CMPEN/EE455: Digital Image Processing I

Computer Project # 1
Introduction to Digital Image Processing using MATLAB

Pavan Gurudath
Date: 09/08/2017

Objectives:

This project aims to familiarize certain concepts of digital image processing such as quantization, effects of spatial and gray-scale resolution, shrinking and zooming out a digital image. The main objectives of this project include the following:

- To downsample the spatial resolution of a given image to different sizes. Furthermore, to upsample the downsampled image using nearest-neighbor interpolation.
- To create an interpolated image using bilinear transformation.
- To reduce the gray-level quantization of the image by reducing the number of bits per pixel.
- To infer the changes compared with the original image when both spatial and gray-level resolution are changed.

METHODS

This project consists of several objectives. To tackle these, we have utilized a digital image named ***'walkbridge.tif'*** that is of the dimension [512 x 512 x 2]. The .tif image file consists of two layers, where the first layer consists of the necessary information while the second layer consists of white pixels throughout. To satisfy the objectives, we require the image that is present in the first layer. Therefore, the first layer is extracted and the subsequent set of processing steps are applied. The new image is of the dimension [512 x 512].

The following set of steps are carried out to complete the respective objectives.

1) To downsample the spatial resolution and further upsample back to original size using nearest-neighbor interpolation:

- After retrieving the first layer of ***'walkbridge.tif'***, the image is processed to downsample from 512x512 to 256x256, 128x128 and 32x32.
- For each of the downsample sized image, an image of their respective sizes is created that is filled with 0 intensity pixels.
- The original image is scanned entirely using temporary variables x and y.
- Using these variables and another set of x and y coordinate variables, the downsampled zero image is filled with intensity values from the 512x512 image.

In this way, the 512x512 image is downsampled to a 256x256, 128x128 and 32x32 image and are stored as Image256, Image128 and Image32 respectively. The images are converted to an unsigned 8bit integer form and are written to the working directory.

In order to upsample Image256, Image128 and Image32, to a 512*512 image, we perform the nearest-neighbor interpolation as follows:

- A blank image of 512*512 is created using the zeros inbuilt matlab function.
- Using the downsampled image, every pixel is replicated upto a certain factor such that after completing the replication, it forms a completed 512x512 image. The for loop runs until this factor in both, x and y direction, for every pixel.
- This factor for an upsampling from MxM to an NxN can be calculated as follows

$$\boldsymbol{factor = \frac{2^M}{2^N} - 1}$$

Thus, an upsampled image of size 512x512 is obtained from Image256, Image128 and Image32 and their resultant images are saved as ImageUpsampled256, ImageUpsampled128 and ImageUpsampled32 respectively.

2) To create an interpolated image using bilinear transformation:

To implement the bilinear interpolation, a file named ***'BilinearInterpolation.m'*** is called from the ***'main.m'*** file.

- Since a new file is being called, the main ***'walkbridge.tif'*** image is read again and the first layer is retrieved.
- The image is downsampled to 32x32 as explained in method 1 and is stored as Image32.

- Image32 is padded with zeroes in the 33rd row and column for the purpose of being able to compute the bilinear interpolation for last set of 16 rows and columns in the 512x512 image.
- A blank 512x512 image with zero intensity pixels is created.
- The pixel values of 32x32 image is placed at appropriate location in the 512x512 image i.e. every pixel from the 32x32 is accessed and is placed in the 512x512 by running the matrix in steps of 16 columns and 16 rows. In this way the 32*32 (1024) pixels are placed in the 512x512 image.
- The bilinear interpolation takes place in two steps i.e. first the algorithm runs in the x-direction and then in the y-direction.
- Using the coordinate convention of matlab, i.e. y-axis in the horizontal direction and x-axis in vertical direction, the bilinear interpolation is implemented first in the horizontal direction and then in the vertical direction.
- If the intermediate pixel is say x,y and the two known pixels in that row are $\mathbf{f}(\boldsymbol{x_1}, \boldsymbol{y_1})$ and $\mathbf{f}(\boldsymbol{x_1}, \boldsymbol{y_2})$, then **f(x,y)** is calculated as follows:

$$\mathbf{f} = \frac{(\boldsymbol{y_2} - \mathbf{y})}{(\boldsymbol{y_2} - \boldsymbol{y_1})}\mathbf{f}(\boldsymbol{x_1}, \boldsymbol{y_1}) + \frac{(\boldsymbol{y} - \boldsymbol{y_1})}{(\boldsymbol{y_2} - \boldsymbol{y_1})}\mathbf{f}(\boldsymbol{x_1}, \boldsymbol{y_2})$$

- Now all the 16th multiple rows are filled with intensity values throughout the entire range of columns. This matrix is stored as some temporary image ***'a'***.
- With the help of temporary image ***'a'***, we compute the intensities of all the pixels in each row using the two known pixels in that row which are computed from the previous step. Let the two known pixels be $\mathbf{f}(\boldsymbol{x_1}, \boldsymbol{y_1})$ and $\mathbf{f}(\boldsymbol{x_2}, \boldsymbol{y_1})$, then the intensity of the unknown pixel $\mathbf{f}(\boldsymbol{x_0}, \boldsymbol{y_0})$ is calculated as follows:

$$\boldsymbol{f_0} = \frac{(\boldsymbol{x_2} - \mathbf{x})}{(\boldsymbol{x_2} - \boldsymbol{x_1})}\mathbf{f}(\boldsymbol{x_1}, \boldsymbol{y_1}) + \frac{(\boldsymbol{x} - \boldsymbol{x_1})}{(\boldsymbol{x} - \boldsymbol{x_1})}\mathbf{f}(\boldsymbol{x_2}, \boldsymbol{y_1})$$

- This image obtained thus obtained is stored with the pixel intensity values of only the 512 rows and columns.

  In this way, the bilinear interpolation is done and the image is stored as finalImageUpsampled512.

3) **To change the gray-level quantization by reducing the number of bits per pixel from 8 to 7, 6, 5, 4, 3, 2 and 1:**
   To achieve this objective, we create a function named ***'nBitPlane'*** that is called from the main function. This function does the necessary computations to reduce the number of bits per pixel.

- For the n-bit per pixel, we call the function and pass the parameters of the original 512x512 image along with the value of n that it needs to compute.
- If the input image is '**I**' and the output image is '**ImageSliced**', then the formulae to compute a pixel that would reduce the number of bits would be

$$\boldsymbol{ImageSliced}(\boldsymbol{x}, \boldsymbol{y}) = \left(\boldsymbol{round}\left(\frac{\boldsymbol{I}(\boldsymbol{x}, \boldsymbol{y})}{\mathbf{2}^{\boldsymbol{n}}}\right)\right) \times \mathbf{2}^{\boldsymbol{n}} \qquad \forall\, \boldsymbol{n} \in [\mathbf{1}, \mathbf{7}]$$

In this way, the output image ***'ImageSliced'***, is computed.

4) **To change both the spatial resolution and the gray-level resolution and observe the changes compared with the original image:**
   In order to achieve this objective, we follow the procedure followed in objective 1 followed by objective 3.

- The image ***'walkbridge.tif'*** is downsampled from a 512x512 image to a 256x256 image.
- From the 256x256 image, the image gray levels are restricted to 6bits per pixel using the procedure mentioned in objective 3.
- The output image is then written into the working directory.

RESULTS

This section details all the images that are obtained after completing each of the objective. Firstly, the image that is considered as the input image for all the objectives i.e. ‘walkbridge.tif’ is as shown in Fig 1.

Figure 1: Input Image - 'Walkbridge.tif'

This image consists of two layers, where the first layer contains the information while the second layer consists of white pixels. The size of the image is 512x512. The two layers are as represented in Fig 2 and Fig 3.

Figure 2: First layer of input image

Please note that the second layer of the input image as shown in Figure 3 has been edited with a border in order to indicate the white pixel image.

Figure 3: Second layer of input image

Therefore the image shown in figure 2 is utilized to process and thereby satisfy the objectives. The output images shown in figure 4, 5 and 6 represent the downsampled images of size 256x256, 128x128 and 32x32 respectively.

1. 256x256

Figure 4: Downsampled image from 512x512 to 256x256

2. 128x128

Figure 5: Downsampled image from 512x512 to 128x128

3. 32x32

Please note that Figure 6 is 32x32 image. It has been enlarged to a size 2x2 inch to be visible on the screen.

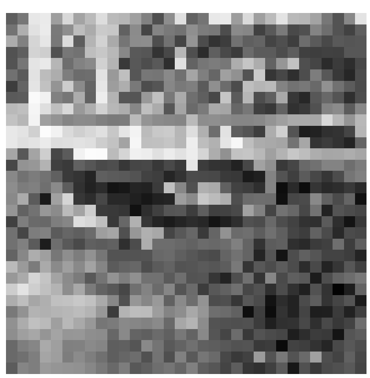

Figure 6: Downsampled image from 512x512 to 32x32

The final outcome of objective 1 is to upsample the downsampled images shown in Figure 4, Figure 5 and Figure 6 to a 512x512 image. The output images are as shown.

Figure 7 represents a 512x512 image that was downsampled to 256x256 and then upsampled to 512x512 image.

Figure 7: Image upsampled from 256x256 to 512x512

Figure 8 represents a 512x512 image that was downsampled to 128x128 and then upsampled to 512x512 image.

Figure 8: Image upsampled from 128x128 to 512x512

Figure 9 represents a 512x512 image that was downsampled to 32x32 and then upsampled to 512x512 image.

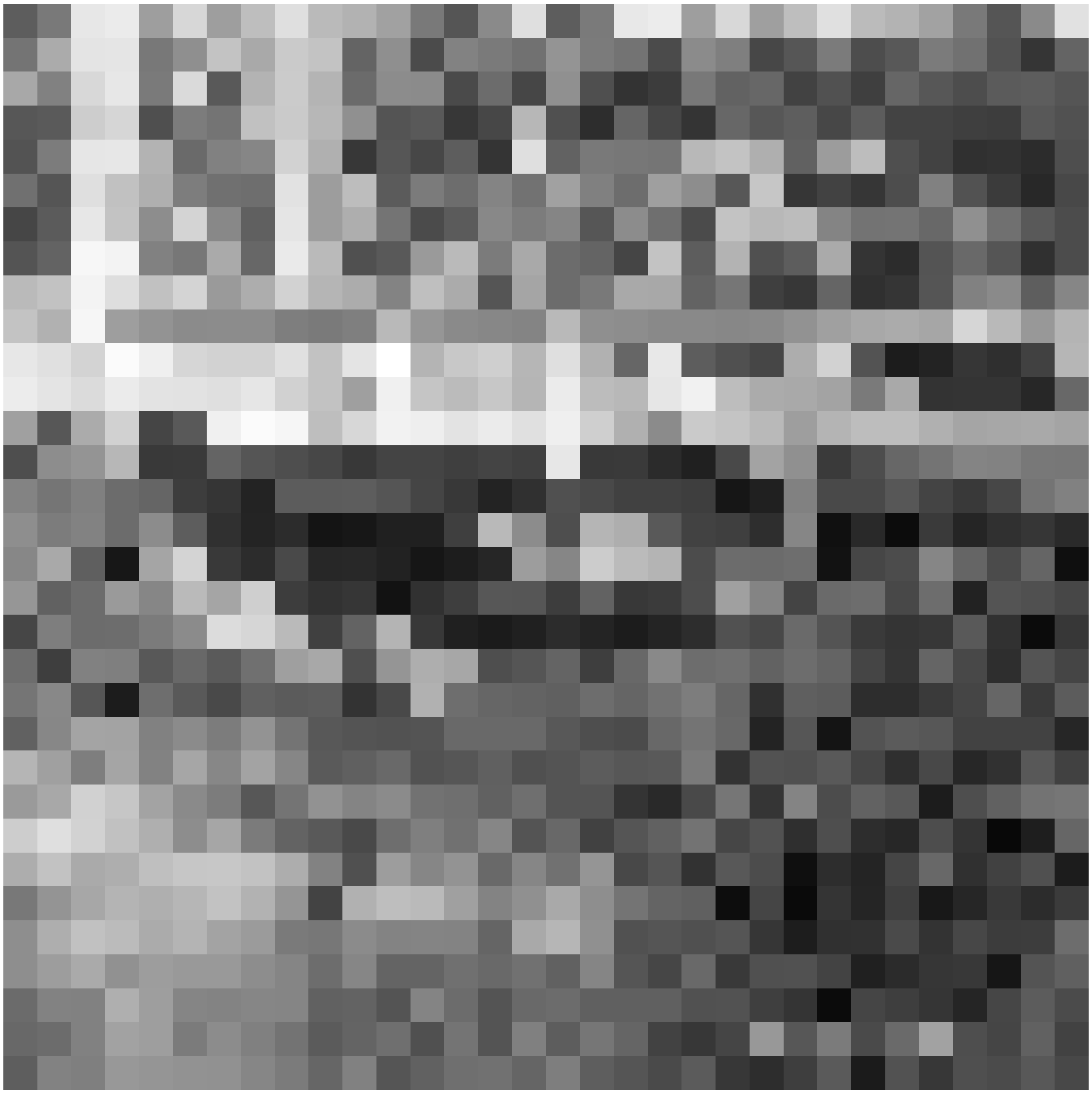

Figure 9: Image upsampled from 32x32 to 512x512

The following observations are made after completing the first objective:

- We notice that the quality of our input image reduced while downsampling the image to other sizes.
- The reduction in quality of the downsampled image was seen more as we reduced the size of the pixel, i.e. the quality of 32x32 downsampled image was worse than 256x256 image. This is due to the omission of pixel intensity values as we reduce size.
- It is noticed that upsampling an image using nearest neighbor interpolation allows us to achieve the size of the image required, however the image quality is not maintained.

The second objective required us to interpolate the downsampled 32x32 image as shown in figure 6 to a 512x512 image using bilinear interpolation. The output image is as shown.

Figure 10: Bilinear Interpolated image from 32x32 to 512x512

We observe that the upsampling from a 32x32 image to a 512x512 image using bilinear interpolation produces a better result in terms of the quality of the image perceived by human eye as compared to nearest neighbor interpolation.

The third objective of this project was to change the gray-level quantization of the original 512x512 image by reducing the number of bits per pixel from 8 to 7, 6, 5, 4, 3, 2 and 1 bits/pixel. The output images are shown from figure 11 to figure 17 respectively.

Figure 11: 7 bits per pixel gray level quantization

Figure 12: 6 bits per pixel gray level quantization

Figure 13: 5 bits per pixel gray level quantization

Figure 14: 4 bits per pixel gray level quantization

Figure 15: 3 bits per pixel gray level quantization


Figure 16: 2 bits per pixel gray level quantization

Figure 17: 1 bit per pixel gray level quantization

The following table illustrates the observations that are made from the set of images obtained.

Number of bits per pixel	Number of intensity levels	Range of intensity
8	256	[0, 1, 2,…, 255]
7	128	[0,2,4, 6,…, 254]
6	64	[0,4,8,12,…,252]
5	32	[0,8, 16,…, 248]
4	16	[0,16, 32,..., 240]
3	8	[0, 32, 64,…, 224]
2	4	[0, 64, 128, 192]
1	2	[0, 128]

From the images and the table above, we observe that as the number of bits per pixel decreases, the number of gray level intensities available to us also decreases. Thereby, it decreases the smoothness of the original image.

Lastly, the outcome of our final objective is seen in two steps. The first step involves the spatial resolution from a 512x512 image to a 256x256 image which is shown in figure 18. The second step involves the gray scale resolution to 6bits per pixel as shown in figure 19.

Figure 18: Spatial resolution from 512x512 to 256x256

Figure 19: Spatial resolution to 256x256 and gray scale resolution of 6bits per pixel

It is observed that only a minor degradation in image quality can be perceived by human eye.

CONCLUSION:

The outcomes of the images after having completed each objective, the following conclusions can be made:

- Downsampling an image affects the quality of the image as we reduce the number of pixels. However, the quality of a downsampled image by a small factor (relative to the size of the image) is not distinguishable by human eye when compared with the original image.
- The change in image quality depends upon the application for which the donwsampled image is being used.
- Bilinear interpolation is better than nearest neighbor interpolation.
- No perceivable effects can be seen when gray scale quantization is done from 8bits to 5bits per pixel. However, as the number of bits decrease from 4 to 1, there is degradation of image quality.