1 Written

1a.

$$\alpha_i(C) = \int C(\lambda) S_i(\lambda)\, d\lambda \tag{1}$$

$$\alpha_i(M) = \int \underbrace{\left[\sum_{k=1}^{2} \beta_k(C) P_k(\lambda)\right]}_{M(\lambda)} S_i(\lambda)\, d\lambda \tag{2}$$

From (1):

$$\begin{aligned}
\alpha_1(C) &= \int_5^6 \left(\frac{1}{2}\lambda - 2\right) d\lambda && (3)\\
&= \left[\frac{1}{4}\lambda^2 - 2\lambda\right]_5^6 + [\lambda]_6^7 && (4)\\
&= \frac{3}{4} + 1 && (5)\\
&= \frac{7}{4} && (6)
\end{aligned}$$

$$\begin{aligned}
\alpha_2(C) &= \int_5^6 \left(\frac{1}{2}\right) d\lambda && (7)\\
&= \frac{1}{2} && (8)
\end{aligned}$$

If $P_1(\lambda) = \delta(\lambda - 5)$ and $P_2(\lambda) = \delta(\lambda - 7)$ then, from (2)

$$\begin{aligned}
\alpha_1(M) &= \int_4^6 [\beta_1\delta(\lambda-5) + \beta_2\delta(\lambda-7)] \left(\frac{1}{2}\lambda - 2\right) d\lambda && (9)\\
&\quad + \int_6^8 [\beta_1\delta(\lambda-5) + \beta_2\delta(\lambda-7)]\, d\lambda\\
&= \left[\left(\frac{1}{2}\right)\beta_1 + \left(\frac{3}{2}\right)\beta_2\right] + [\beta_1 + \beta_2] && (10)\\
&= \frac{3}{2}\beta_1 + \frac{5}{2}\beta_2 = \frac{7}{4} && (11)
\end{aligned}$$

$$\begin{aligned}
\alpha_2(M) &= \int_4^6 [\beta_1\delta(\lambda-5) + \beta_2\delta(\lambda-7)] \left(\frac{1}{2}\right) d\lambda && (12)\\
&= \frac{1}{2}\beta_1 + \frac{1}{2}\beta_2 = \frac{1}{2} && (13)
\end{aligned}$$

Solving for β_1 and β_2

$$\begin{aligned}
\frac{3}{2}\beta_1 + \frac{5}{2}\beta_2 &= \frac{7}{4} && (14)\\
\beta_1 + \beta_2 &= 1 && (15)\\
\beta_1 &= \frac{3}{4} && (16)\\
\beta_2 &= \frac{1}{4} && (17)
\end{aligned}$$

2 Computer Assignment

The program for searching is listed in Listing 1. The code to separate an image into its RGB channels is listed in Listing 2. Figure 1 shows the results of q1.tif. Figure 2 shows the results of q2.tif. In both cases, the leftmost image is the query, followed by the top result, and then the next best result.

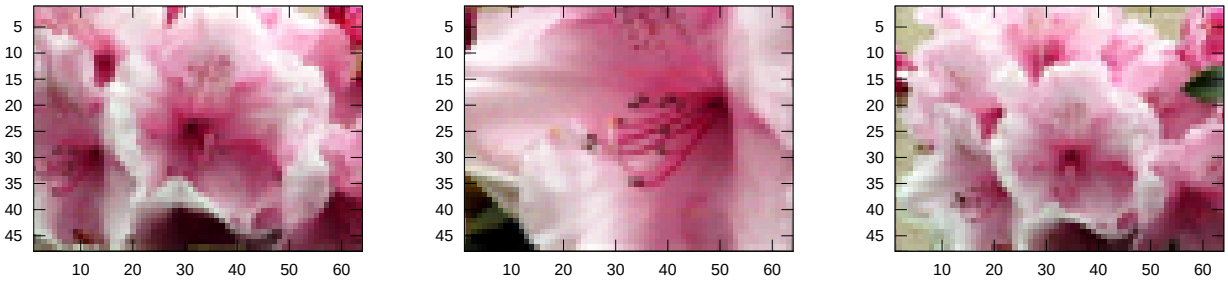

Figure 1: Query q1.tif

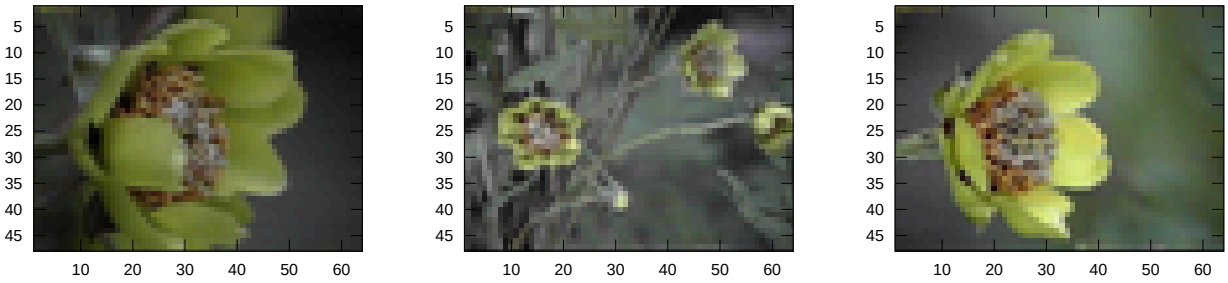

Figure 2: Query q2.tif

The colorplot3d program shows the 5 images in a 3D space. As expected, images 1 and 4 are close to each other and 2 and 5 are close to each other. 1 and 4 seem like predominately green images, however the 3D space shows that images 2 and 5 have a higher intensity of green. This can be explained by the amount of white in these images, which would also explain the high intensity of blue in those images. Image 1 and 3 are quite close to each other even though they look nothing alike. That means that this search technique wont be very good if a query was very similar to image 1, maybe with a little more red and blue in it.

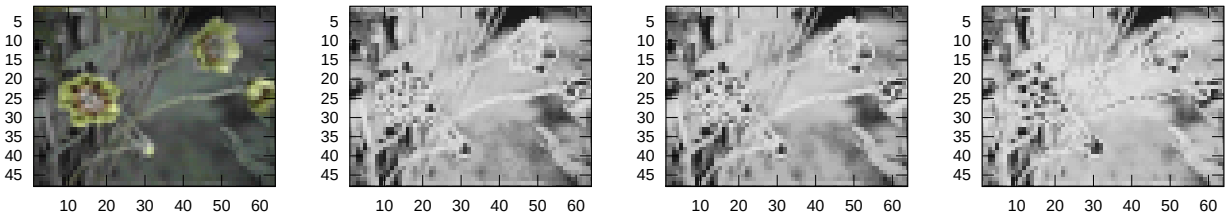

Figure 3: Image 1.tif along with its RGB channels

3 Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%         File: search.m
%       Author: Jay Mundrawala(jay@ir.iit.edu)
%      Created: Sat Feb 6 2010
%
%  Description: This script reads the tif files in the images directory
%               and calculates their average values for the red, green,
%               and blue channels. It then asks the user for query images,
%               and displays the two closest images to the query image.
%
%        Usage: This script works correctly with octave. Make sure pwd is
%               the root directory of the project.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Print current working directory...the images folder should be visible to
%this directory.
printf("pwd=%s\n",pwd());

%Get non query images from the images directory
dir_list = dir('./images/');
files = {};
for i = 1:length(dir_list)
    if(length(regexpi(dir_list(i).name, '^(\d+)\.tif$', 'match')) > 0)
        files(end+1) = ["images/", dir_list(i).name];
    end
end

%Store mean rgb values for each image
db_rgb = {};
for i = 1:length(files)
    db_img = imread(files{i});
    db_mean_r = mean(mean(db_img(:,:,1)));
    db_mean_g = mean(mean(db_img(:,:,2)));
    db_mean_b = mean(mean(db_img(:,:,3)));
    db_rgb{end+1} = [db_mean_r, db_mean_g, db_mean_b];
end

%Query Interaction
c=0;
while(1)
    q_img = imread(input('query: ','s'));
    q_mean_r = mean(mean(q_img(:,:,1)));
    q_mean_g = mean(mean(q_img(:,:,2)));
    q_mean_b = mean(mean(q_img(:,:,3)));

    dist_ary = [];
    top1 = 1;
    top2 = 0;
    for i = 1:length(db_rgb)
        rgb = db_rgb{i};
        dist = sqrt((q_mean_r - rgb(1))^2 +
                    (q_mean_g - rgb(2))^2 +
                    (q_mean_b - rgb(3))^2);
        dist_ary(i) = dist;
        if(i > 1)
            if(dist < dist_ary(top1))
                top2 = top1;
                top1 = i;
            elseif(top2 == 0)
                top2 = i;
            elseif(dist < dist_ary(top2))
                top2 = i;
            end
        end
    end

    t1 = imread(files{top1});
    t2 = imread(files{top2});
    figure;
```

```
    subplot(1,3,1), imagesc(q_img); axis image;
    subplot(1,3,2), imagesc(t1); axis image;
    subplot(1,3,3), imagesc(t2); axis image;
    output = ["data/s", num2str(c), ".eps"];
    printf("Saving query as %s\n", output);
    print(output, '-deps');
    pause(1);
    c++;
end
```

Listing 1: Search script

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %
3  %          File: rbg_channel.m
4  %        Author: Jay Mundrawala(jay@ir.iit.edu)
5  %       Created: Sat Feb 6 2010
6  %
7  %   Description: This script reads the tif files and displays the original
8  %                image, followed by the its RGB channels seperated.
9  %
10 %         Usage: This script works correctly with octave. For matlab, lines
11 %         27 and 28 need to commented out.
12 %
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14
15 c = 0;
16 while(1)
17     q_img = imread(input('query: ','s'));
18     q_r = q_img(:,:,1);
19     q_g = q_img(:,:,2);
20     q_b = q_img(:,:,3);
21
22     figure;
23     subplot(1,4,1), imagesc(q_img); axis image;
24     subplot(1,4,2), imagesc(q_r); axis image;
25     subplot(1,4,3), imagesc(q_g); axis image;
26     subplot(1,4,4), imagesc(q_b); axis image;
27
28     output = ["data/c", num2str(c++), ".eps"]
29     print(output, "-deps");
30     pause(1);
31 end
```

Listing 2: RGB Channel script