**Project Title:  Image Processing Fundamentals**

**Project Number**: Project 1

**Course Number**: CEG 7850-01

**Student's Name**: Alex Reigle

**Date Due**: September 7, 2021

**Date Submitted**:  September 7, 2021

**Declaration Statement:**

I hereby declare that this Report and the Matlab codes were written/prepared entirely by me based on my own work, and I have not used any material from another Project at another department/ university/college anywhere else, including Wright State. I also declare that I did not seek or receive assistance from any other person and I did not help any other person to prepare their reports or code.  The report mentions explicitly all sources of information in the reference list. I am aware of the fact that violation of these clauses is regarded as cheating and can result in invalidation of the paper with zero grade. Cheating or attempted cheating or assistance in cheating is reportable to the appropriate authority and may result in the expulsion of the student, in accordance with the University and College Policies.

**Running the main.m file in the same folder as it is originally zipped (with the other files present) will allow for replication of the results presented.**

**Problem 1:** Reduce the Number of Intensity Levels in an Image.

1. Write a computer program to reduce the number of intensity levels in an image from 256 to 2, in integer powers of 2. The desired number of intensity levels should be a variable input to your program.

By running my MatLab function as follows:

*Ans =* *Project1\_problem1\_ReduceGreyScale(dripBottlePath, desiredNumberOfIntensityLevels)*

Where *dripBottlePath* is the entire file location of the image whose intensity levels are being changed and *desiredNumberOfIntensityLevels* is the argument defining the number of intensities returned in the *Ans* image.

For this problem, the original image of *drip-bottle-256.tif* is plotted using *imshow(image)* as follows:

A close-up of a light bulb

Description automatically generated with medium confidence

Figure 1: Original Drip Bottle image recreated and plotted in MatLab from provided drip bottle image

To solve this problem, I create a variable which is the ratio by which we scale the image relative to its original maximum intensity value. This equation is as follows:

*IntensityScalingVariable = origrinalMaxIntensity / newMaxIntensity*

I then multiply each element of the image by this scale factor to get the solution image below.

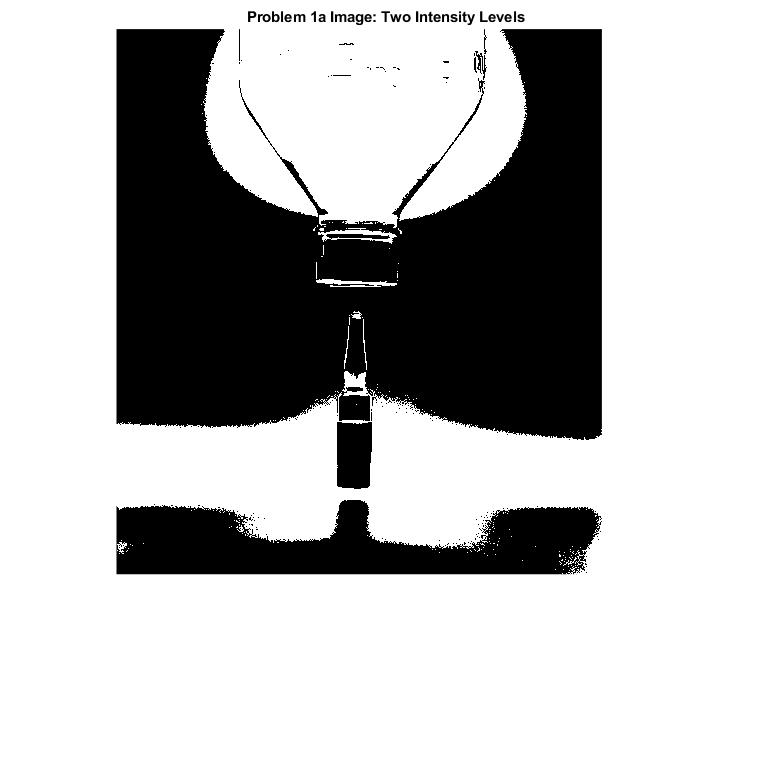


Figure 2: Drip Bottle image with the number of intensity levels reduced from 256 to 2.

1. For the drip-bottle Fig. 2.24(a) replicate the results shown in Fig. 2.24 of the textbook.

For problem 1b, the same function as described above was applied to the original drip bottle image. The values given to the argument were assigned as follows:

*desiredNumberOfIntensityLevels* = [ 256 182 64 32]

This results in the expected decrease in image quality as shown below:

Diagram

Description automatically generated

Figure 3: The drip bottle image with the corresponding a-d intensity levels.

However, the image quality change is much more obvious when the images are zoomed in on, as below:

A picture containing jack, electronics

Description automatically generated

Figure 4: The same images from Fig. 3 have been zoomed in on the center of the image to better show the reduction in the number of grey levels. This is most apparent in the background of Fig. 4(d).

**Problem 2: Zooming and Shrinking Images**

1. Download the Chronometer (‘Chronometer.tif’) image in Fig. 2.23(a) from Pilot and use your program to shrink the image by a factor of 4.

By resizing the image with the MatLab function imresize() as follows:

*Ans = imresize(chronometer, 1/ShrinkFactor)*

Where Shrink factor in this instance is 4. This results in the following image:

A picture containing watch, time, white, different

Description automatically generated

Figure 5: The Chronometer image file provided on Pilot after the number of pixels is reduced by a factor of 4.

A picture containing watch, different

Description automatically generated

Figure 6: Original Chronometer file.

To compare this result to the original image (Fig. 6) we must zoom into the images, which makes the difference obvious. This is shown below in Fig. 7.

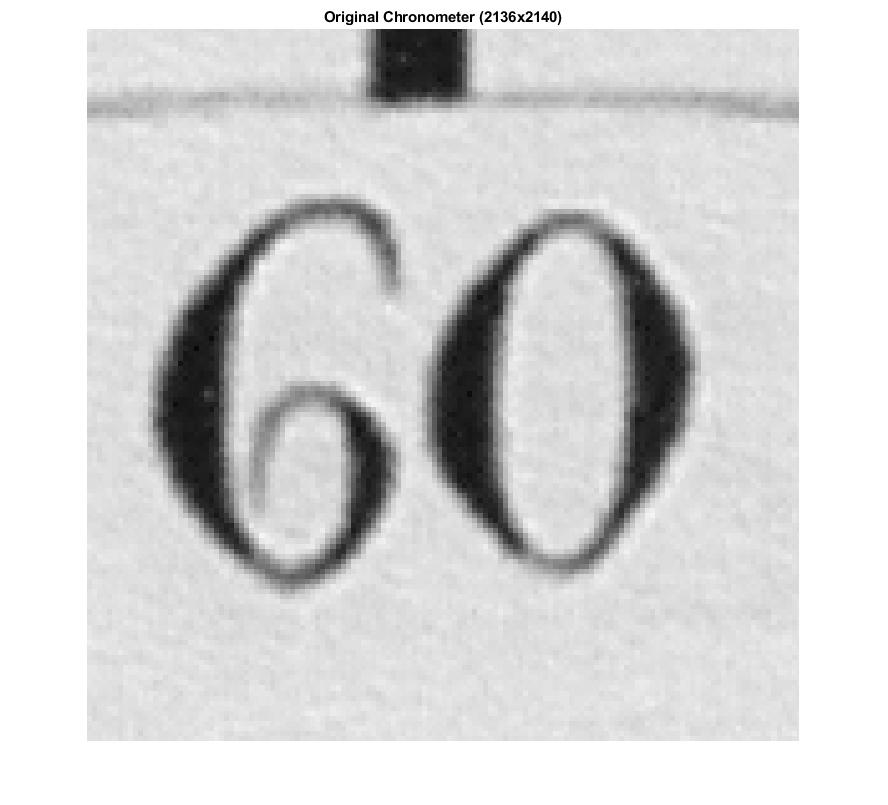
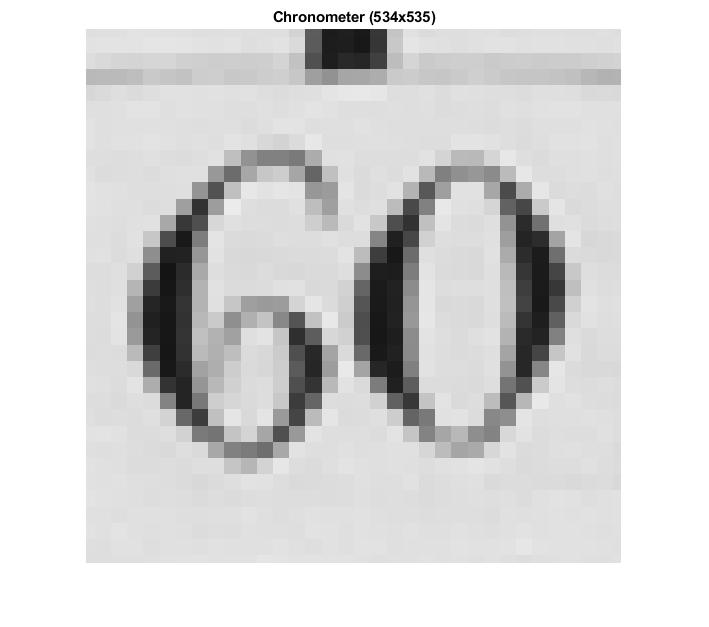


Figure 7: Zoomed in Fig 5 (left) next to zoomed in Fig. 6 (right).

1. Write a program to zoom the image in (a) back to the resolution of the original using “pixel replication”. Explain the reason for their differences.

The image in Fig. 9 shows the image quality is the same as the Chronometer in Fig. 5 with one- fourth of the original pixels. This is because we only copied the pixels that were already present and no effort to improve our guess of the value of the pixel.

A picture containing watch, different

Description automatically generated

Figure 8: Chronometer image after the number of pixels has been restored to the original number using pixel replication.

A picture containing icon

Description automatically generated

Figure 9: A zoomed in view of the Chronometer image in Fig. 8

1. Repeat part (a)-(b), but in this case use “bilinear transformation”. (Hint: MatLab commands: imresize or interp2)

By applying the *imresize(image, [size1 size2])* function in MatLab, with the specified size1 and size2 being the number of rows and columns of the Chronometer image after being reduced in size by a factor of four (Fig. 5) resulting in pixel (rows)x(columns) of (534)x(535). The image described can be found below in Fig. 10.

A picture containing watch, time, white, different

Description automatically generatedA picture containing text

Description automatically generated

Figure 10: The Chronometer image after being resized again to the size of (534)x(535) with the whole view of the image (left) and a zoomed in view (right).

After applying a bilinear transformation algorithm to the image in Fig. 10, the image quality improved is obvious. In this case, the algorithm used is the *interp2()* MatLab function suggested I n the problem description. The results are shown in Fig. 11, below.

A picture containing watch

Description automatically generatedA picture containing text

Description automatically generated

Figure 11: The results of a bilinear transformation on the image from Fig 10 showing the whole view of the image (left) and a zoomed in view (right).

In comparing Fig. 11 to the results in Fig. 9, it becomes obvious that the bilinear zooming is much more effective. The quality is improved due to the weighted estimated used in interp2() to estimate what the value of the new pixels (from zooming) instead of simple replicating them as is done in the pixel replication algorithm.