Digital Image Processing (CSE/ECE 478)
Monsoon-2018
Assignment-1 (200 points)
Posted on: 06/08/18
Due on: 16/08/18

---

1. (30 points) The function `domColors(im, k)` takes an image `im` and an integer `k` and returns a list of `k` most frequently occurring colors.
   1. Implement the function. The code should work for both grayscale and color images. (15 points)
   2. Implement a display function which takes the image `im` as input, the list returned by `domColors(im, k)` and displays the image, the most dominant color and the palette of `k` dominant colors (see Figure 1). (10 points)
   3. What could be some uses of this function ? (5 points)

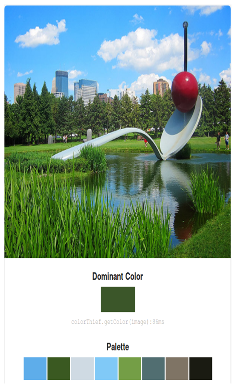
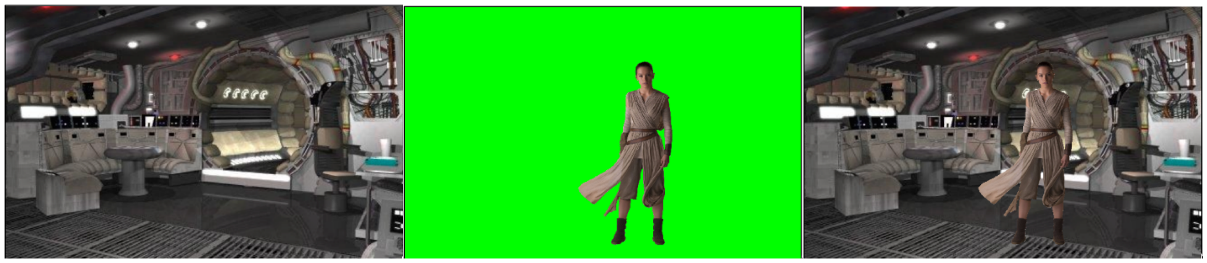


Figure 1: An example of input image and output for question 1.

2. (30 points) In the technique of chroma keying, a particular color or a range of colors is used to represent a key color that can later be made transparent using a computational process – see Figure 2.



(a) background image (b) foreground image (c) After chroma keying

Figure 2: Chroma Keying

- Implement a function `ChromaKey` which takes two images `fg,bg` as input and RGB color `keyColor` corresponding to key color and outputs the chroma-keyed result. For example, Fig 2a would be `bg`, Fig 2b would be `fg`, `keyColor` would be $[0, 255, 0]$ (green). Figure 2c is the result. (15 points)
- Show the output in a format similar to Figure 2 for the object in Figure 3. Choose a suitable/realistic background image of your choice. (10 points)
- Show the outputs for other object and background images of your choice. Be creative ! (5 points)

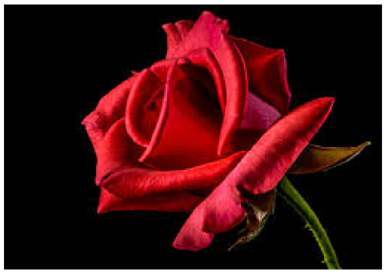
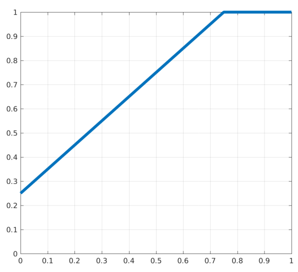
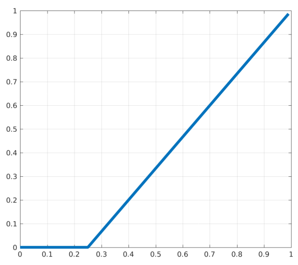
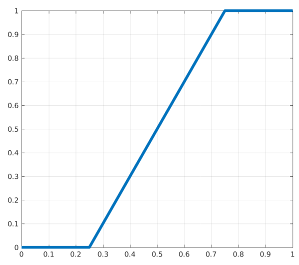
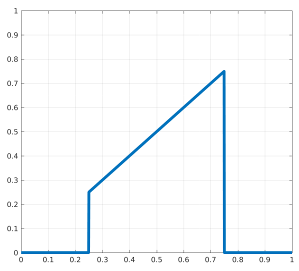
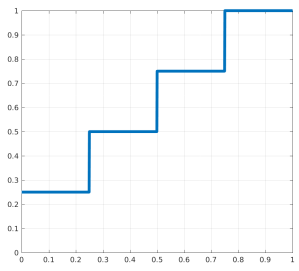


Figure 3: Rose for question 2.

multirow

3. (30 points) 1. Write a function to implement a **piecewise** linear transform $z' = g(z) = K_1^i \cdot z + K_2^i; \quad a^i \leq z \leq b^i$. The function `transformIntensity(im, K1, K2, a, b)` takes an input image `im`, coefficients `K1,K2` and intervals `a,b` for each linear segment and produces the transformed output image (15 points).

   2. Write the analytical forms for the following transforms and using your implementation from part (a), produce transformed outputs of `lena` image for each function (15 points).

4. (20 points)
    1. Write a function to compute and plot histogram of an image. (5 pts)
    2. Make an observation table noting how many bytes are required to store histogram representation and to store the raw image (pixels) for these sizes of `lena` image : (16x16, 32x32, 64x64, 128x128, 256x256). Normalize the histograms by total pixel counts and compare them across different image sizes. What is your understanding and observations (15 points).

5. (35 points)
    1. Implement the following pixel-level transforms and show results on images provided separately.
        (a) Histogram Equalization - (Images - `hist-equal.jpg` and `hist-equal2.jpg`) (10 points)
        (b) Local Histogram Equalization - (Images - `hist-equal.jpg` and `hist-equal2.jpg`) (10 points)
        (c) Histogram Matching - (Match histogram of `hist-match-1.jpg` to `hist-match-2.jpg`) (10 points)
    2. Based on the results, what are your observations ? Compare the result of your histogram equalization code ((1) above) to the result obtained using MATLAB's in-built function `histeq`. Are the outputs same ? If not, why not ? (5 points)

6. (25 points)
    1. Write a function `BitQuantizeImage` which takes an 8-bit image `im` and k, the number of bits to which the image needs to be quantized to and returns the k-bit quantized image. (10 points)
    2. What is the range of k for 8-bit images ? In general, for an n-bit image, what is the range as a function of n ? (5 points)
    3. Ensure that the function `BitQuantizeImage` performs error-checking keeping the valid range of k in mind. (5 points)
    4. Write a display function which takes `im` as input and plots k-bit quantized images for all valid values of k. (5 points)

7. (30 points) Figure 4 contains examples of a High Contrast Image and a Low Contrast Image. Choose 3 High Contrast and 3 Low Contrast Images of your choice and plot their histograms. Keep the resolution of all the images the same to compare adequately.

8. (20 points) Write code to display different bit planes of an 8-bit gray-scale image. Try this on various images. Which bit plane seems to have most information from original image?

9. (30 points) Binarization or Thresholding typically is a crucial step towads machine-based understanding content of document images. For the sample document images (`palm-leaf-1.jpg` and `palm-leaf-2.jpg`) are attached separately. Write a function `BinarizeImage` which outputs a binarized version of input image `im`. Note that the input image may be RGB.

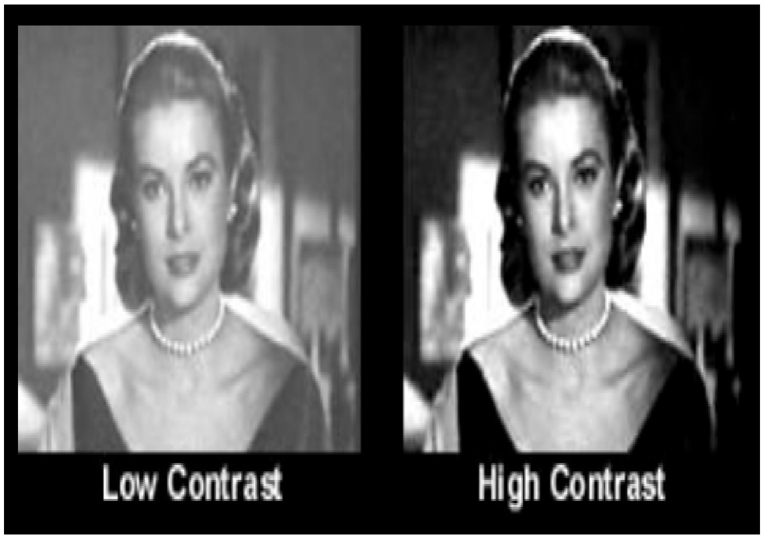


Figure 4: Low and High Contrast Images

- Compare global thresholding and locally adaptive thresholding. For global thresholding, use a threshold chosen by Otsu's method. The function you code can take thresholding type as an argument and any other optional arguments (e.g. required for locally adaptive thresholding). (15 points)
- Write a script which plots the original image, the binarized version for the methods mentioned in (2). (5 points)
- Comment on the effectiveness of global thresholding vs locally adaptive thresholding. For instance, what are the choices that need to be made for locally adaptive thresholding ? What is the effect of those choices on output ? (10 points)