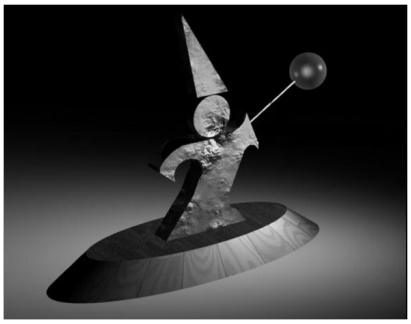
BBM 415 - Image Processing Practicum

HACETTEPE UNIVERSITY
Department of Computer Engineering

Problem Set 1
Fall 2021-2022
TA. Yunus Bilge
Dr. Aydın Kaya

Due Date: 23:59:59, 12/11/2021 Friday

PART 1: Dithering



(a) Original 8-bit image

(b) 1 bit image-Quantized

(c) 1 bit image-Floyd-Steinberg dithering

Dithering is a technique that is used to prevent errors because of quantization. As you learned in the class quantization is a discretization method which can cause errors due to limited intensity resolution. As an example, consider an 8-bit grayscale image (in Fig. 1 (a)) is converted a 1-bit image as a result of quantization as in Figure 1 (b). As you can see the loss of the visual content in the image is a lot. The shape of the object is damaged, some meaningless contours are occurred etc. Dithering methods prevent these kinds of errors by creating the illusion of continuous-tone images as you can see in Figure 1 (c). The quantized image generated with Floyd-Steinberg dithering algorithm [1].

The Floyd-Steinberg dithering algorithm is based on error diffusion. The main step of the algorithm is distributing the quantization error to neighborhood according to the distribution given below.

$$\begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & * & 7/16 & \dots \\ \dots & 3/16 & 5/16 & 1/16 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

$*$,... represent the current pixel and blank pixels which are not important for current operation, respectively.

The goal of this part of assignment to implement Floyd-Steinberg dithering algorithm and analyzing the results by comparing them with the quantized images without dithering.

The Implementation Details

1. You will implement two Python files for this part. First one is a Python script named "pa1_1.py". It has to contain following steps:
   - Read grayscale image
   - Quantize image without dithering
   - Call function for dithering
2. Second Python file, "pa1_2.py", will contain a function named "FloydSteinberg", $FloydSteinberg(image, q)$. q is the parameter for the quantization.
   - Implement the algorithm by following the given steps:

```
for each y from top to bottom
  for each x from left to right
    oldpixel = pixel[x][y]
    newpixel = find_quantized_value(oldpixel)
    pixel[x][y] = newpixel
    quant_error = oldpixel - newpixel
    pixel[x + 1][y ] := pixel[x + 1][y ] + quant_error * 7 / 16
    pixel[x - 1][y + 1] := pixel[x - 1][y + 1] + quant_error * 3 / 16
    pixel[x ][y + 1] := pixel[x ][y + 1] + quant_error * 5 / 16
    pixel[x + 1][y + 1] := pixel[x + 1][y + 1] + quant_error * 1 / 16
```

What should you write in the report?

- How is the given method achieves to prevent quantization error? Explain with examples.
- What is the behavior of the algorithm for different q parameters?
- Compare the quantized image and dithered image for different q parameters.
- What are the disadvantages of Floyd-Steinberg dithering algorithm? Explain with examples.

PART 2 : Color Transfer

Source image

Target Palette

Result of Color Transfer

The goal of this part of the assignment is to implement a method that transfers color characteristics of an image to another. You will implement one of the early study in this area Reinhard et al. [2]. The method consists of a simple statistical analysis.

The Implementation Details

1. You will edit two Python files for this part. First one is a Python script named "pa2_1.py". It has to contain following steps:
   - Read source image which is the original image that will be manipulated
   - Read target image which will be used for color characteristics
   - Call function for color transfer
2. Second Python file will contain a function named "colorTransfer" and has a prototype $colorTransfer(source, target)$
   - Implement the algorithm to transfer color between images by following the given steps:

Color Transfer Algorithm

1. Apply the given transformation to convert RGB source and target images to LMS cone space:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2. Convert data to logarithmic space for both source and target images:

$$\begin{aligned} \boldsymbol{L} &= \log L \\ \boldsymbol{M} &= \log M \\ \boldsymbol{S} &= \log S \end{aligned}$$

3. Apply the given transformation to convert to $l\alpha\beta$ space:

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{bmatrix}$$

4. Compute the mean and variance of the images for l,α,β channels. For the source image means are represented with $mean_{l_s}$,$mean_{\alpha_s}$,$mean_{\beta_s}$ and variances are represented with var_{l_s},var_{α_s},var_{β_s}. For the target image means are represented with $mean_{l_t}$,$mean_{\alpha_t}$,$mean_{\beta_t}$ and variances are represented with var_{l_t},var_{α_t},var_{β_t}.

5. Subtract the mean of source image from the source image:

$$\begin{aligned} l^* &= l - mean_{l_s} \\ \alpha^* &= \alpha - mean_{\alpha_s} \\ \beta^* &= \beta - mean_{\beta_s} \end{aligned}$$

6. Scale the data points by the respective standard deviations:

$$\begin{aligned} l^{`} &= \frac{var_{l_t}}{var_{l_s}} l^* \\ \alpha^{`} &= \frac{var_{\alpha_t}}{var_{\alpha_s}} \alpha^* \\ \beta^{`} &= \frac{var_{\beta_t}}{var_{\beta_s}} * \end{aligned}$$

7. Add the target's mean to the scaled data points:

$$\begin{aligned} l^{``} &= l^{`} + mean_{l_t} \\ \alpha^{``} &= \alpha^{`} + mean_{\alpha_t} \\ \beta^{``} &= \beta^{`} + mean_{\beta_t} \end{aligned}$$

8. Apply transform matrix to convert $l\alpha\beta$ to LMS:

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix}$$

9. Go back to linear space:

$$L = L^{10}$$
$$M = M^{10}$$
$$S = S^{10}$$

10. Apply transform matrix to convert LMS to RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

11. Observe the result.

What should you write in the report?

- Why does the algorithm change color space from RGB to Lab?
- Show the results of your implementation for several images (you can use different images except the provided ones).
- What are the disadvantages of the given color transfer algorithm? Show some failure results of the algorithm. Comment about the reasons.

Grading

The assignment will be graded out of 100:

- **50 (part 1):** CODE: 0 (no implementation), 5 (a partially correct solution), 20 (a correct solution) and REPORT: 30
- **50 (part 2):** CODE: 0 (no implementation), 5 (a partially correct solution), 20 (a correct solution) and REPORT: 30

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial grey scale. Proceedings of the Society of Information Display 17, 75–77 (1976)

[2] Erik Reinhard, Michael Ashikhmin, Bruce Gooch and Peter Shirley, 'Color Transfer between Images', IEEE CG&A special issue on Applied Perception, Vol 21, No 5, pp 34-41, September - October 2001