BBM 415 - Image Processing Practicum
Problem Set 1

Didar Tüfekçigil 21827938

November 14, 2021

1 Part 1: Dithering

1.1 Quantization

In the first part, quantization algorithm was applied to the example pictures. Quantization of images is achieved by applying a simple rounding operation with a constant, which was "q" in this case.

Algorithm 1 Quantization Algorithm

for pixel in image **do**
    $pixel \leftarrow round(pixel/q) * q$

In fig.1 and fig.2, the results of quantization can be seen clearly. As quantization factor "q" increases, the discretization of images increases while the resolution decreases.

1.2 Quantization with different values

In fig.1 and fig.2, different values of q has been applied to the images which produce images with lesser color, and therefore, reduces resolution of the image.

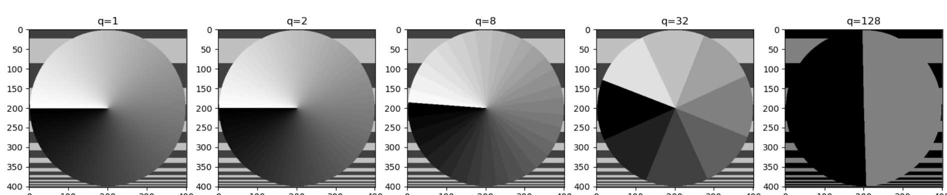

Figure 1: Quantization with different q parameters

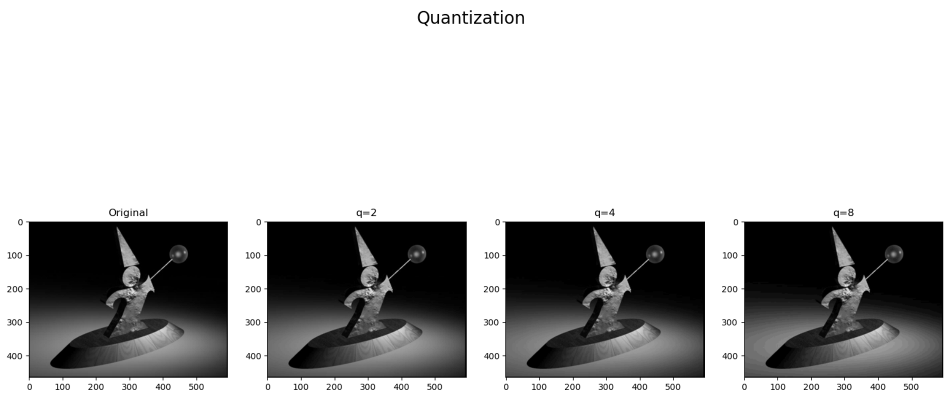

Figure 2: Quantization with different q parameters

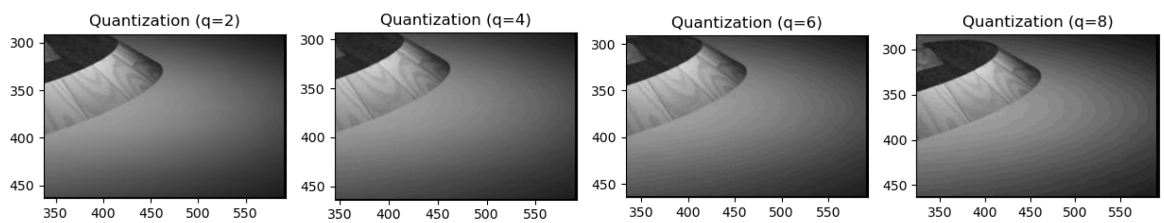

Figure 3: A closer look at fig.2

1.3 Dithering

Dithering is a type of image processing that gives the appearance of color depth in images with a limited color palette, as a result, decreases the effect of quantization. In this particular assignment, Floyt-Steinberg dithering is used with the following algorithm:

Algorithm 2 Floyd-Steinberg Dithering Algorithm

for each y from top to bottom **do**
  for each x from left to right **do**
    $oldpixel \leftarrow pixels[x][y]$
    $newpixel \leftarrow find\_quantized\_value(oldpixel)$
    $pixels[x][y] \leftarrow newpixel$
    $quant\_error \leftarrow oldpixel - newpixel$
    $pixels[x+1][y] \leftarrow pixels[x+1][y] + quant\_error 7/16$
    $pixels[x-1][y+1] \leftarrow pixels[x-1][y+1] + quant\_error 3/16$
    $pixels[x][y+1] \leftarrow pixels[x][y+1] + quant\_error 5/16$
    $pixels[x+1][y+1] \leftarrow pixels[x+1][y+1] + quant\_error 1/16$

To find the quantized value of the old pixel with aim of diffusion the error that is caused by it, below algorithm is used:

Algorithm 3 Find quantized value

$quantized\_val \leftarrow round(old\_val(q1))(q1)$
$\boldsymbol{return} quantized\_val$ =0

1.4 How does it work?

Floyd-Steinberg dithering is based upon the diffusion of the error of neighbor pixels. It adds some of the residual quantization error of a particular pixel onto its neighbor pixels by multiplying it with predefined constants. If it finds a pixel of 96 gray, it determines that 96 is closer to 0 than to 255 - and so it makes the pixel black. Therefore, the resultant pixel values create an illusion to the human eyes which can be commented as reducing the effect of quantization.

Given different values of q values, as it increases, the resolution of the image increases. With small values of q, algorithm produces visibly spotted or stippled appearance (fig.4 and fig.5, first images), while high values of q produce smoother color distributions. Spotted appearance results from working with a small number of available colors.

The main drawback of Floyd-Steinberg dithering is that it is inherently serial. A simultaneous error diffusion may result in faster and improved algorithm for better resolution with decreasing the computational cost.

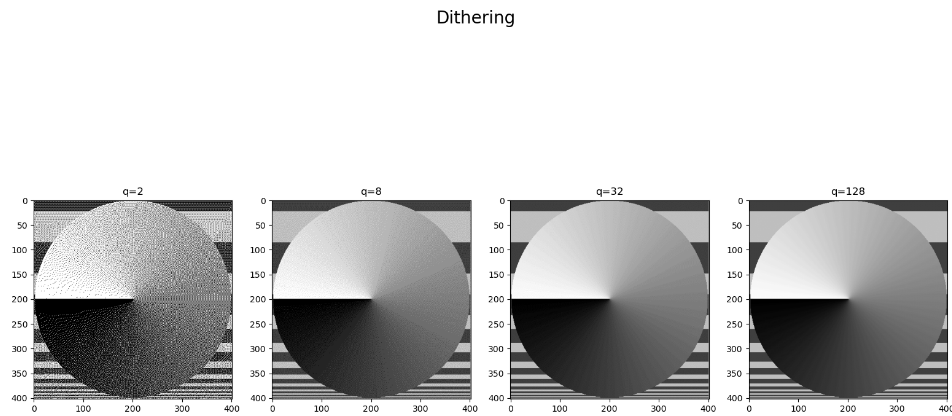

Figure 4: Dithering with 4 different values: 2, 8, 32, 128

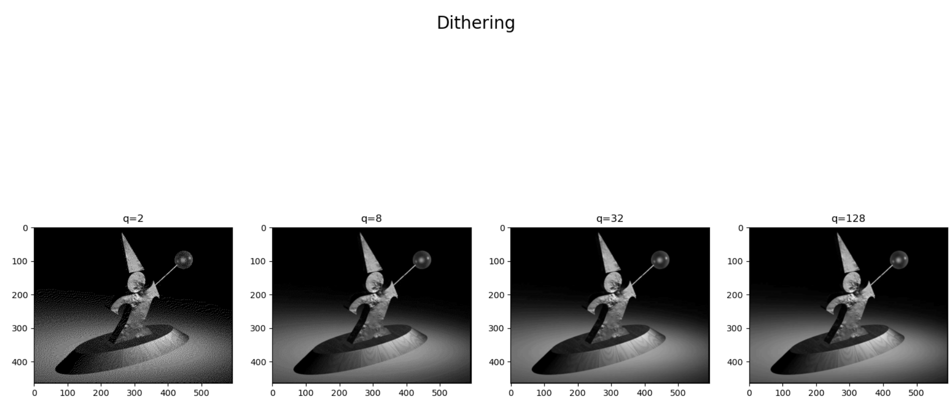

Figure 5: Dithering with 4 different values: 2, 8, 32, 128

In the first resultant image part in fig.4 and fig.5, a stippled appearance stands out because of the small values of q. As the q increases, a visible improvement can be noticed.

2 Part 2: Color Transfer

Changing the color of an image is one of the most popular image processing tasks. Often, this entails removing a strong and undesirable color cast from an image. In this part, a set of transformation operations are applied to the reference images to take color attributes from one image and apply them to another.

2.1 Why LAB color space conversion?

LAB color is intended to be as close to human vision as possible and composes of 3 channels: one is dedicated to Luminosity, the other two are dedicated to the color information. Converting images from RGB to LAB color space reduces colors which results in the algorithm to run faster. In addition to that, having a separate Luminosity channel lets us change the brightness of the image without actually causing an effect on colors. However, these operations on images using RGB are difficult to apply.

2.2 Drawbacks

Algorithm gives the expected results when the reference images's color distributions are not so similar. In fig.17, two flower photos were used as reference images which have similar color distributions in terms of RGB values. The resultant image (fig.18) shows an unpleasant appearance because of the similarity in composition between the source and target images.

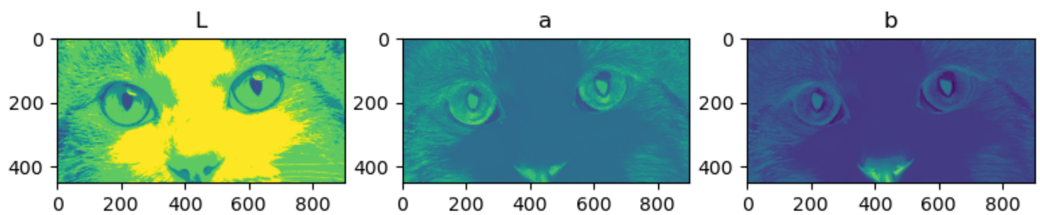

Figure 6: Lab channels of the cat image (fig.15): Luminosity, a and b respectively.

Figure 7: Images to apply color transfer to (source image in left, target image in right

Figure 8: Color transferred image (src: Scotland House, tgt: Scotland plain)

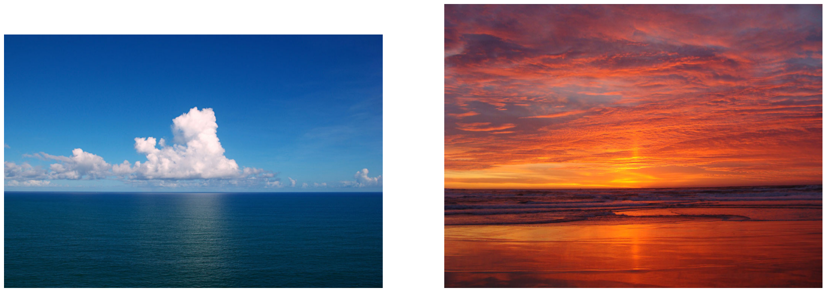

Figure 9: Images to apply color transfer to (source image in left, target image in right

Figure 10: Color transferred image (src: Ocean day, tgt: Ocean Sunset)

Figure 11: Images to apply color transfer to (source image in left, target image in right

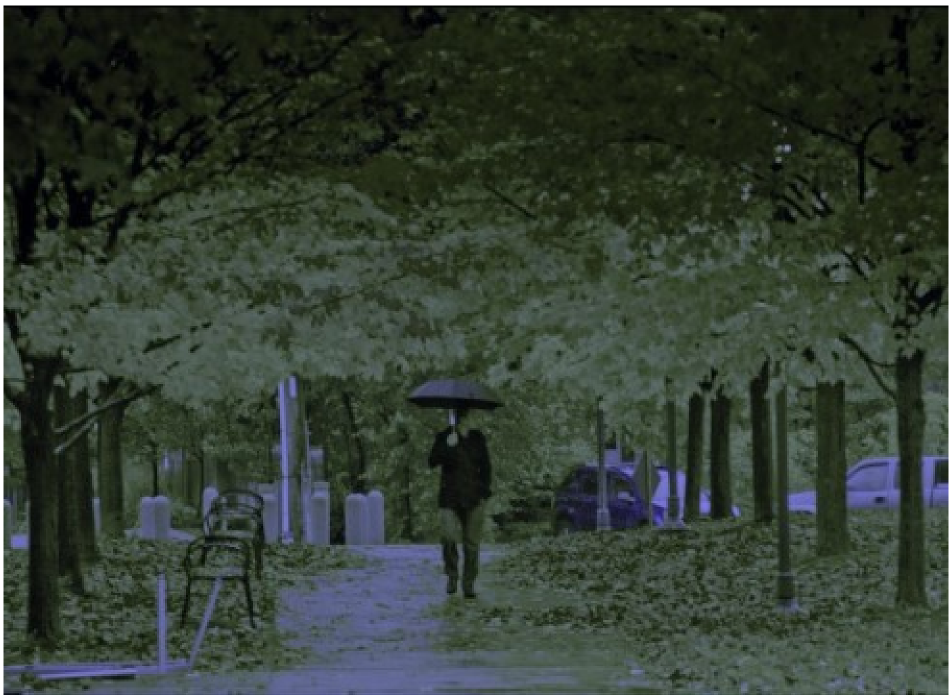

Figure 12: Color transferred image (src: Autumn, tgt: Woods)

Figure 13: Images to apply color transfer to (source image in left, target image in right

Figure 14: Color transferred image (src: Tree, tgt: Dark Night)

Reference images with contrast rgb color scales such as in fig.13 generates great results (fig.14) by the algorithm.

Figure 15: Images to apply color transfer to (source image in left, target image in right

Figure 16: Color transferred image (src: Cat, tgt: Space)

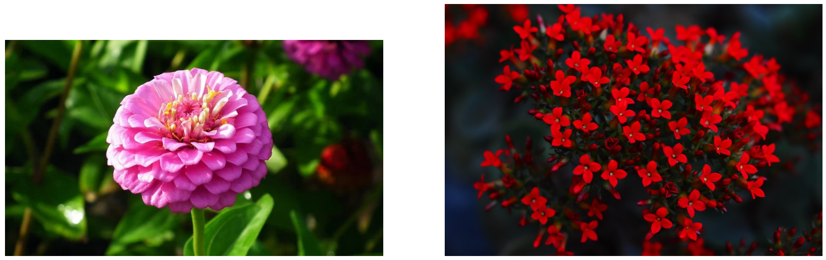

Figure 17: Images to apply color transfer to (source image in left, target image in right

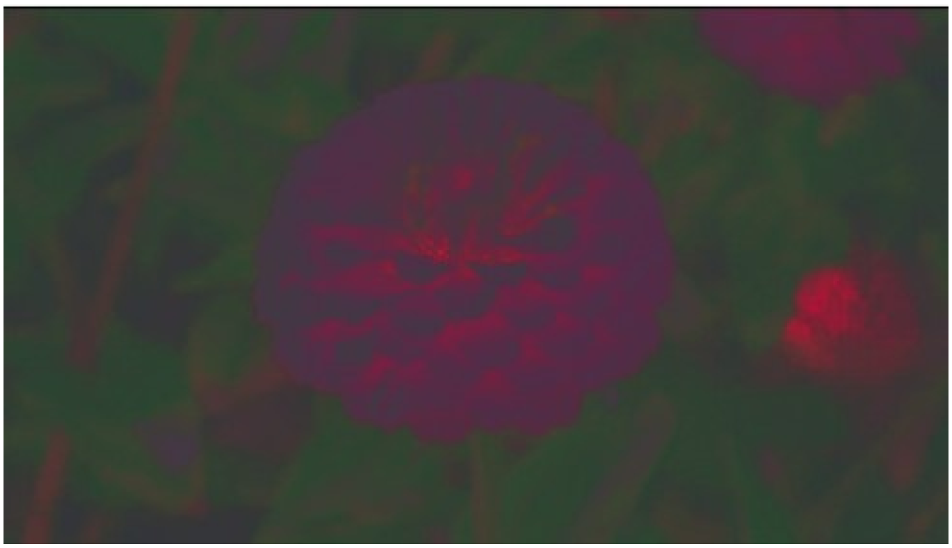

Figure 18: Color transferred image (src: Pink Flowers, tgt: Red Flowers)