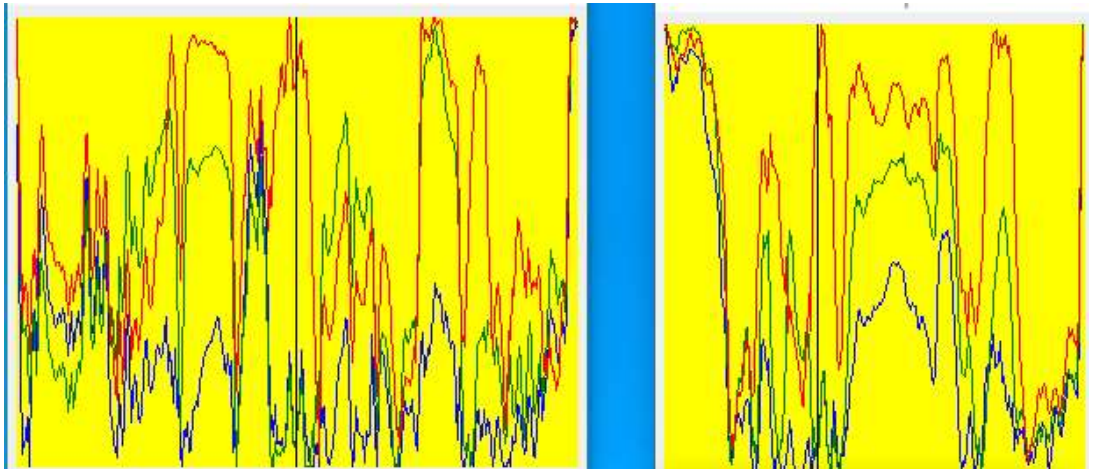
Parts **21.8 - 21.12** will deal with point **(48,279)** to present the profiles of the 4 versions of the image

Pay attention:
in all the profiles presentations row view is in the right hand side and column view is in the left hand side

21.4 – original color image and its profiles (row and column) – for a certain point (152,157)

A20.bmp (original color image)

This point appears to be a bright red color, given the high red value and low green and blue values.



location

RGB values

Point:	Mouse	Expl.	(M-E)
Row	152	152	0
Column	157	157	0
Red	214	214	0
Green	14	14	0
Blue	17	17	0

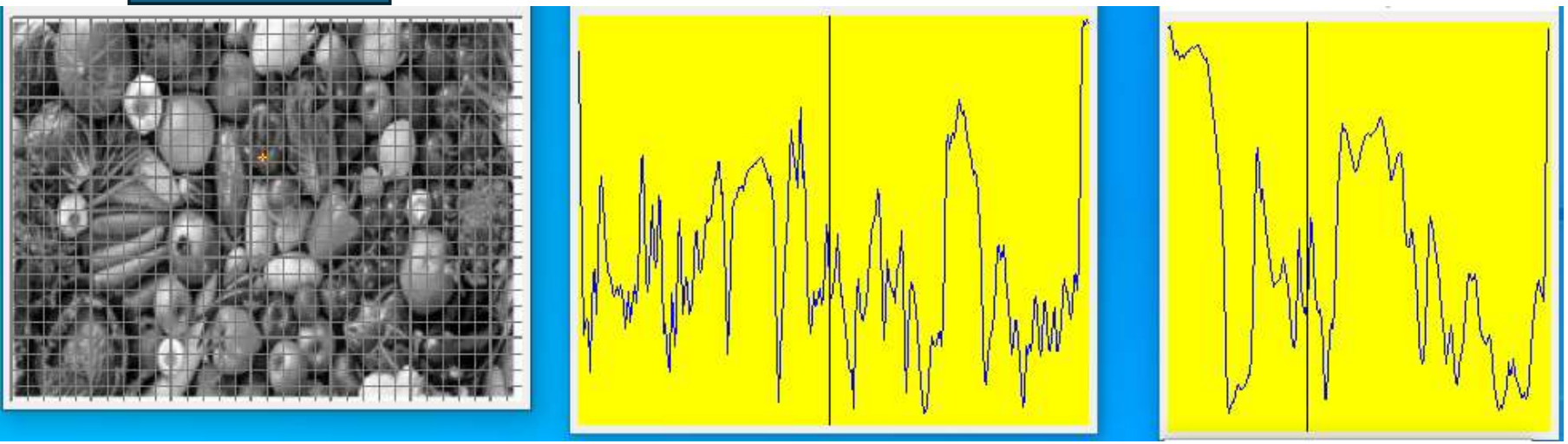
Row Profile (Row = 152):
The red, green, and blue profiles along row 152 show significant variation. The red channel shows multiple peaks, indicating areas with high red intensity. The green and blue channels are generally lower, with occasional peaks corresponding to different colors in the image.

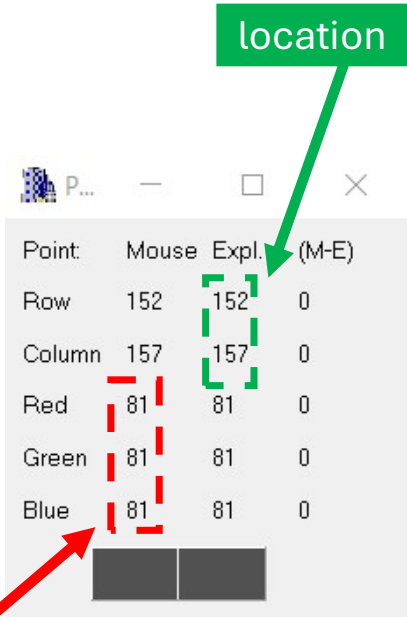
Column Profile (Column = 157):
The column profile also exhibits significant variation in all three channels. The red channel again shows prominent peaks, corresponding to red objects in the image. The green and blue channels show lower values, consistent with the overall color distribution.

21.5 – A20gray.bmp image and its profiles (row and column) – for a certain point (152,157)

A20gray.bmp

This point has a moderate grayscale intensity, indicating it is neither very bright nor very dark.





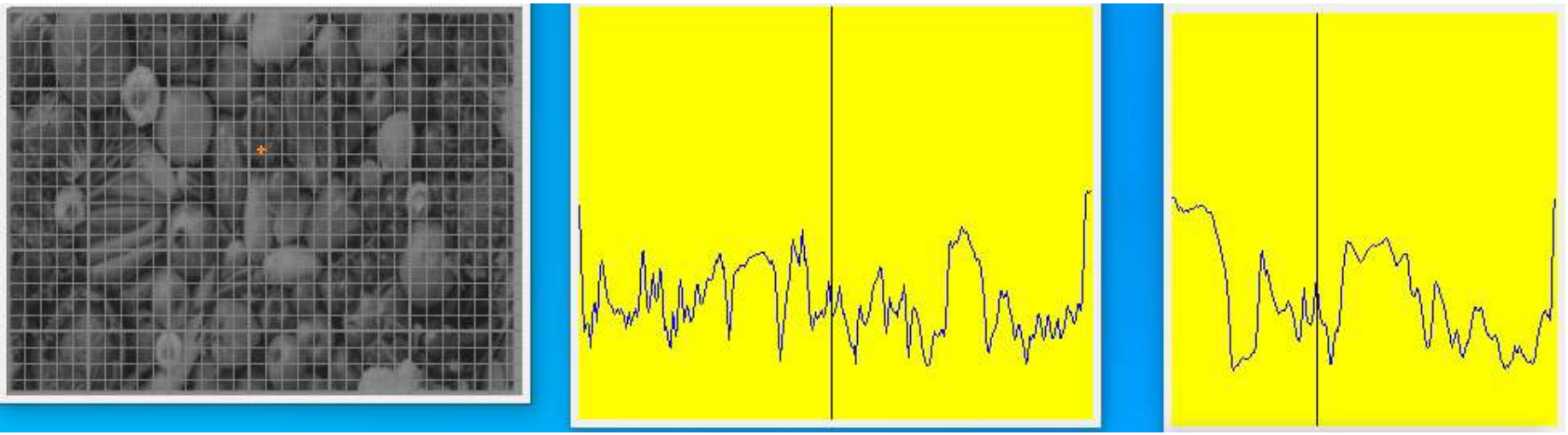
Row Profile (Row = 152):
The profile for row 152 shows significant variation, but with a single color channel (grayscale). There are several peaks and valleys, suggesting areas of varying intensity along this row. The overall intensity variation is smoother compared to the RGB profiles due to the single channel representation.

Column Profile (Column = 157):
Similar to the row profile, the column profile shows varying intensity values. Peaks and valleys in the profile indicate changes in intensity along this column.

21.6 – A211.bmp image and its profiles (row and column) – for a certain point (152,157)

A211.bmp

This value remains the same as the previous grayscale image ("A20gray"), indicating the LUT transformation didn't affect **this particular** pixel's value.



RGB values

location

Point:	Mouse	Expl.	(M-E)
Row	152	152	0
Column	157	157	0
Red	81	81	0
Green	81	81	0
Blue	81	81	0

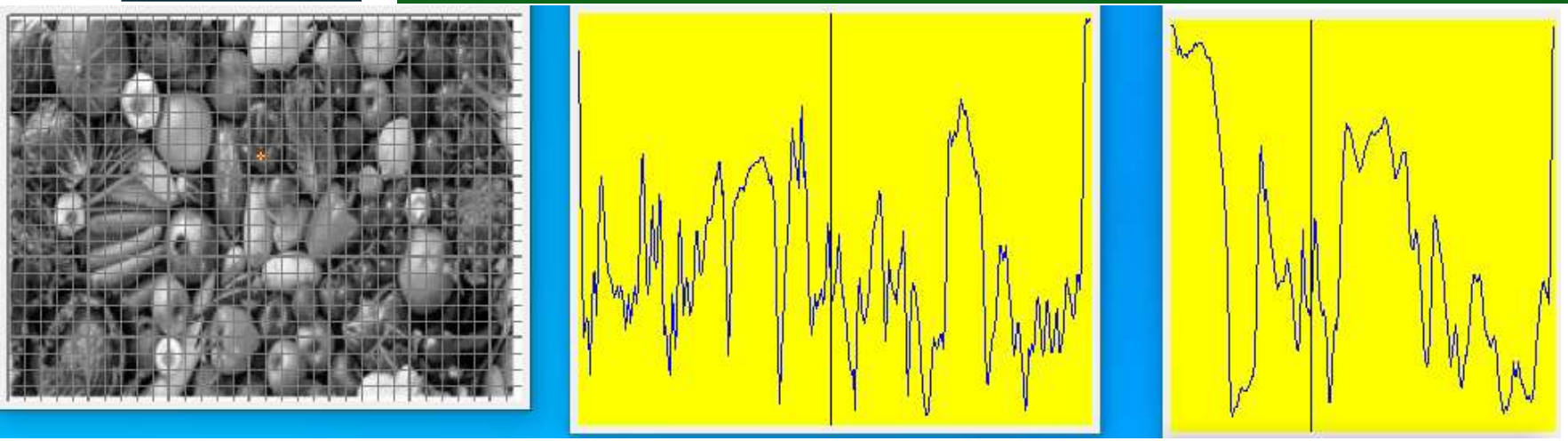
Row Profile (Row = 152):
The row profile shows similar variations to the previous grayscale image ("A20gray"), but the intensity changes may appear more pronounced or compressed depending on the LUT applied. Peaks and valleys indicate regions of varying intensity, but overall the shape of the profile remains consistent with previous observations.

Column Profile (Column = 157):
The column profile follows a similar pattern to the row profile, showing varying intensity values. Peaks and valleys indicate changes in intensity along this column, which could be enhanced or diminished based on the LUT transformation.

21.6 – A212.bmp image and its profiles (row and column) – for a certain point (152,157)

A212.bmp

The grayscale value is slightly lower than the previous value of 81, indicating some adjustment due to the Min-Max algorithm.



RGB values

location

Point:	Mouse	Expl.	(M-E)
Row	152	152	0
Column	157	157	0
Red	79	79	0
Green	79	79	0
Blue	79	79	0

Row Profile (Row = 152):
The row profile shows significant variation, similar to the previous grayscale images. Peaks and valleys indicate regions of varying intensity along this row. The adjustments due to the Min-Max algorithm might have made some intensity changes more pronounced.

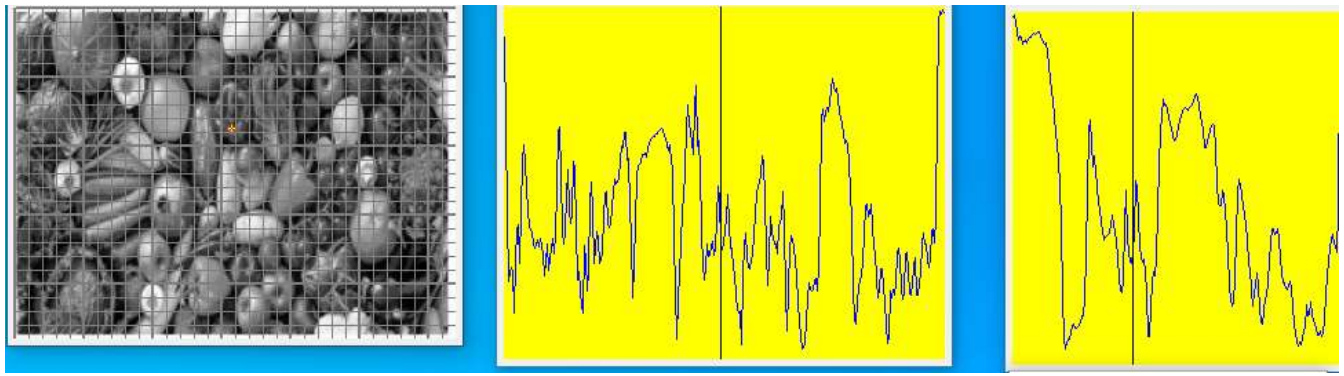
Column Profile (Column = 157):
The column profile follows a similar pattern to the row profile, showing varying intensity values.The Min-Max algorithm likely increased the contrast, making the transitions between different intensity levels more noticeable.

21.7 – comparison of the results – for a certain point (152,157)

A201gray.bmp

Original gray image (without further manipulation)

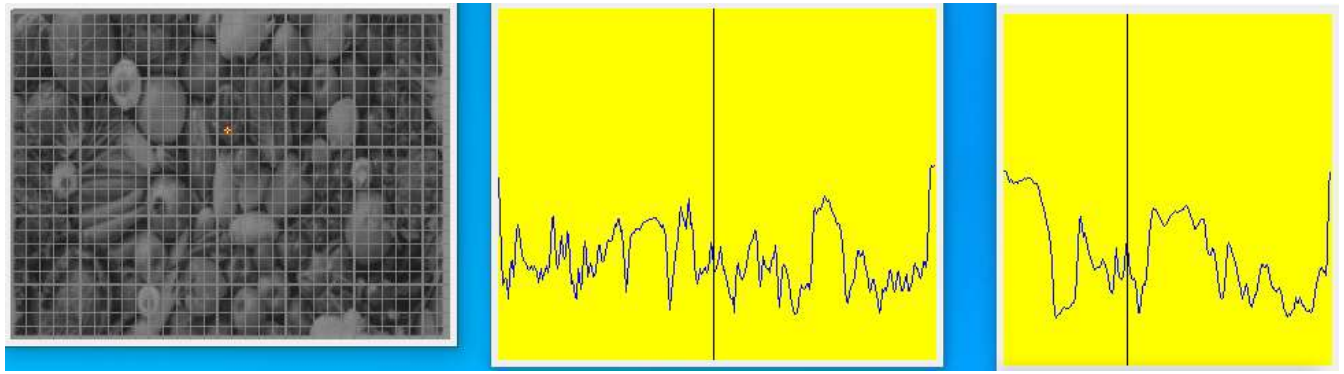
(81,81,81)=R,G,B



A2111.bmp

With spoiled_Image algorithm

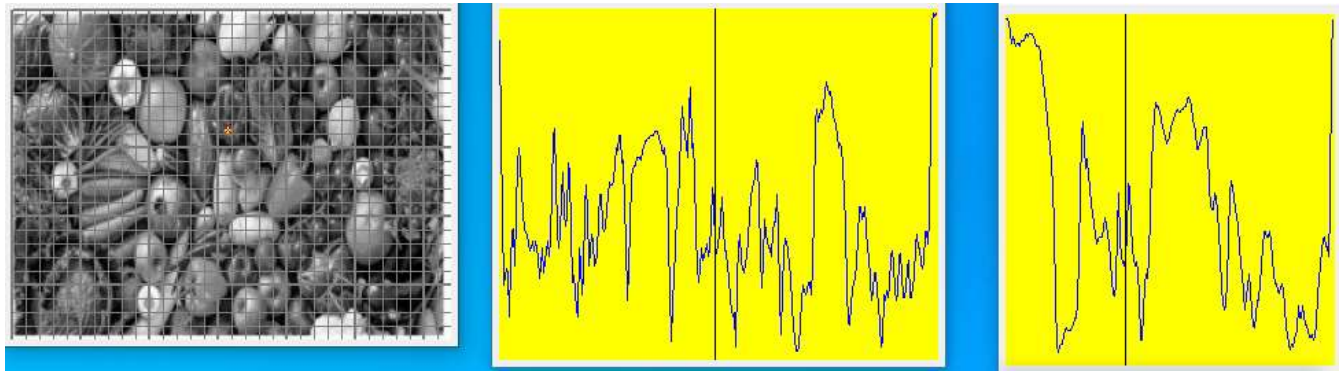
(81,81,81)=R,G,B



A2121.bmp

With min_max_algorithm

(79,79,79)=R,G,B

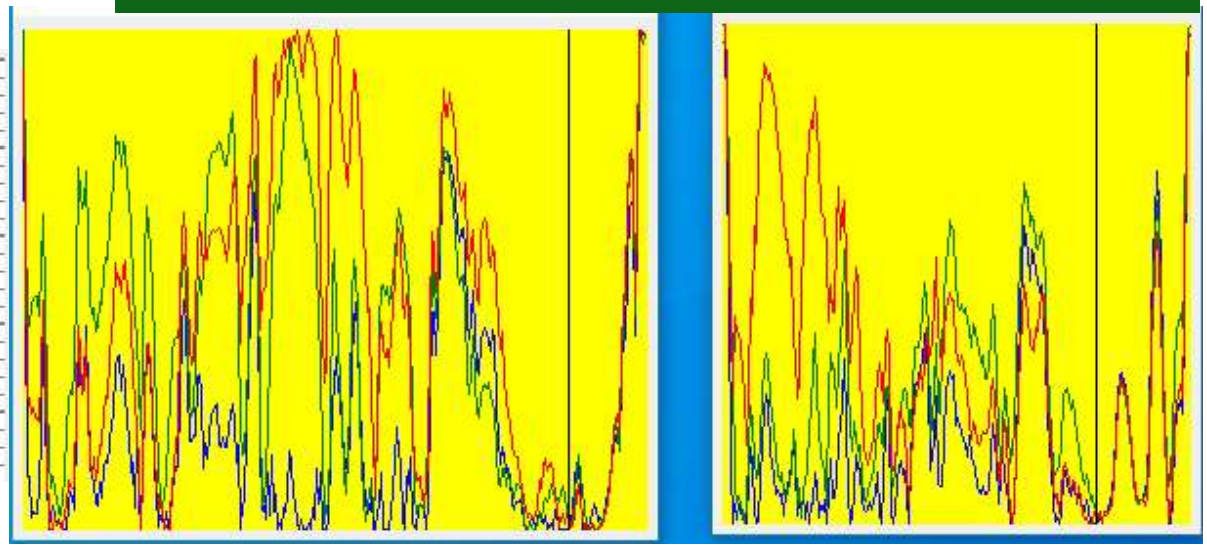


Both for the point (152,157)

21.8 – original color image and its profiles (row and column) – for a certain point (48,279)

A20.bmp (original color image)

This point appears to be very dark, primarily composed of green with a slight amount of red.



location

RGB values

Point:	Mouse	Expl	(M-E)
Row	49	49	0
Column	279	279	0
Red	6	6	0
Green	8	8	0
Blue	0	0	0

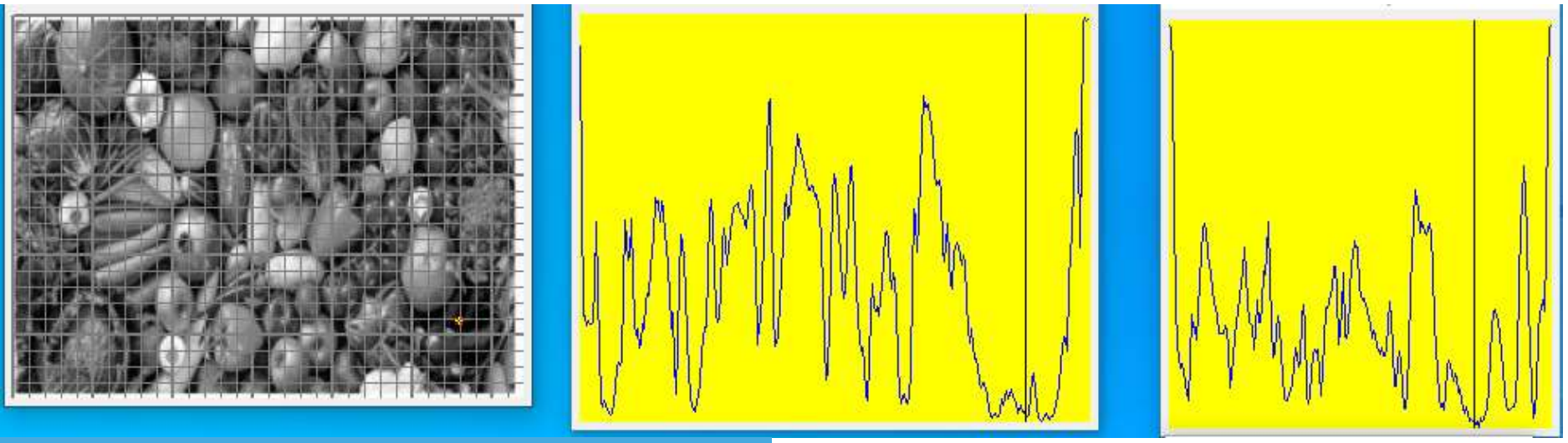
Row Profile (Row = 48):
The red, green, and blue profiles along row 49 show significant variation. The red channel shows multiple peaks, but overall lower values compared to the green channel. The green channel has higher peaks, indicating areas with higher green intensity. The blue channel is consistently low, which corresponds to the chosen point's low blue value.

Column Profile (Column = 279):
The column profile also exhibits significant variation in all three channels. The red channel shows lower values with occasional peaks. The green channel shows more pronounced peaks, reflecting higher green intensity areas. The blue channel is mostly low, consistent with the overall color distribution at the chosen point.

21.9 – A20gray.bmp image and its profiles (row and column) – for a certain point (48,279)

A20gray.bmp

This point is very dark, consistent with the low color values in the RGB image.



RGB values

location

Point:	Mouse	Expl	(M-E)
Row	48	48	0
Column	279	279	0
Red	4	4	0
Green	4	4	0
Blue	4	4	0

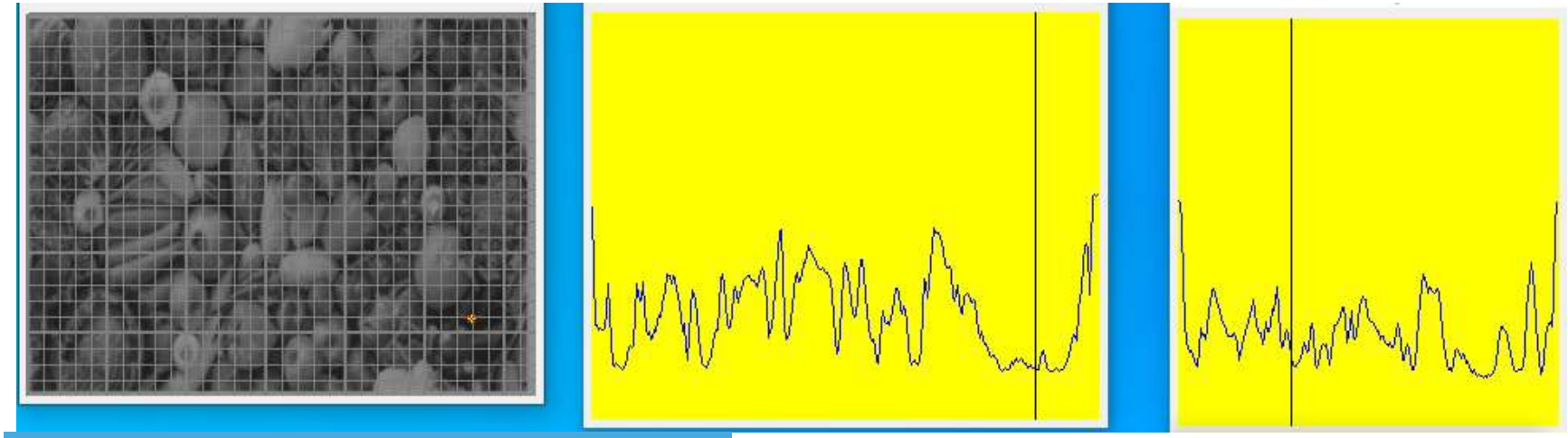
Row Profile (Row = 48):
The profile for row 49 shows significant variation in intensity values. Peaks and valleys indicate regions of varying intensity along this row, but the overall intensity is lower compared to the original RGB profile. The grayscale profile is smoother and less complex than the RGB profile due to the single channel representation.

Column Profile (Column = 279):
The column profile follows a similar pattern to the row profile, showing varying intensity values. The peaks and valleys indicate changes in intensity along this column, similar to the row profile.

21.10 – A211.bmp image and its profiles (row and column) – for a certain point (48,279)

A211.bmp

This value is higher than the previous grayscale value of 4, indicating the LUT transformation has increased the intensity at this point.



location

Point:	Mouse	Expl	(M-E)
Row	48	48	0
Column	279	279	0
Red	31	31	0
Green	31	31	0
Blue	31	31	0

RGB values

Row Profile (Row = 48):
The profile for row 49 shows significant variation in intensity values. Peaks and valleys indicate regions of varying intensity along this row. The overall intensity is higher compared to the original grayscale profile. The LUT transformation has adjusted the intensity values, making the variations more pronounced.

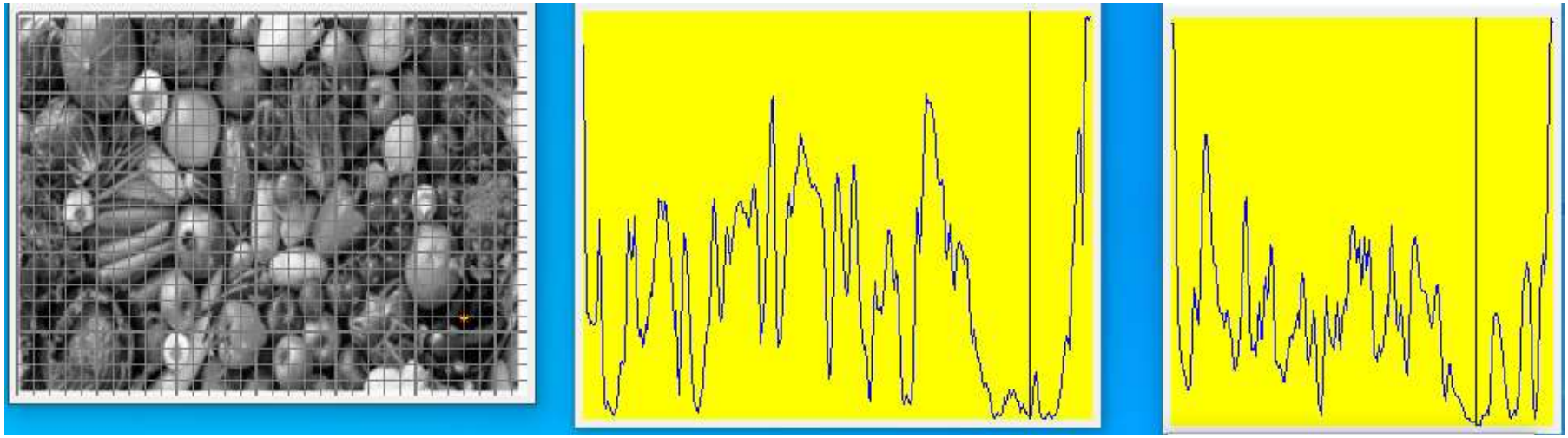
Column Profile (Column = 279):
The column profile follows a similar pattern to the row profile, showing varying intensity values. The peaks and valleys indicate changes in intensity along this column, similar to the row profile.

21.11 – A212.bmp image and its profiles (row and column) – for a certain point (48,279)

A212.bmp

This value is lower than the previous grayscale values of 4 and 31, indicating the Min-Max algorithm has adjusted the intensity to a very low level at this point.

location



Point:	Mouse	Expl.	(M-E)
Row	48	48	0
Column	279	279	0
Red	2	2	0
Green	2	2	0
Blue	2	2	0

RGB values

Row Profile (Row = 48):
The profile for row 49 shows significant variation in intensity values. Peaks and valleys indicate regions of varying intensity along this row. The overall intensity is generally lower compared to the original grayscale profile. The Min-Max algorithm has stretched the intensity values to span a wider range, resulting in a more pronounced dynamic range.

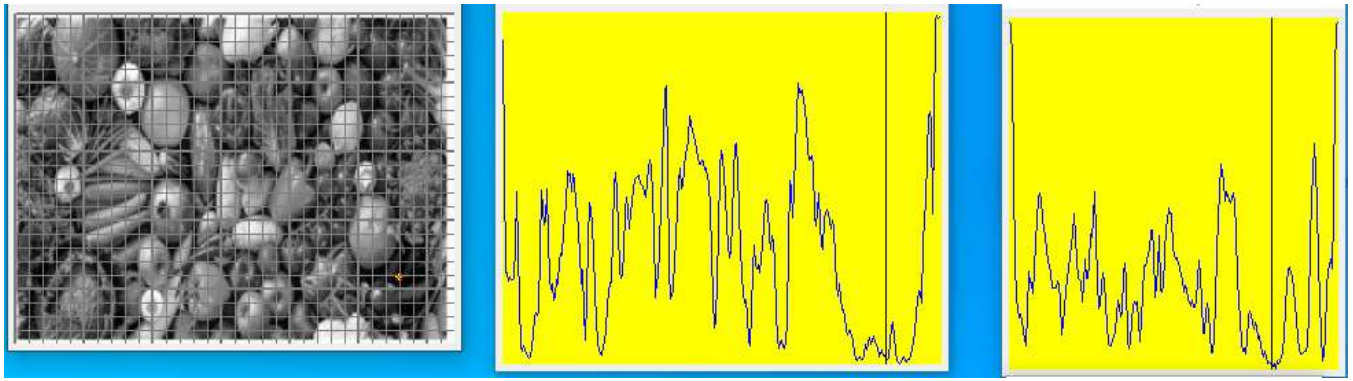
Column Profile (Column = 279):
The column profile follows a similar pattern to the row profile, showing varying intensity values. The peaks and valleys indicate changes in intensity along this column, similar to the row profile.

21.12 – comparison of the results – for a certain point (48,279)

A201gray.bmp

Original gray image (without further manipulation)

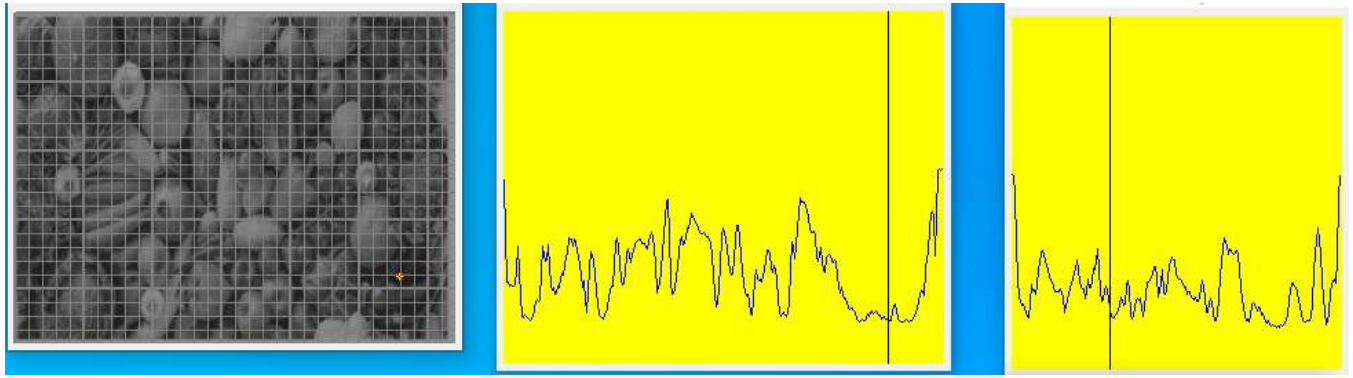
(4,4,4)=R,G,B



A2111.bmp

With spoiled_Image algorithm

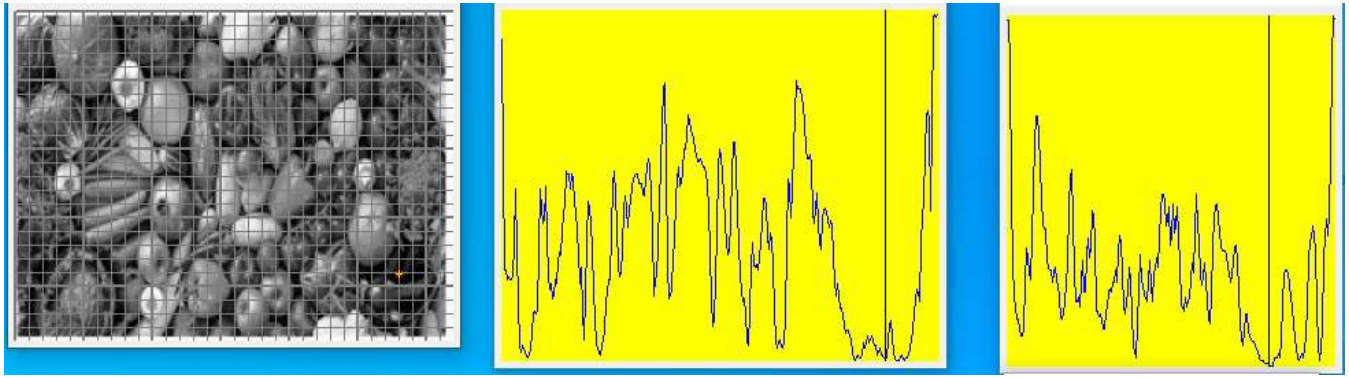
(31,31,31)=R,G,B



A2121.bmp

With min_max_algorithm

(2,2,2)=R,G,B



Both for the point (48,279)

21.13 What did we learned

1) We learned how to implement algorithms and functions that deals with image properties such as: contrast & brightness.

2) We learned how to find the optimal outcome of an image using mathematical way so the image will look the best.

3) We used, practiced and implemented methods involving LUT which is an essential concept in Image Processing