Digital Image Formation and Processing

Functions Implemented:

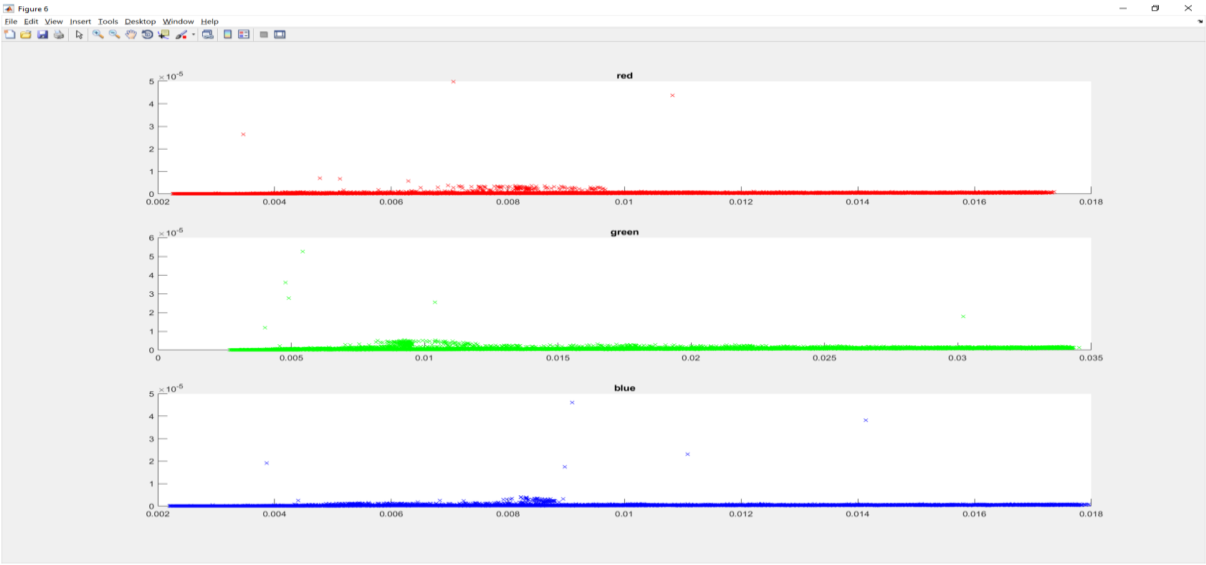
Fn Name	Description
main [input = none] [output = none]	This function is main method of the assignment, which call the other functions as required in tasks. It also loads the input images and convert them into double. Also, it provides input to other methods.
visualiseImgAndIllusBayerArray [input = img1,img2] [output = none]	This function illustrates the bayer array in the input image. It takes the input of both images.
getRGBImgFromBayerArray [input = bayer mosaic grayscale image] [output = rgb image]	This function takes the raw bayer mosiac gray scale image and output the the rgb channels according to 'RGGB' pattern.[1]
visualiseSigDepdtVarInEachSubChannel [input = img channel] [output ={mean_vec,var_vec}]	This function takes input an image channel and returns vector of mean and variances for [8 8] blocks of image channel. It uses blocproc to calculate these results.
applyFwdRootTransformation [input = img channel, a, b] [output = transformed img channel]	This function implements forward root transformation as described Task 6, it takes input of image channel and coefficients a and b derved from fitting straight line on noisy image's mean and var scatter plot for the same channel.
decorr [input = rgb image] [output = rgb image with decorrelated colors]	This function decorrelates the colors in rgb image using the mask provided in reference paper on DCT denoising [2]. This function takes input a rgb image and returns an rgb image.
applyDCTFilter [input = image channel, threshold] [output = image channel]	This function implemets the dct transform, thresholding and idct transform steps of denoising algortihm provided in reference paper. [2]
inversedecorr [input = rgb image] [output = rgb image with decorrelated colors inversed]	This function inverses the decorrelated colors in rgb image using the mask provided in reference paper on DCT denoising [2]. This function takes input a rgb image and returns an rgb image.
applyInverseRootTransformation [input = img channel, a, b] [output = transformed img channel]	This function implements inverse root transformation as described Task 9, it takes input of image channel and coefficients a and b derved from fitting straight line on noisy image's mean and var scatter plot for the same channel.
demosaicing [input =inverse transfromed image] [output = demosaiced image]	This function implements simple demosaicing on basis of interpolating the nearest neighbour pixel [3]. It takes input as inverse root transformed image and returns rgb image.

Task12And13 [input = demosaiced image] [output = none]	This task implements the White balancing, saturation correction described in Tasks 12 and 13 of assignment.

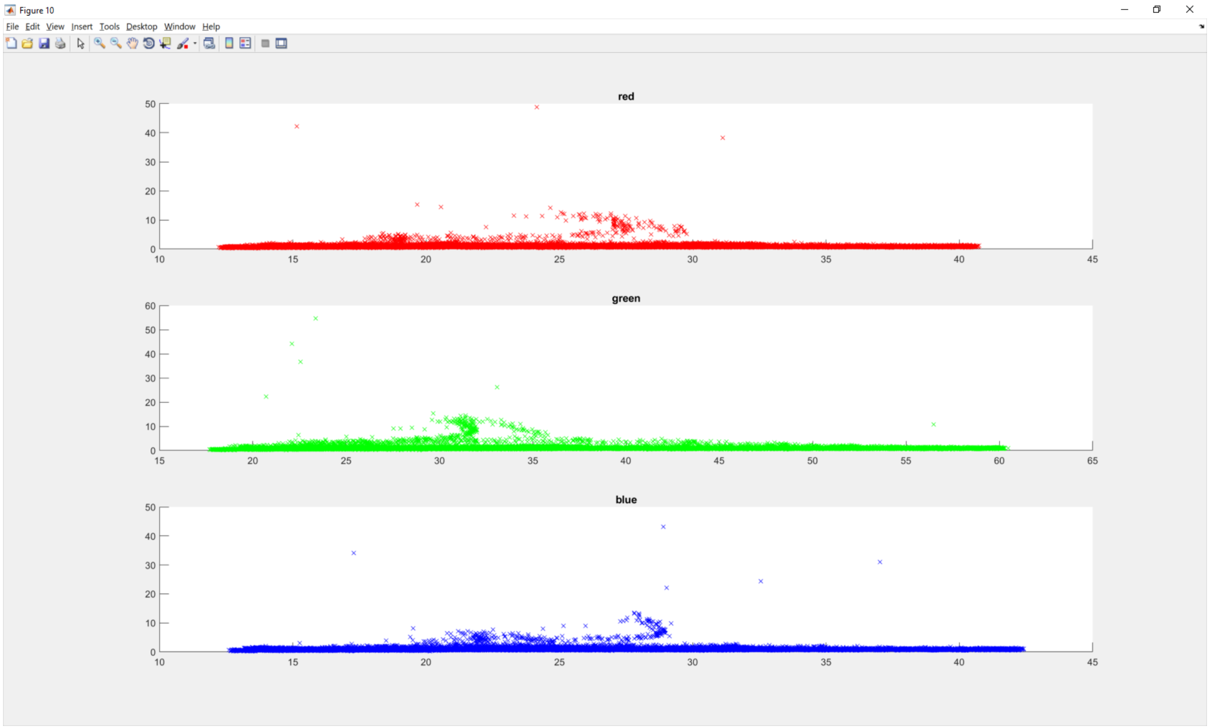
Answers :

Task 7:

Before Transformation

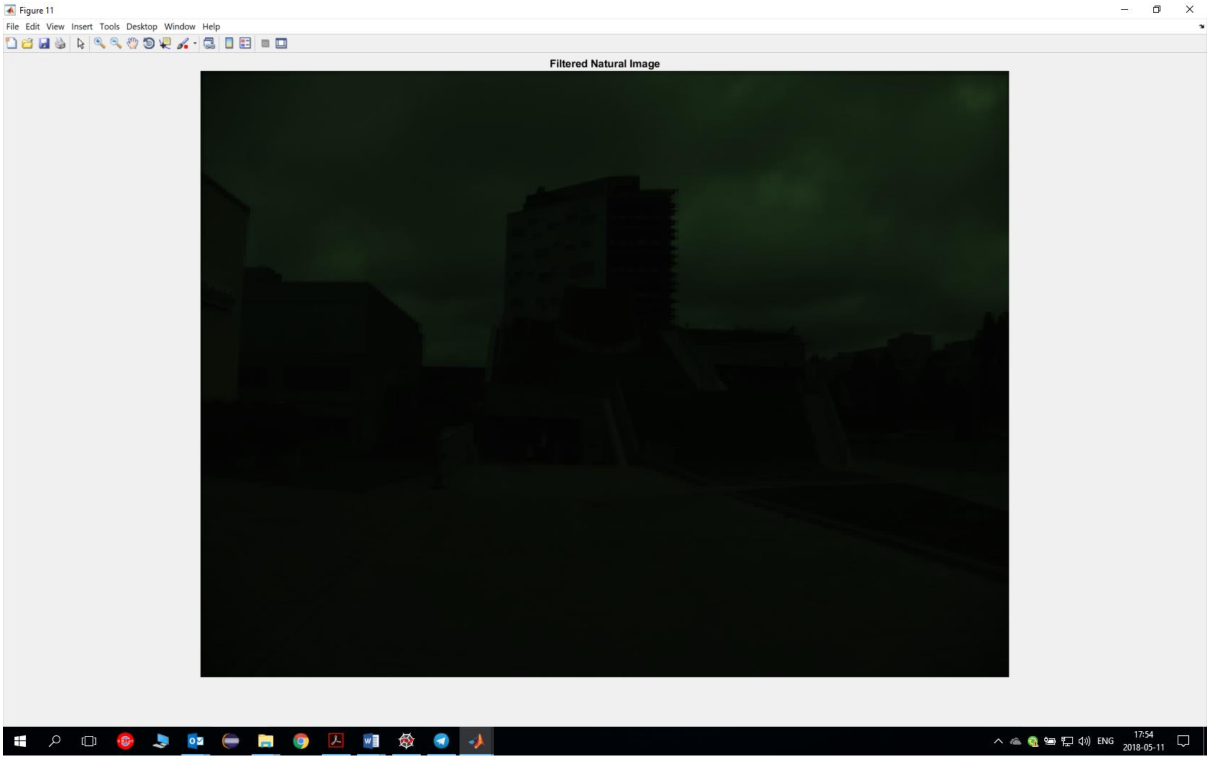


After Transformation



Outliers in the plots above represent noisy pixel intensity as noise gives higher variance in the patch of image where we are calculating the mean and variances. After transformation spread of distributions seems to be reduced and outliers looks more prominent.

Task 10:



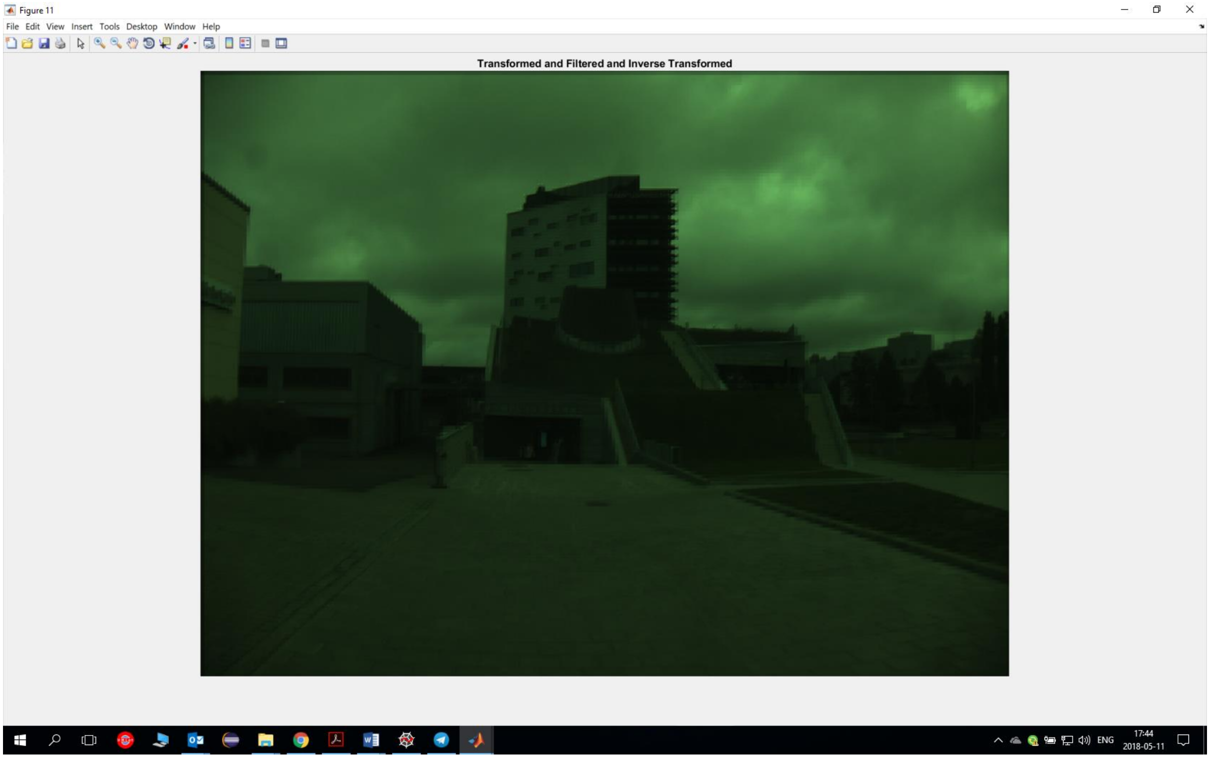


Figure 11
File Edit View Insert Tools Desktop Window Help
Transformed and Filtered and Inverse Transformed
ENG
17:44
2018-05-11

Root transformation in general expand the dynamic range of input intensity level as can be seen in the above two images. First image which was only de-noised looks very dark as compared to the second image which was transformed and inverse transformed. I am not sure why inverse root transform is not exactly inverse function of root transform. Mathematically, I can understand that not every function is invertible but not how it is being used in this task.

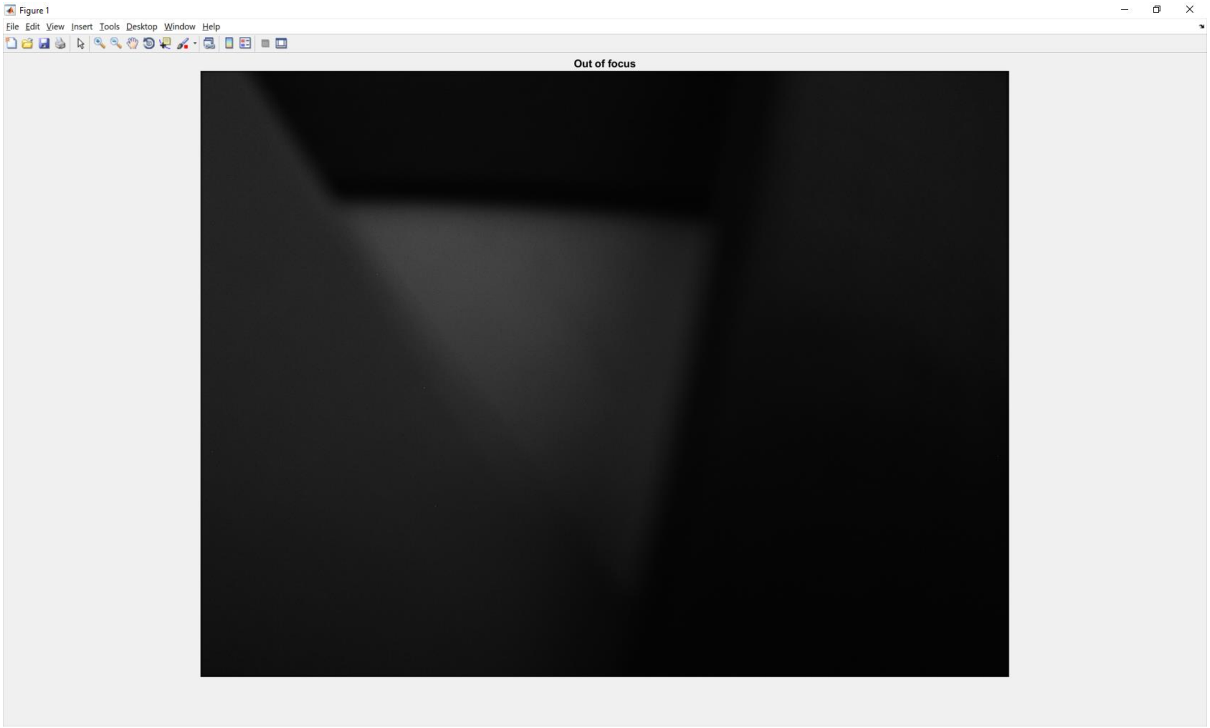
References:

[1] https://se.mathworks.com/matlabcentral/answers/353074-is-there-a-built-in-demosaicing-option-in-matlab-which-doesn-t-use-interpolation-and-instead-just-re

[2] http://www.ipol.im/pub/art/2011/ys-dct/

[3] https://se.mathworks.com/matlabcentral/fileexchange/5219-bilinear-terpolation?focused=6123822&tab=function

Outputs :



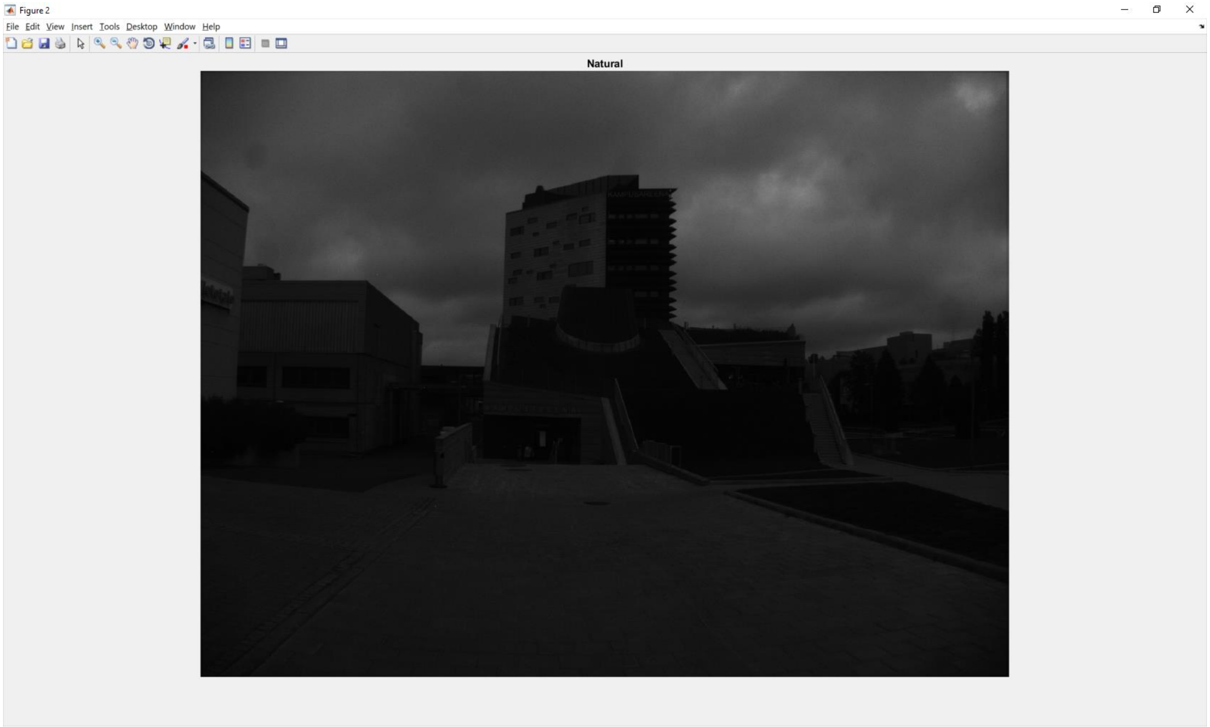


Figure 2
File Edit View Insert Tools Desktop Window Help
Natural

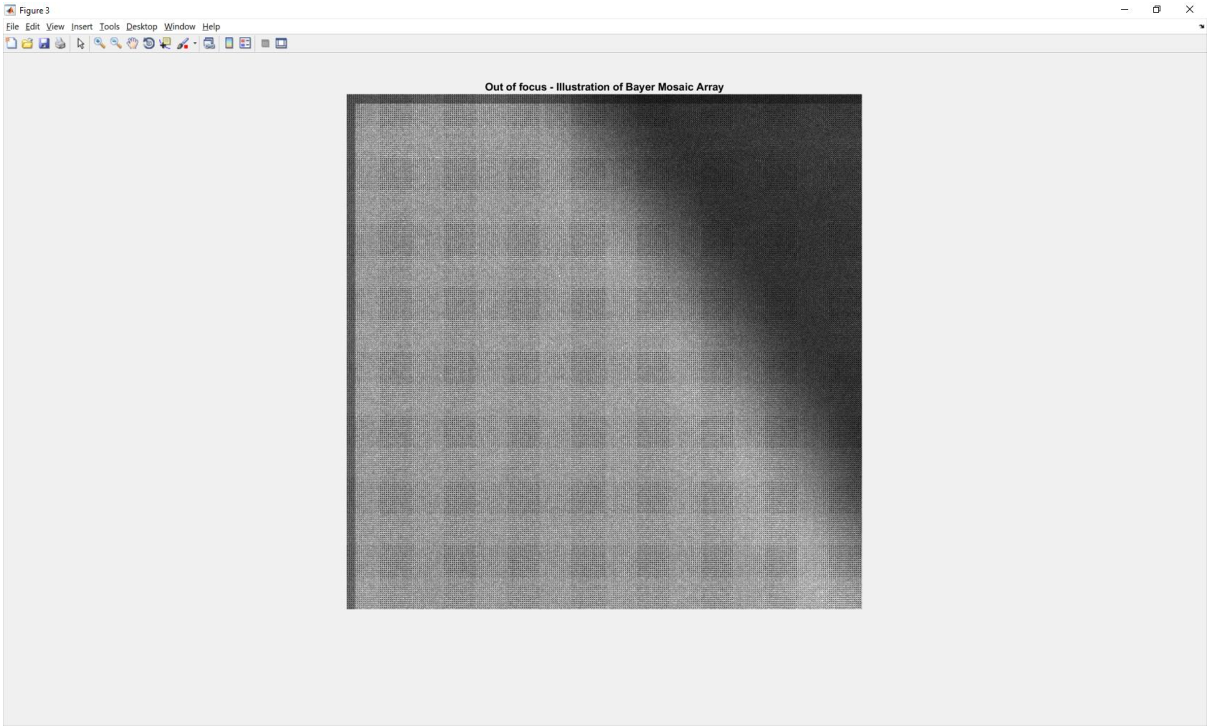


Figure 3
File Edit View Insert Tools Desktop Window Help
Out of focus - Illustration of Bayer Mosaic Array

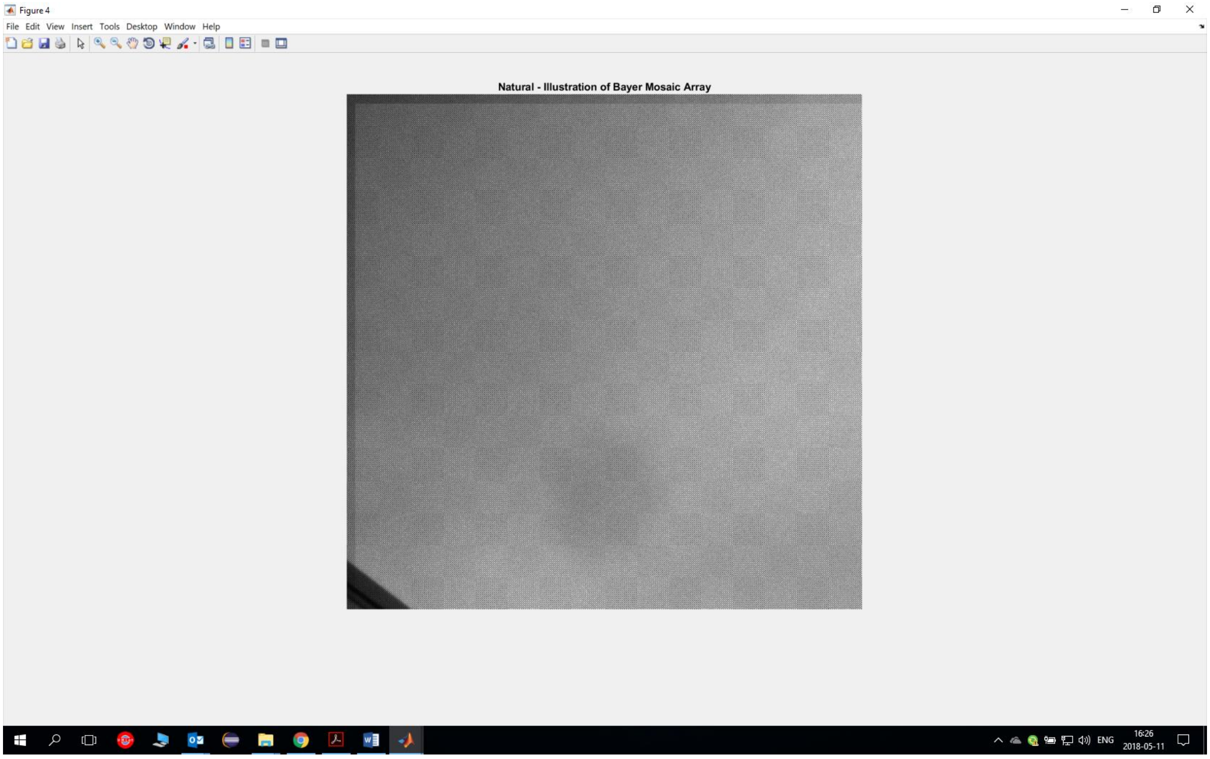


Figure 4
File Edit View Insert Tools Desktop Window Help
Natural - Illustration of Bayer Mosaic Array
ENG
16:26
2018-05-11

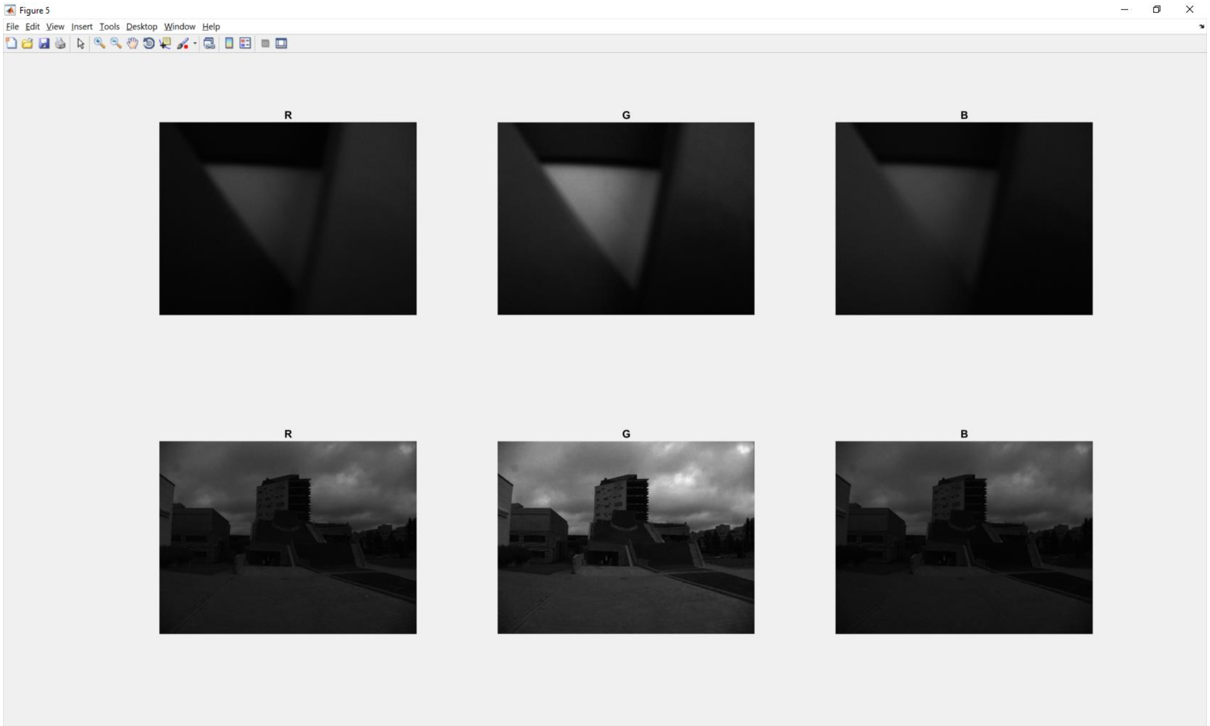


Figure 5
File Edit View Insert Tools Desktop Window Help
R
G
B
R
G
B

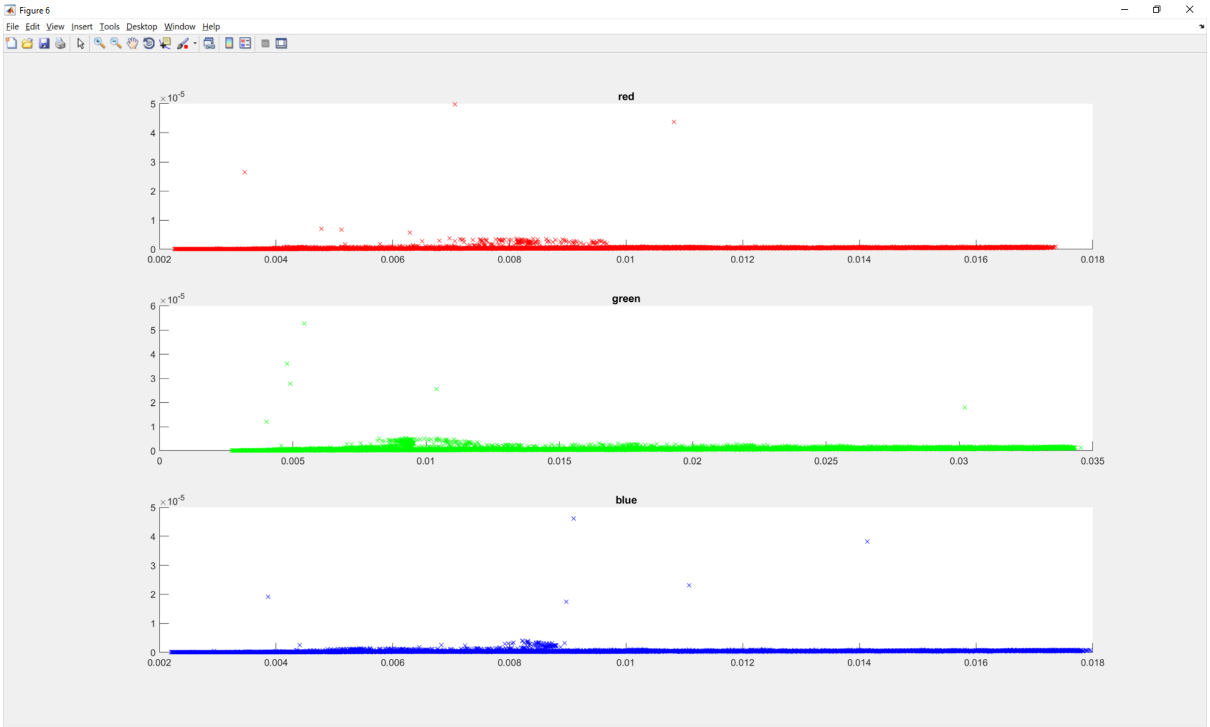


Figure 6
File Edit View Insert Tools Desktop Window Help
red
×10^-5
5
4
3
2
1
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
green
×10^-5
6
5
4
3
2
1
0
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
blue
×10^-5
5
4
3
2
1
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018

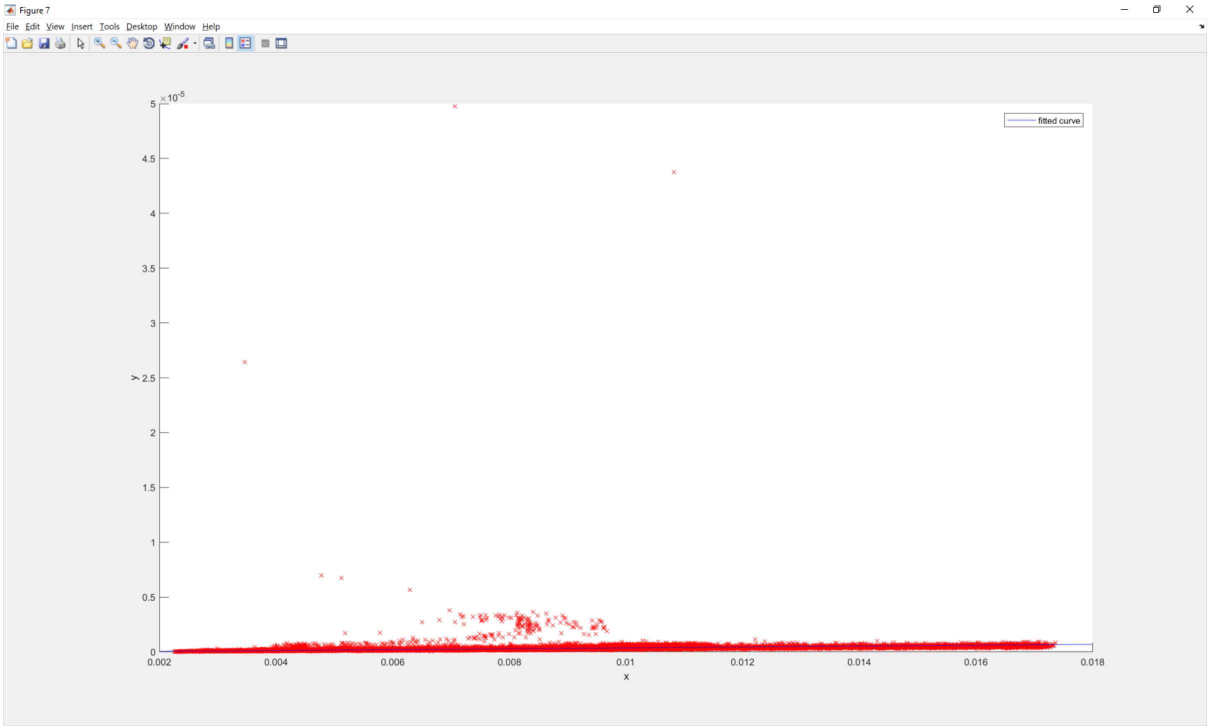


Figure 7
File Edit View Insert Tools Desktop Window Help
×10^-5
5
4.5
4
3.5
3
y 2.5
2
1.5
1
0.5
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
x
fitted curve

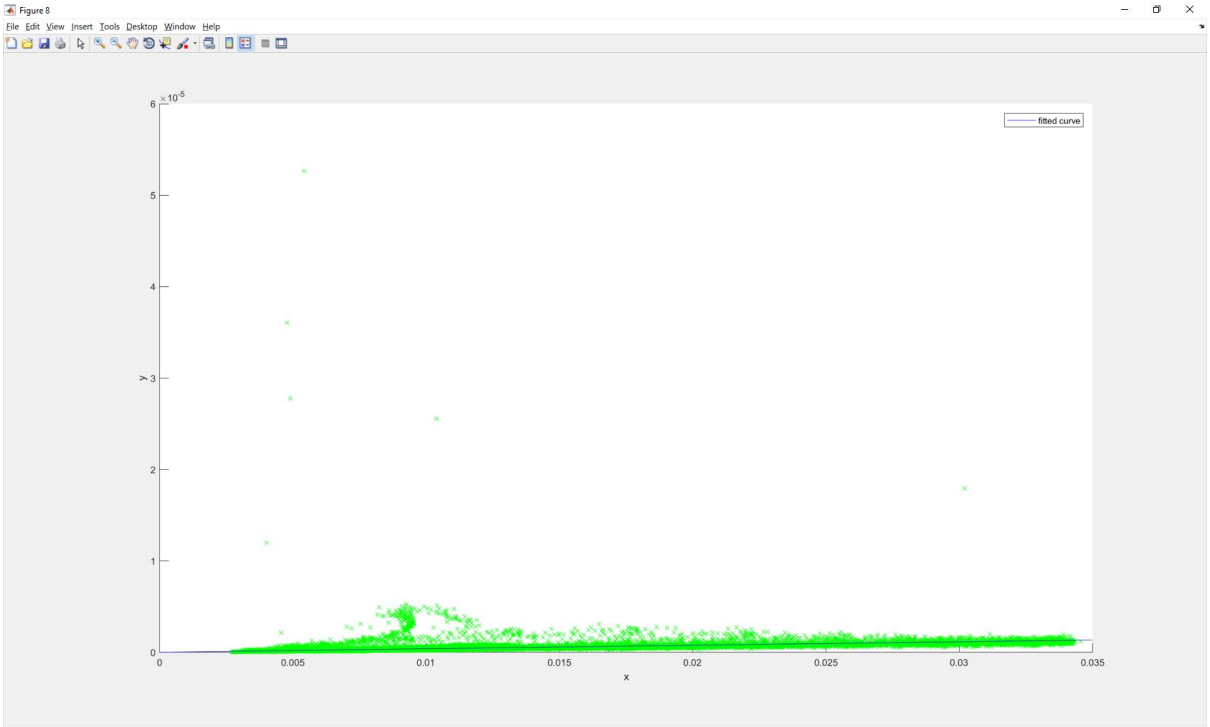


Figure 8
File Edit View Insert Tools Desktop Window Help
×10^{-5}
6
5
4
3
2
1
0
y
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
x
fitted curve

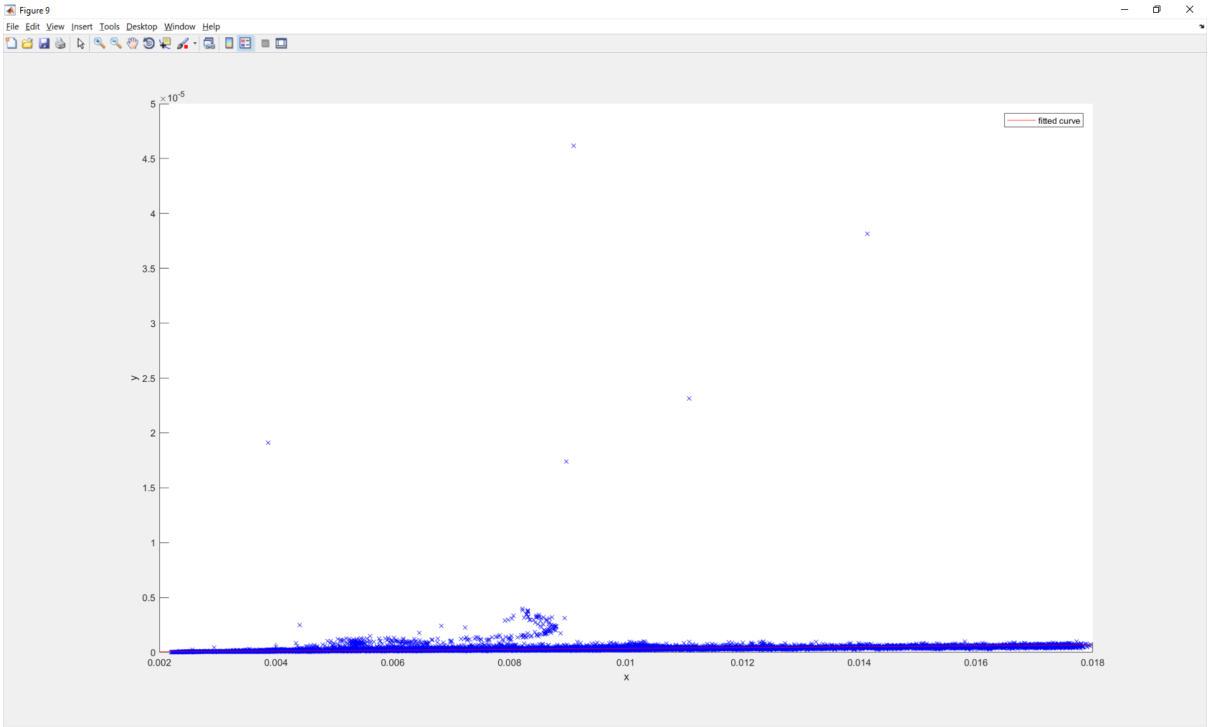


Figure 9
File Edit View Insert Tools Desktop Window Help
×10^{-5}
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
y
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
x
fitted curve

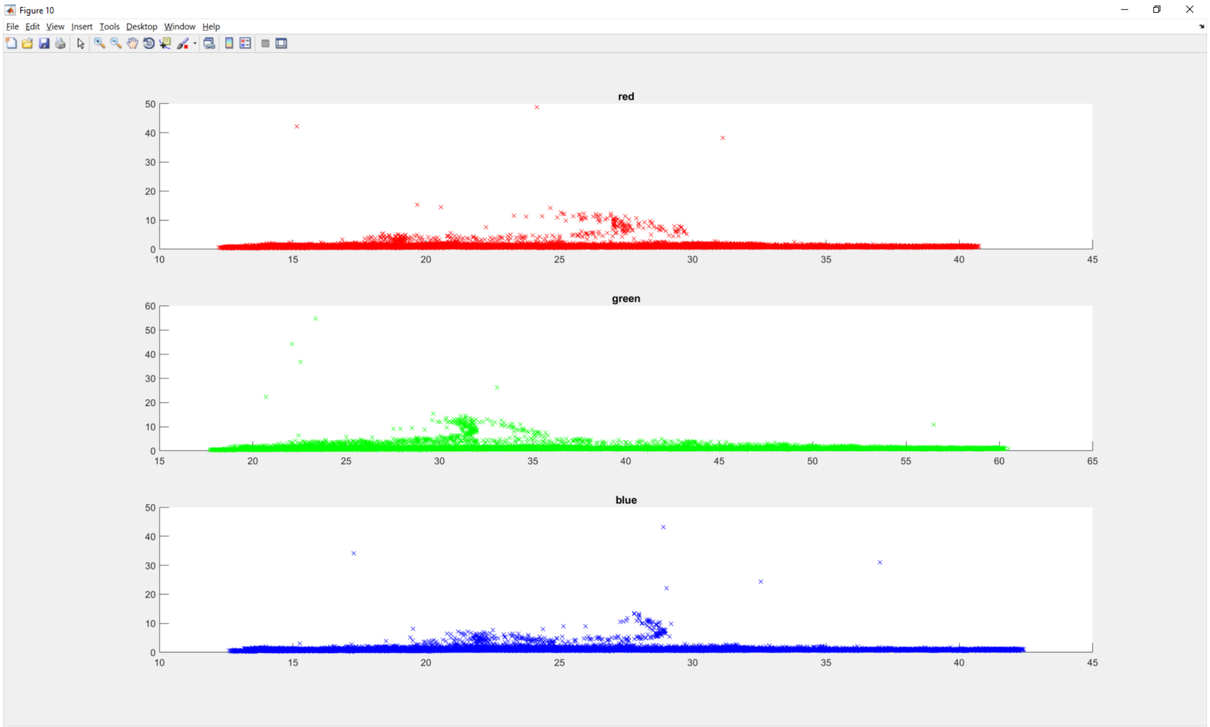


Figure 10
File Edit View Insert Tools Desktop Window Help
red
50
40
30
20
10
0
10
15
20
25
30
35
40
45
green
60
50
40
30
20
10
0
15
20
25
30
35
40
45
50
55
60
65
blue
50
40
30
20
10
0
10
15
20
25
30
35
40
45

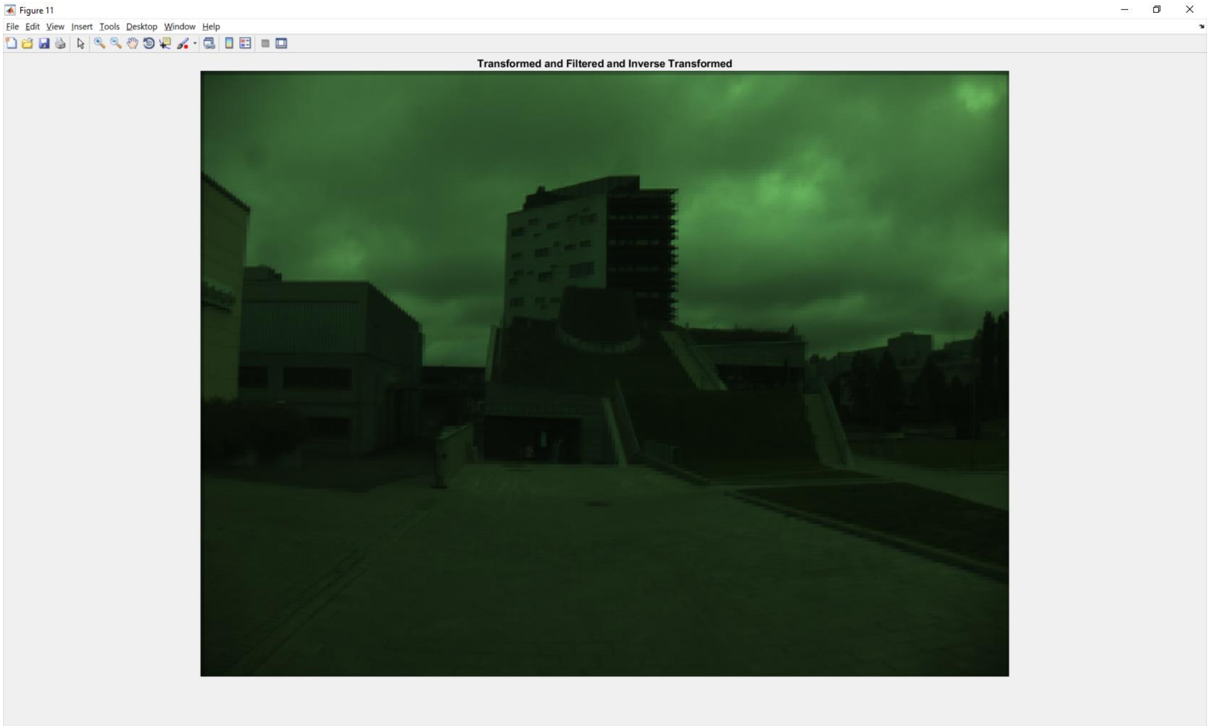


Figure 11
File Edit View Insert Tools Desktop Window Help
Transformed and Filtered and Inverse Transformed

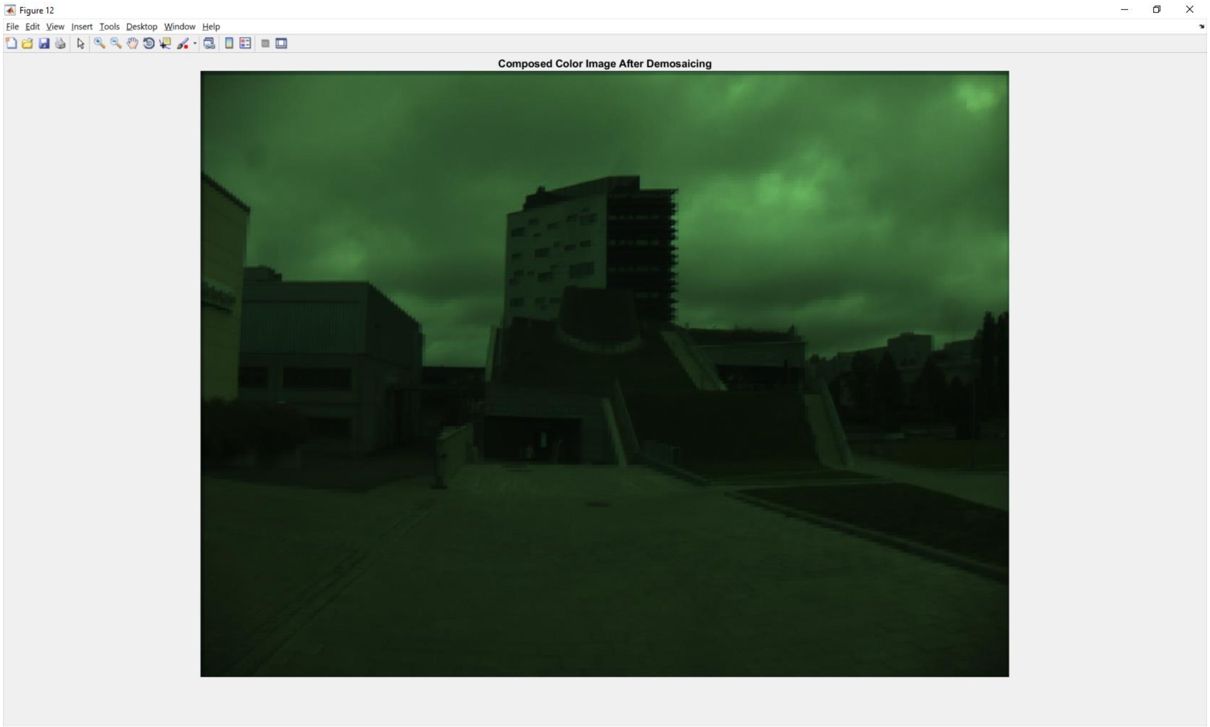


Figure 12
File Edit View Insert Tools Desktop Window Help
Composed Color Image After Demosaicing

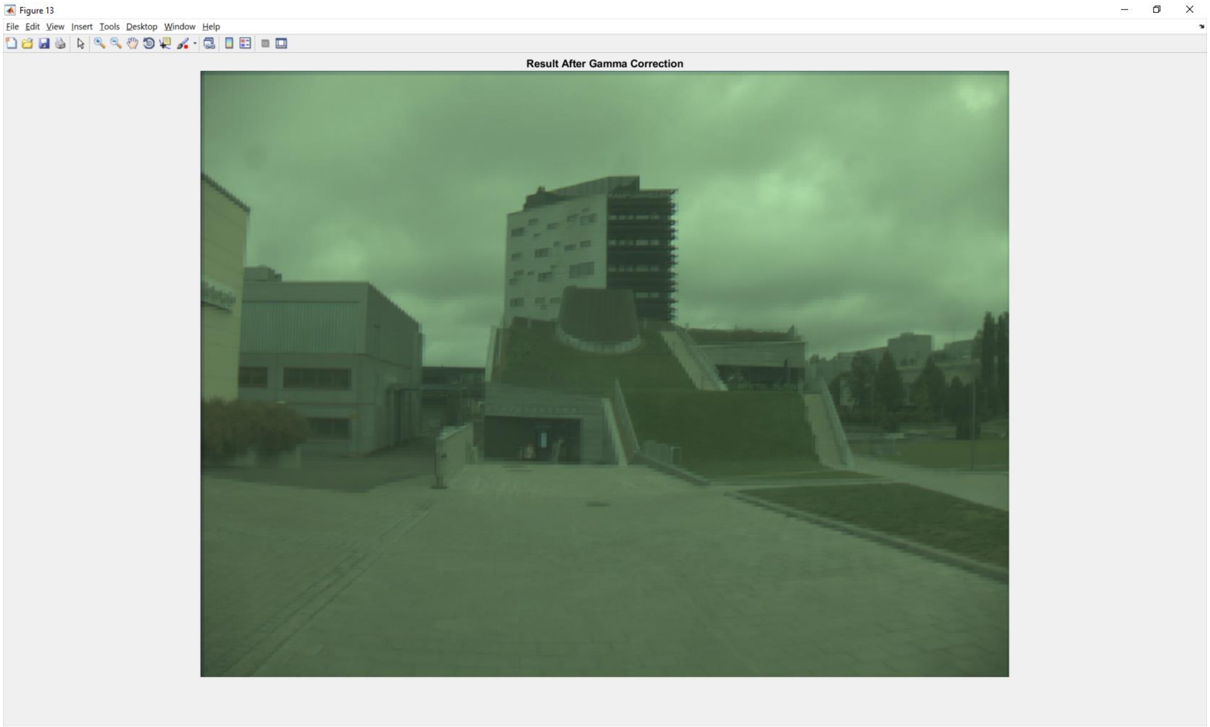


Figure 13
File Edit View Insert Tools Desktop Window Help
Result After Gamma Correction