Homework 2 solution template
CMPSCI 370 Spring 2019, UMass Amherst
Name: DongWon Park

Here is a template that your solutions should roughly follow. Include outputs as figures, and code should be included in the end.

1 Light

(a) Formula for S_{TOTAL}.
S_{TOTAL} = 2 * [$\sum_\lambda S_{\text{INC}}(\lambda)$] + 3 * [$\sum_\lambda S_{\text{LED}}(\lambda)$]

(b) Tristimulus theory.

(1) Value of the matrix R

$$R = \begin{bmatrix} 0.0114 & 0.0564 & 0.1551 \\ 0.0264 & 0.1841 & 0.1225 \\ 0.2305 & 0.0553 & 0.0052 \end{bmatrix}$$

(2) Coefficient for the colors
turquoise: b_1: 2.1739, b_2: 4.4402, b_3: 0.0928
goldenrod: b_1: 0.4831, b_2: 0.0169, b_3: 5.4819

2 White balance

1. Proof for the formula of L
Our assumption is that average pixel color (C_{ave}) under white light ($L = (1.0, 1.0, 1.0)$) is gray $(128, 128, 128)$. Then, from the equation $L * C = I$, each pixel color is $C = \frac{I}{L} = (\frac{i_r}{l_r}, \frac{i_g}{l_g}, \frac{i_b}{l_b})$ and also we can write down the average pixel color as $C_{ave} = (\frac{r_{ave}}{1.0}, \frac{g_{ave}}{1.0}, \frac{b_{ave}}{1.0})$, where $l_r = l_g = l_b = 1.0$ everywhere because the light is neutral on our assumption, and where r_{ave},g_{ave} and b_{ave} are average reflected color of the image.
Then, $C_{ave} = (\frac{r_{ave}}{1.0}, \frac{g_{ave}}{1.0}, \frac{b_{ave}}{1.0})$ = (128, 128, 128)
$(\frac{r_{ave}}{i_r}, \frac{g_{ave}}{i_g}, \frac{b_{ave}}{i_b})$ = (128, 128, 128)
$(\frac{r_{ave}}{128}, \frac{g_{ave}}{128}, \frac{b_{ave}}{128})$ = (i_r, i_g, i_b) = L
Thus, $L = (\frac{r_{ave}}{128}, \frac{g_{ave}}{128}, \frac{b_{ave}}{128})$

2. Value of L.

   Light l_r: 0.7951, l_g: 0.8745, l_b: 1.1960

Figure 1: Output for the white balance

3 Hybrid images

σ_1 = 6, σ_2 = 8

wall-e image E.T. image

Figure 2: Source images.

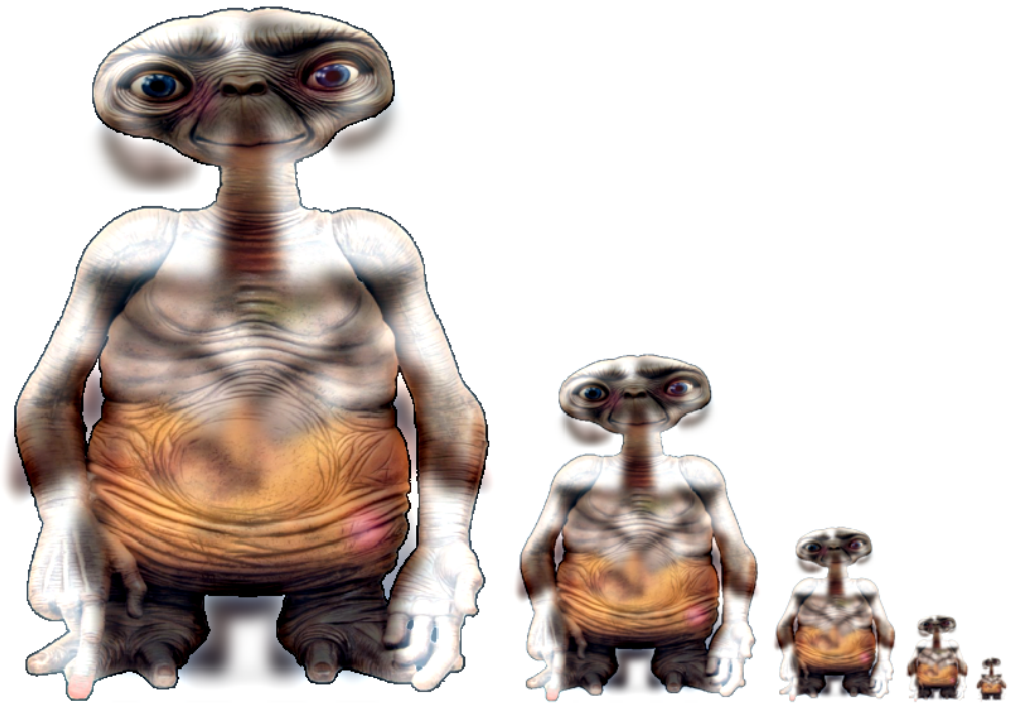

Figure 3: Output of hybid image of the WALL-E and E.T.. The image was created with $\sigma_1 = 6$ and $\sigma_2 = 8$.

4 Solution code

Include the source code for your solutions as seen below (only the files you implemented are necessary). In latex the command `verbatiminput{alignChannels.m}` allows you to include the code verbatim as seen below. Regardless of how you do this the main requirement is that the included code is readable (use proper formatting, variable names, etc.) A screenshot of your code works too provided you include a link to source files.

4..1 Computing matrix R

```
function R = response()
    F1 = [ 0.00, 0.01, 0.01, 0.02, 0.01, 0.02, 0.07, 0.29, 0.35, 0.12 ];
    F2 = [ 0.00, 0.01, 0.02, 0.11, 0.20, 0.25, 0.21, 0.10, 0.01, 0.00 ];
    F3 = [ 0.03, 0.10, 0.25, 0.27, 0.13, 0.02, 0.01, 0.01, 0.00, 0.00 ];
    Sr = [ 0.16, 0.26, 0.28, 0.15, 0.10, 0.03, 0.02, 0.00, 0.00, 0.00 ];
    Sg = [ 0.00, 0.00, 0.04, 0.23, 0.34, 0.23, 0.15, 0.01, 0.00, 0.00 ];
    Sb = [ 0.00, 0.00, 0.00, 0.00, 0.01, 0.04, 0.08, 0.23, 0.35, 0.29 ];

    St = [Sr; Sg; Sb];
    Ft = [F1; F2; F3];
    R = St * Ft';
end
```

4..2 Solving the multiplier b

```
function B = weights(C)
    R = response()
    B = R^(-1) * C'
end
```

4..3 grayworld.m

```
function [L, C] = grayworld (I)
%     I = im2double(I);
    r_avg = mean2(I(:,:,1))
    g_avg = mean2(I(:,:,2))
    b_avg = mean2(I(:,:,3))

%     L = [ r_avg/0.5, g_avg/0.5, b_avg/0.5]
    L = [ r_avg/128, g_avg/128, b_avg/128]

    C(:,:,1) = I(:,:,1).*(L(1)^(-1));
    C(:,:,2) = I(:,:,2).*(L(2)^(-1));
    C(:,:,3) = I(:,:,3).*(L(3)^(-1));
end
```

4..4 hybridImagem

```
function H = hybridImage(im1, im2, sigma1, sigma2)
    im1d = im2double(im1);
    im2d = im2double(im2);

    filter1 = fspecial('gaussian', 6*sigma1 + 1, sigma1);
    filter2 = fspecial('gaussian', 6*sigma2 + 1, sigma2);

    blurry1 = imfilter(im1d, filter1);
    blurry2 = imfilter(im2d, filter2);

    sharp1 = im1d - blurry1;
    sharp2 = im2d - blurry2;

    H = blurry1 + sharp2;
end
```