BBM 413

ASSIGNMENT I

Name: Abdullah Atahan

Surname: Türk

Student ID: 21827943

Topics:

Part 1: Dithering

Part 2: Colour Transfer

T.A: Yunus Can Bilge

PART I: DITHERING

How is the given method achieves to prevent quantization error? Explain with examples.

In quantization, we aim to reduce the number of colors in the color palette of the image. But because of “quantization error”, we may not get as good results as we would like without dithering.

Dithering is an intentionally applied form of noise used to randomize quantization error, preventing large-scale patterns such as color banding in images.

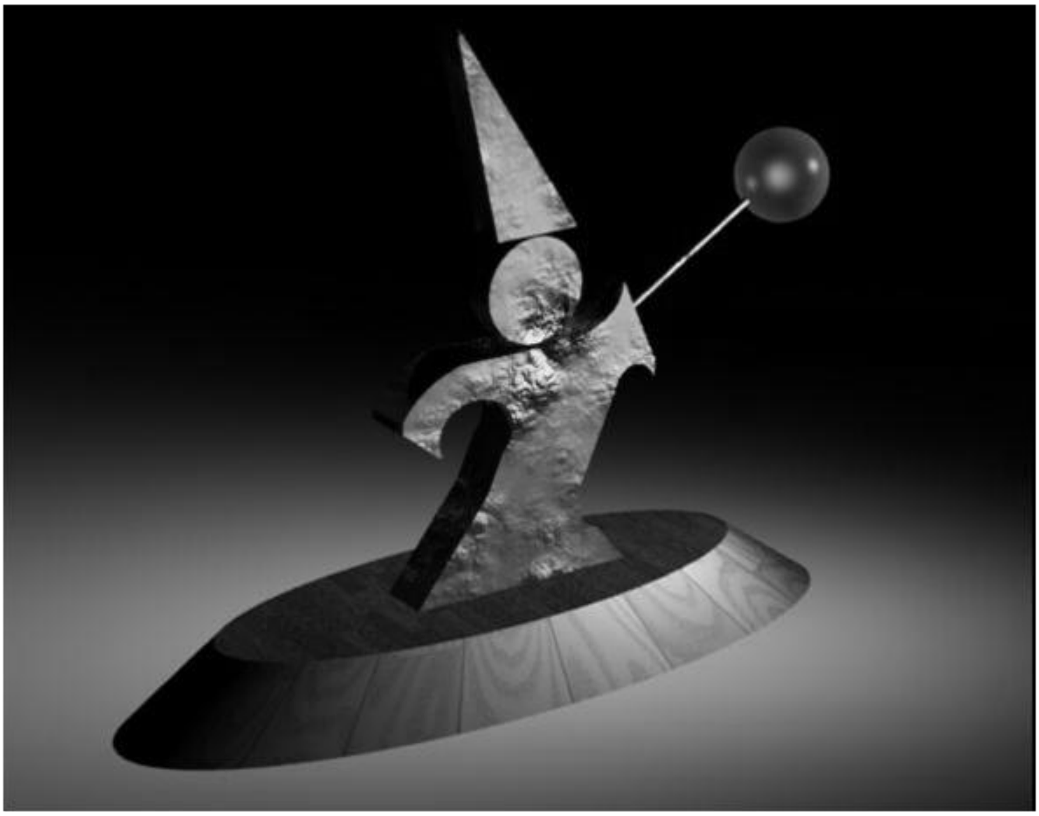
Thanks to “dithering”, we prevent the concentration of pixels with certain color intensity in certain regions to a large extent. In dithering, we calculate the quantization error by using the unquantized and quantized image of the image.

What is the behavior of the algorithm for different q parameters?

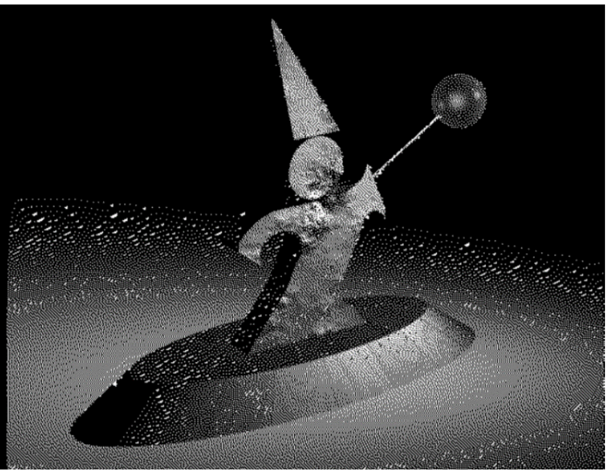
In our algorithm, the q parameter determines how many different color tones the pixels in the picture we will obtain can have. As the q parameter gets larger, the color variety in our image also increases. Since we use gray-scale images in our project, the higher q value means more shades of gray we can use. For example if we take the value of q as 1, there will be just two tones of gray in the picture which is 0 and 255 (Just black and White). We can increase this number of tones by taking larger q values.

Compare the quantized image and dithered image for different q parameters.

We will examine the difference between the original painting and the dithering and quantized images with 2 different images with 5 different q values. We will see in the examples that as the q value increases, we will get a better picture. Also as the q value increases, the difference between quantized and dithered versions of the image will decrease. In addition, as the number of tones in the pictures increases as the q value increases, the depth felt by the picture also increases. Furthermore as q increases, we will see that the noise in the pictures decreases.



Original Image

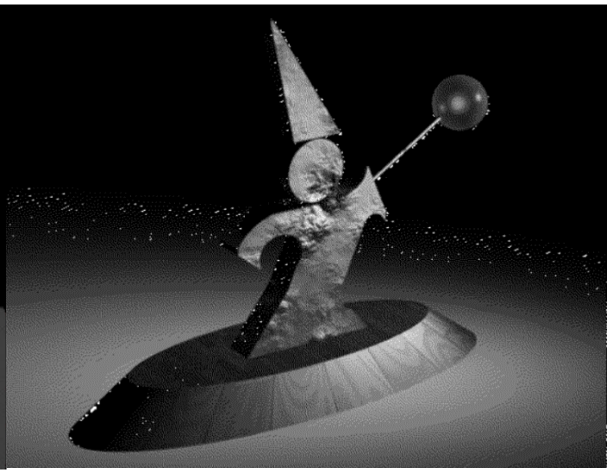


Quantized Image (q = 1) **Dithered Image (q = 1)**

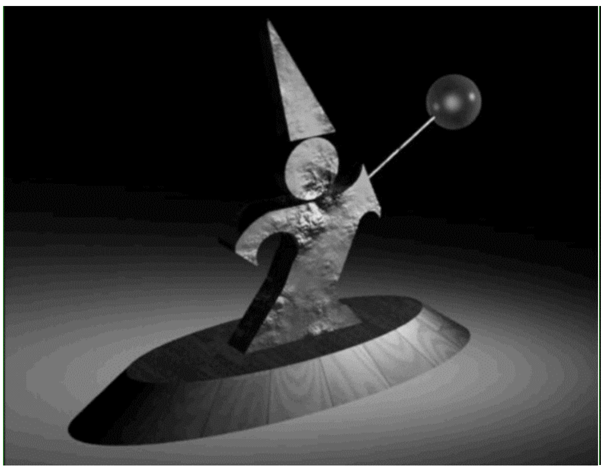
Quantized Image (q = 2)

Dithered Image (q = 2)

Quantized Image (q = 4)

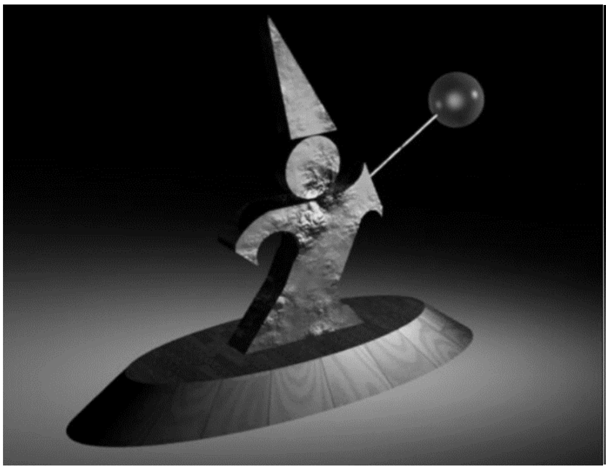


Dıthered Image (q = 4)

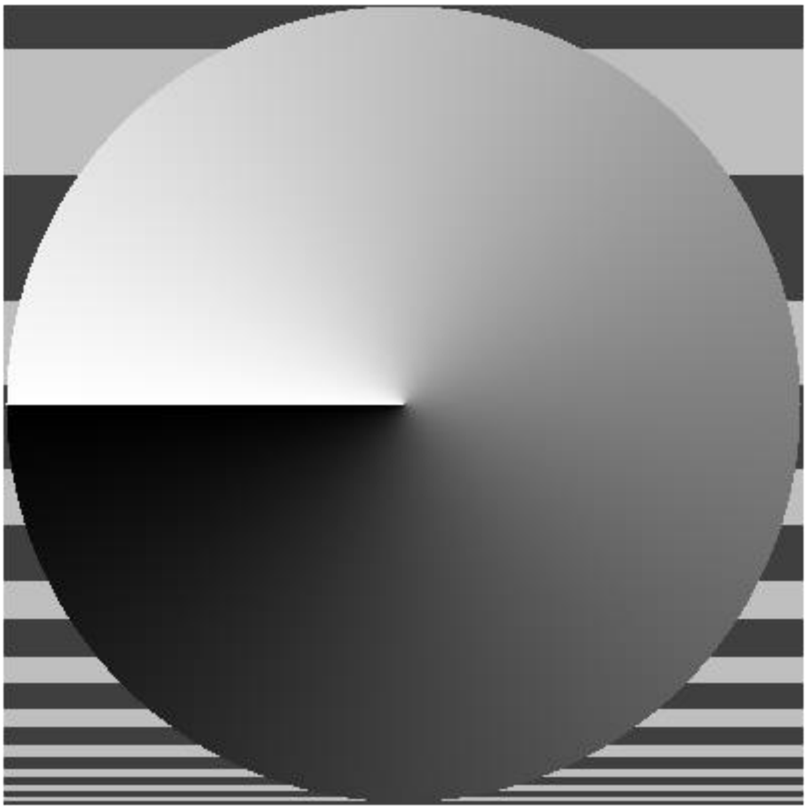


Quantized Image (q=32)

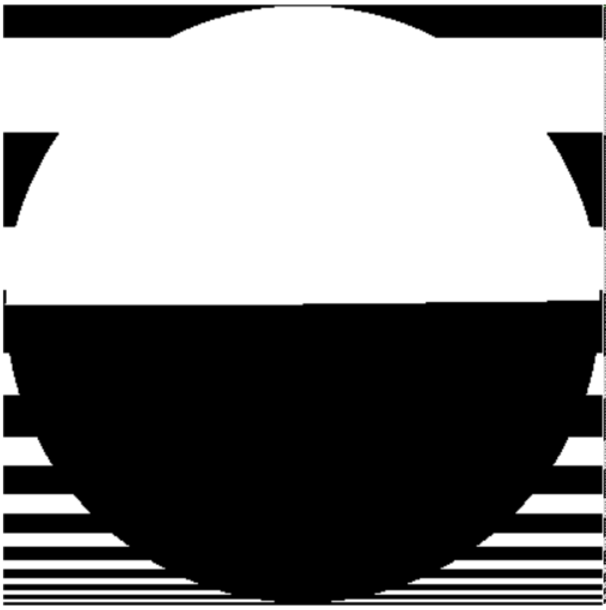
Dithered Image (q = 32)



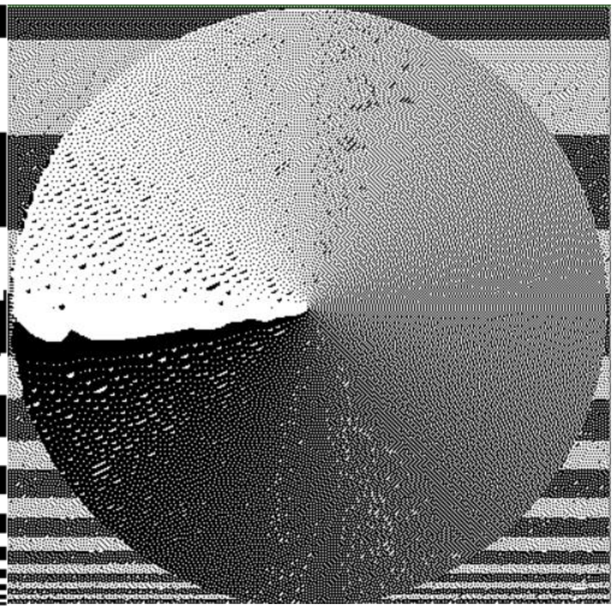

Quantized Image (q = 128) Dithered Image (q = 128)



Original Image

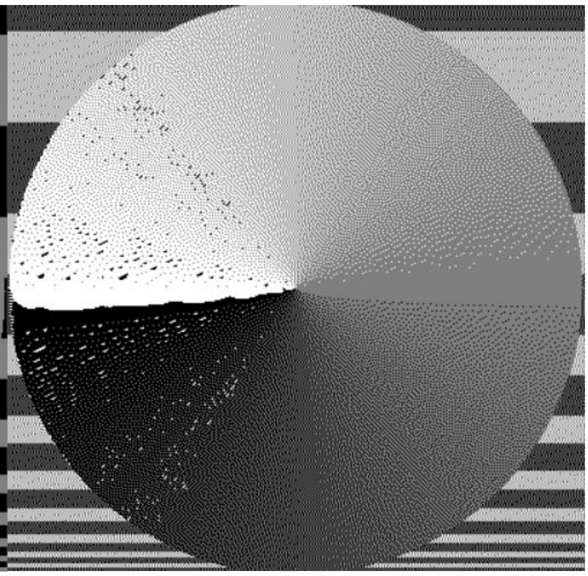


Quantized Image (q = 1)



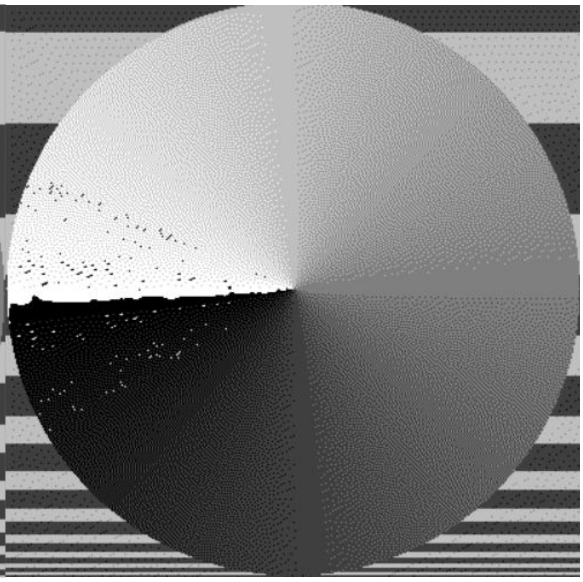
Dithered Image (q = 1)

Quantized Image (q = 2)

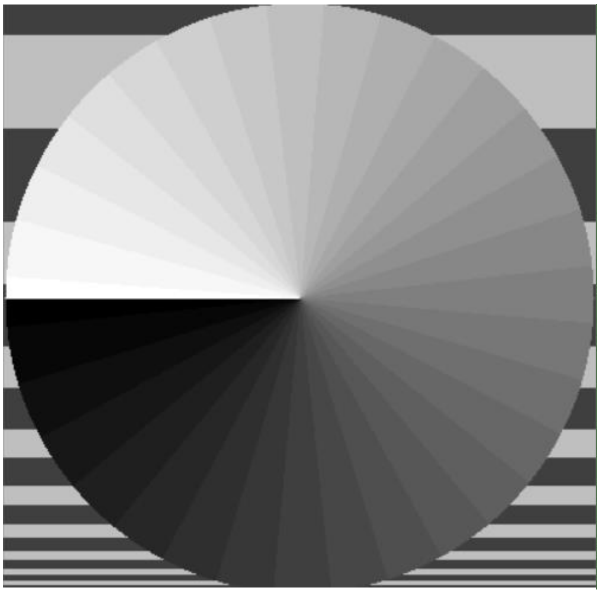


Dithered Image (q = 2)

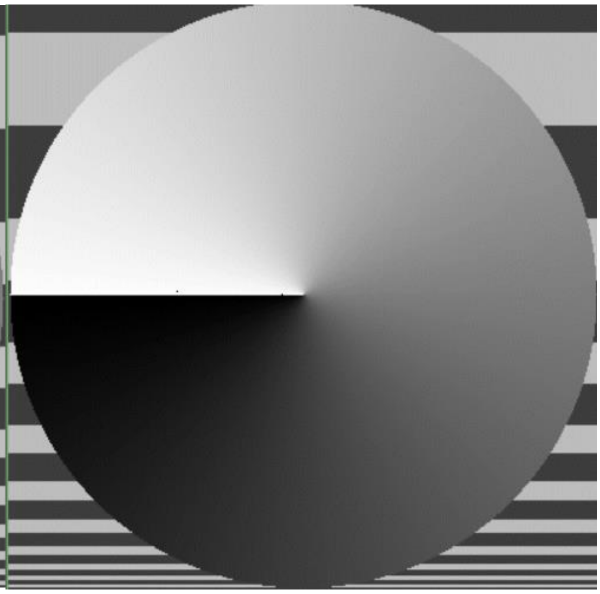
Quantized Image (q = 4)



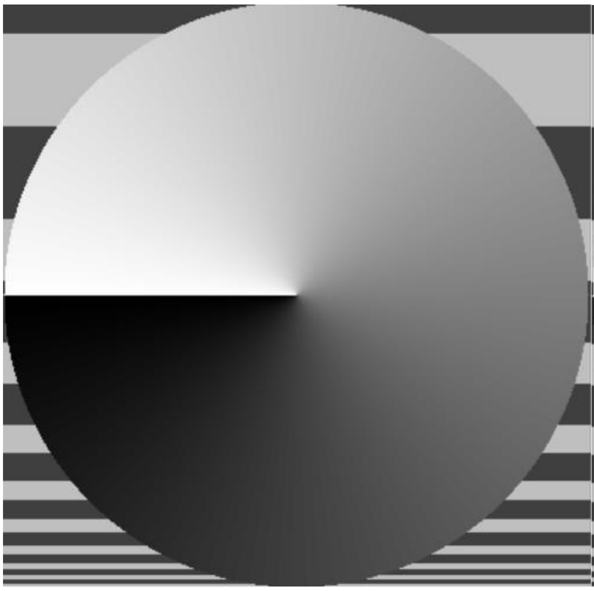
Dithered Image (q = 4)

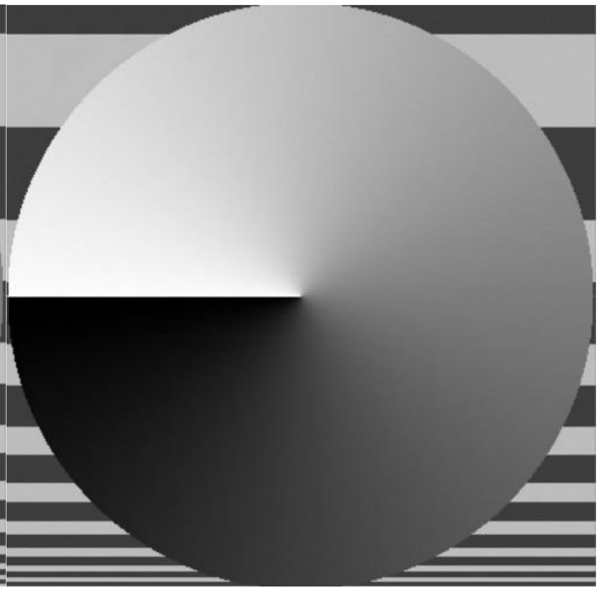


Quantized Image (q = 32)



Dithered Image (q = 32)





Quantized Image (q = 128) **Dithered Image (q = 128)**

What are the disadvantages of Floyd-Steinberg dithering algorithm? Explain with examples.

Altough Floyd-Steinberg algorithm is one of the earliests for dithering, it is still one of the most popular algorihtms. This algorithm uses “Erros-diffusion” technique. FS algorithm only diffuses the error to neighboring pixels. This results in very fine-grained dithering. But of course as with everything else, this algorithm also has several disadvantages.

This algorithm sometimes produces clearly identifiable visually harmful artifacts in highlights and in dark areas, sometimes referred to as worm artifacts. Also at certain intensity levels, patches of regular structure may appear. Not only are such patches visually disturbing, but also the uneven transition between “structured” and “unstructured” areas may be clearly visible and undesirable We can see those drawbacks in my examples.

Furthermore, some images have a sense of depth that creates the illusion of 3rd dimension. When we use F-S algorithm with small q parameter, this sense will disappear. Especially in the second example, we can clearly see that.

And lastly, sometimes adding noise to picture can bring some problems. For exampe in x-ray or MR images this can cause big problems. Because of the noise some details in the image may dissappear.

PART II: COLOR TRANSFER

Why does the algorithm change color space from RGB to Lab?

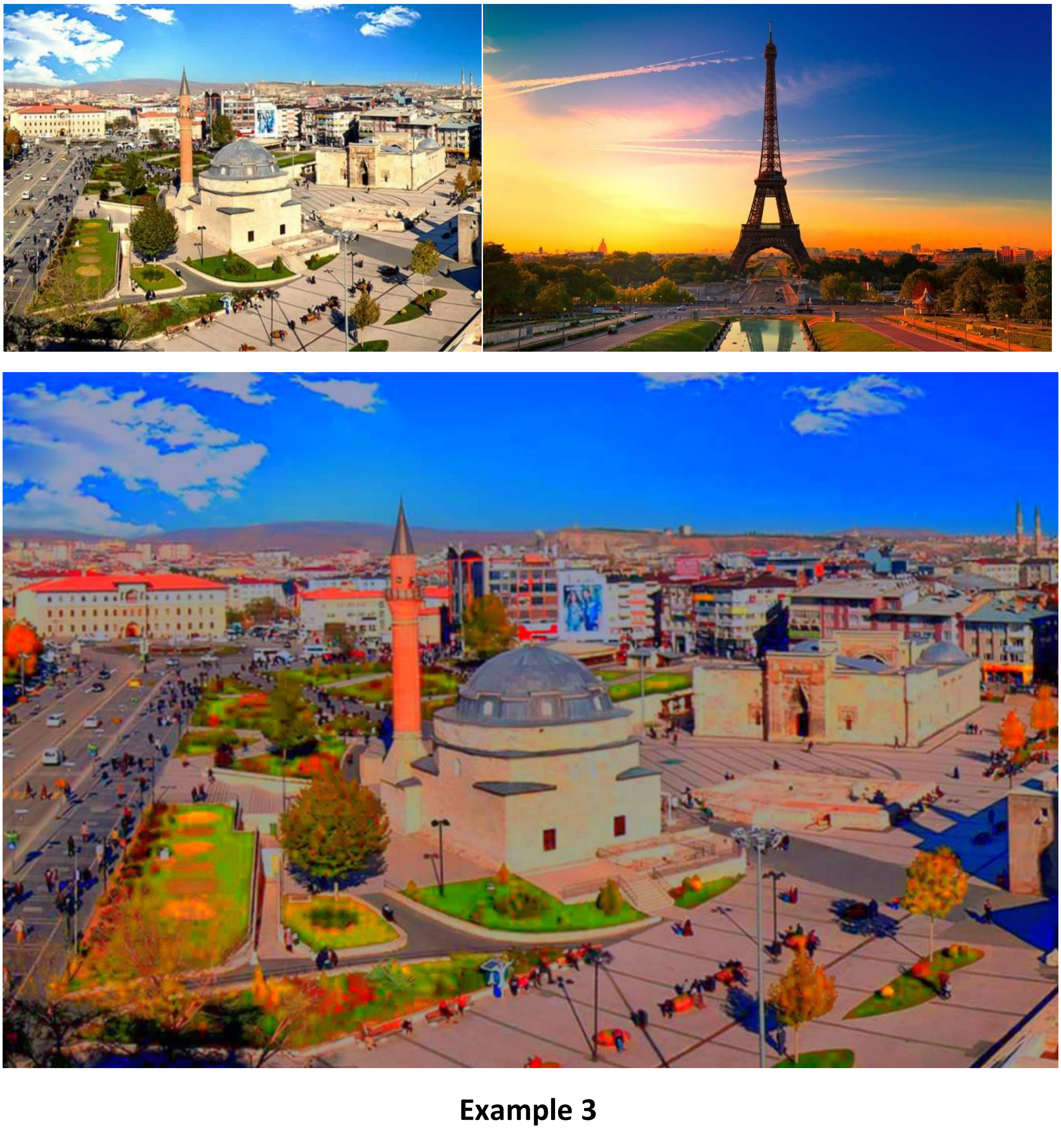
RGB operates on three channels: red, green and blue. Lab is a conversion of the same information to a lightness component L*, and two color components “a* and b*” for the four unique colors of human vision: red, green, blue, and yellow. Lightness is kept separate from color, so that we can adjust one without affecting the other. "Lightness" is designed to approximate human vision, which is very sensitive to green but less to blue. If you brighten in Lab space, the result will often look more correct to the eye, color-wise. In general we can say that when using positive values for the saturation slider in Lab space, the colors come out more “fresh”, while using the same amount of saturation in RGB makes colors look “warmer”. So L*a*b is designed to approximate human vision. Additionally, lab color space is logarithmic, which means to a first approximation that uniform changes in channel intensity tend to be equally detectable.

Show the results of your implementation for several images.

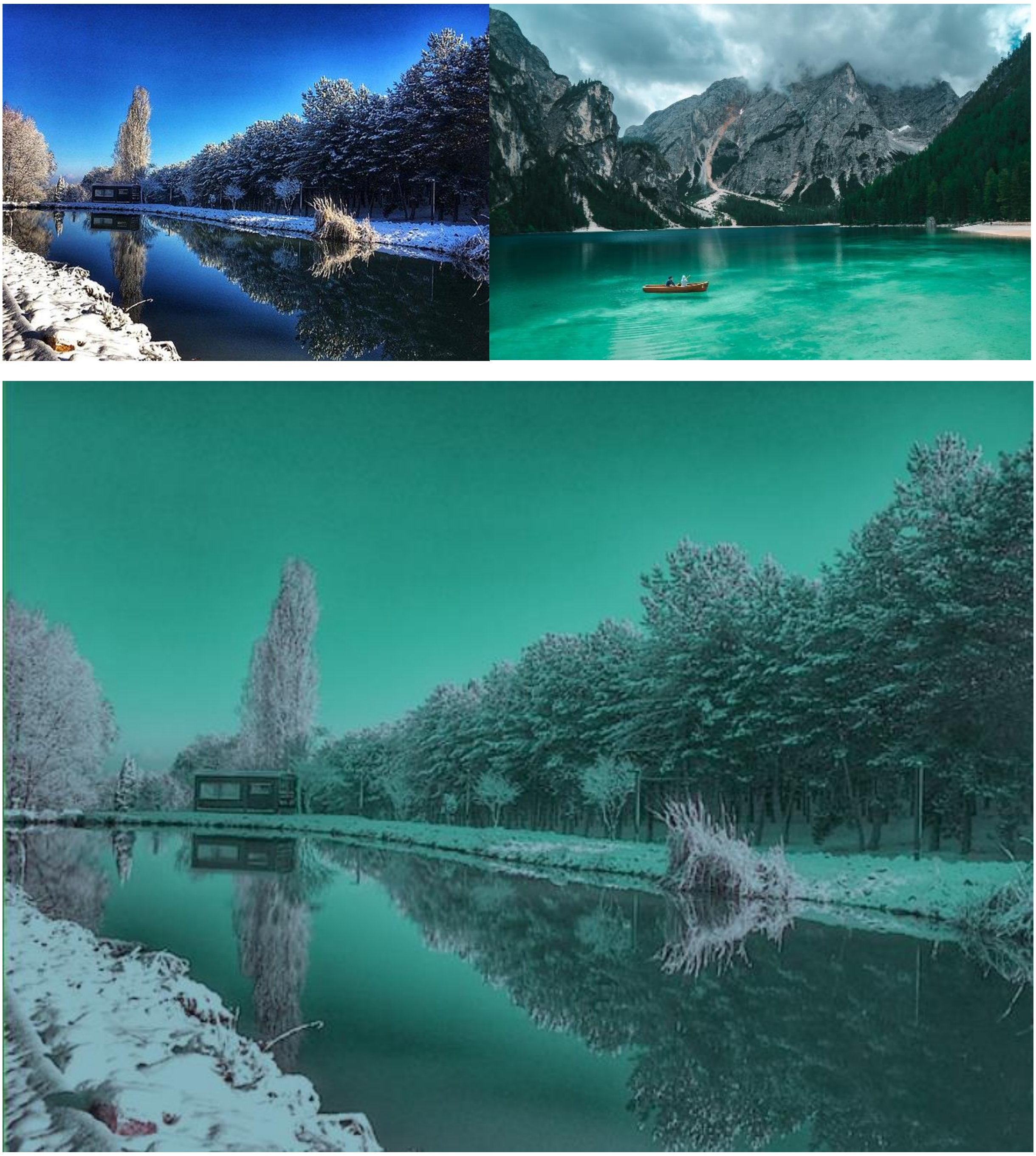
In my examples the left top image will be the source image and the right top image will be target image. The middle bottom image will be the result image of color transfer.

Example 1

Example 2



Example 3



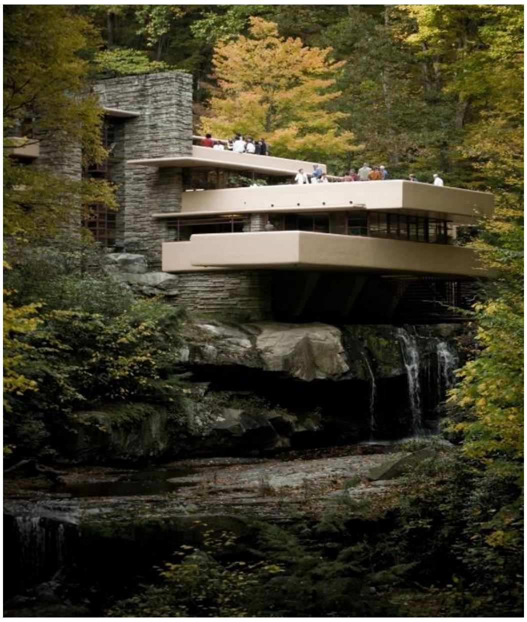
Example 4

Example 5

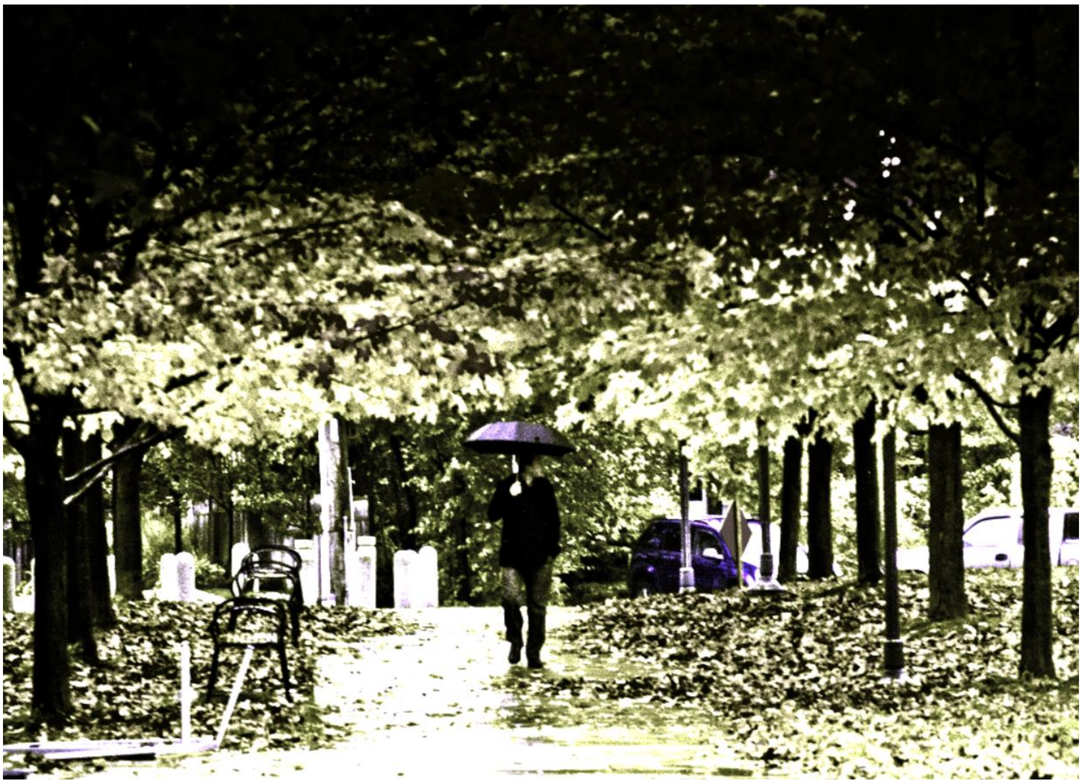
Example 6

What are the disadvantages of the given color transfer algorithm? Show some failure results of the algorithm. Comment about the reasons.

I got some bad looking images when I use two images which has very different color palette. Also using a low resolution image may cause undesirable results. So, we can say that, in this algorithm we shouldn't use too different color palettes and low resolution images. In addition, this algorithm works slowly. Especially when we use image which has higher resolution, the algorithm give the results too late. I my example, again the left top image is source and right top image is target.

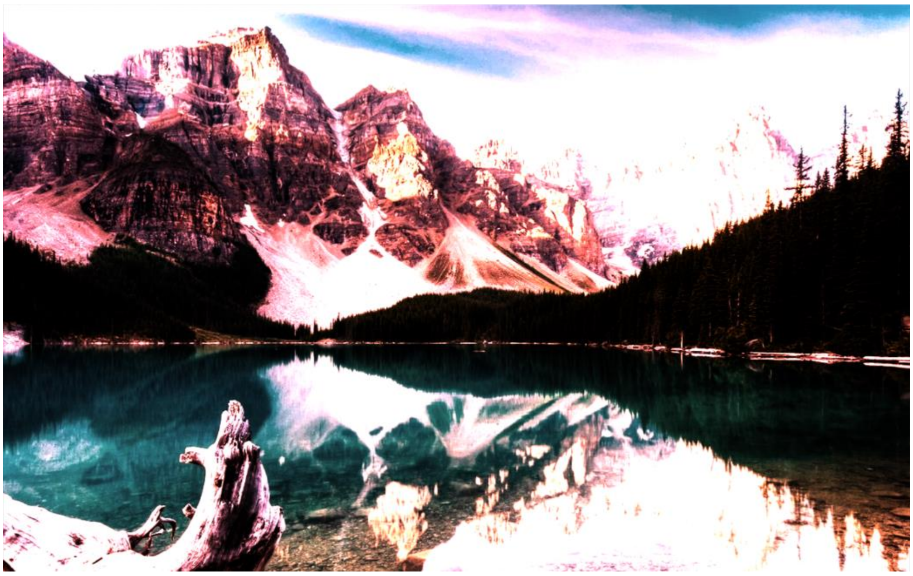


And in this example when I change the source and target image between this two images above, I get this:



Which one is the source and which one is the target doesn’t matter. Still we get bad result.

But sometimes when we change the places of the source image and the target image we can get good results. For instance in my “result of implemention” examples, in example 2, I used the palm tree photograph as source and used the lake photograph as target. I got a really nice image but if I use lake photograph as source and palm tree as target, my result will not be good looking. The mountains and the forest in the picture looks really bad.



REFERENCES AND SOURCES

1.https://rawpedia.rawtherapee.com/RGB_and_Lab

2.https://en.wikipedia.org/wiki/CIELAB_color_space

3.http://www.iro.umontreal.ca/~ostrom/varcoeffED/SIGGRAPH01_varcoeffED.pdf

4. https://www.cs.tau.ac.il/~turkel/imagepapers/ColorTransfer.pdf

5. https://www.colourphil.co.uk/lab_lch_colour_space.shtml

6. R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial greyscale. Proceedings of the Society of Information Display 17, 75–77 (1976)