影像處理導論 HW4

宋其諭 0510888

Project goal

Consider the RGB color image, **violet (clolr).tif** below.

(a) Determine and plot the H, S and I component images.

(b) Apply sphere-based color slicing to the image, using the prototypical color (i) ***a*1** = (134, 51, 143), and (ii) ***a*2** = (131, 132, 4), and the same radius of the sphere, *R*0 = 30.

Steps to be followed:

1. Read a RGB image
2. Represent the RGB image in the range [0 1]
3. Find HSI components

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right\}$$

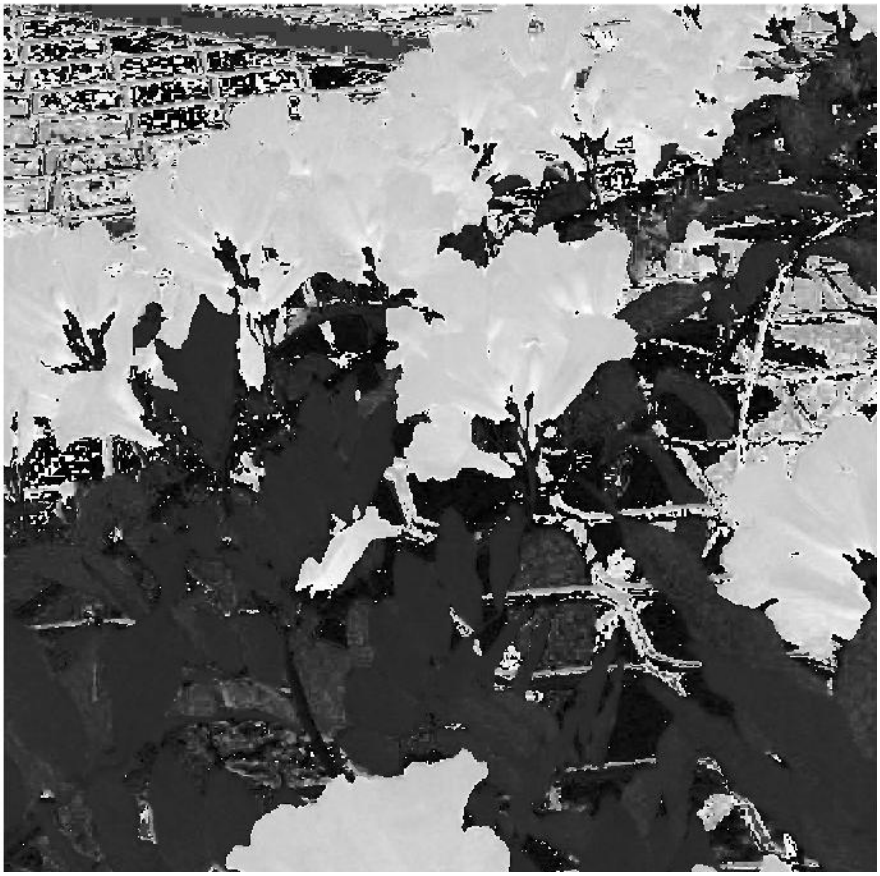
4. H(Hue) = $\begin{cases}\theta & \text{If B<=G}\\ 360-\theta & \text{If B>G}\end{cases}$

5. S(Saturation) = 1- $\frac{3}{(R+G+B)}\left[\min(R,G,B)\right]$

6. I(Intensity)= $\frac{1}{3}(R+G+B)$

1. Figures of H, S and I component images(30%)

Hue component image(H)

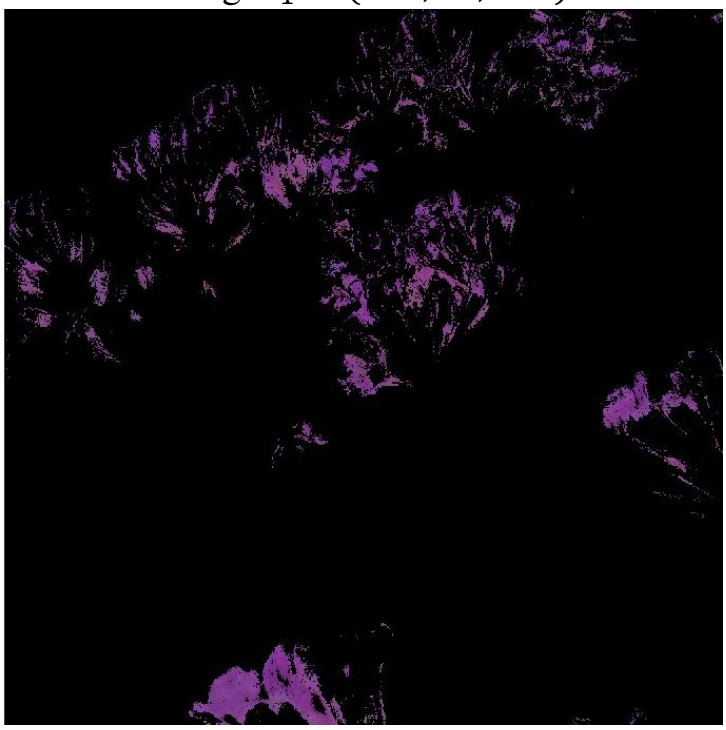


Saturation component image(S)

Intensity component image(I)

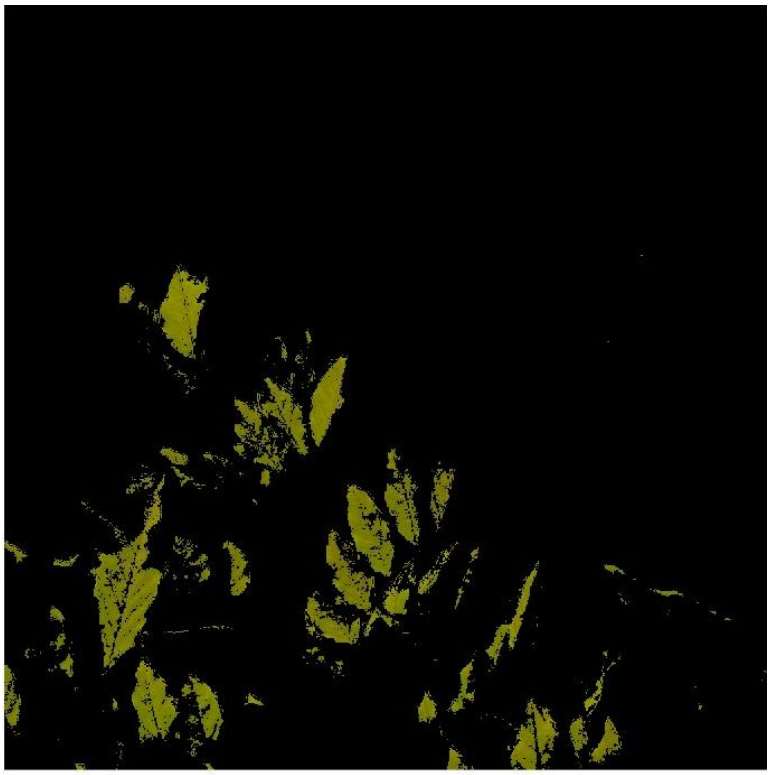
2. Figure of color-slicing image using a_1 (20%)

Color slicing $a_1 = (134, 51, 143)$



3. Figure of color-slicing image using a_2 (20%)

Color slicing $a_1 = (131, 132, 4)$



Source codes

本次實驗使用 Matlab 軟體分析(含註解)

```
clear all;   close all;   clc;
% 1024*1024
origin_img = imread('violet (color).tif');
img = im2double(origin_img);

% (a)-----------------------------------------------------------------
% RGB(double)
R = img(:,:,1);
G = img(:,:,2);
B = img(:,:,3);
% Hue
numi=1/2*((R-G)+(R-B));
denom=((R-G).^2+((R-B).*(G-B))).^0.5;
H=acosd(numi./(denom+0.000001));
H(B>G)=360-H(B>G);
H=H/360;
%Saturation
S=1 - (3./(sum(img,3)+0.000001)).*min(img,[],3);
%Intensity
I=sum(img, 3)./3;

figure;
subplot(2,3,2), imshow(origin_img), title('origin image');
subplot(2,3,4), imshow(H, []), title('Hue');
subplot(2,3,5), imshow(S, []), title('Saturation');
subplot(2,3,6), imshow(I, []), title('Intensity');

% (b)-----------------------------------------------------------------
% RGB(integer)
int_R = double(origin_img(:,:,1));
int_G = double(origin_img(:,:,2));
int_B = double(origin_img(:,:,3));

output = zeros(1024,1024,3);
for i = 1:1024
    for j = 1:1024
```

```
            % a1 = [134 51 143]
            a1 = int_R(i, j)-134;
            b1 = int_G(i, j)-51;
            c1 = int_B(i, j)-143;
            distance1 = a1.^2 + b1.^2 + c1.^2;
            if distance1 <= 900
                output1(i,j,:) = img(i, j, :);
            end
            % a2 = [131 132 4]
            a2 = int_R(i, j)-131;
            b2 = int_G(i, j)-132;
            c2 = int_B(i, j)-4;
            distance2 = a2.^2 + b2.^2 + c2.^2;
            if distance2 <= 900
                output2(i,j,:) = img(i, j, :);
            end
        end
    end
    figure;
    subplot(1,3,1), imshow(origin_img), title('origin image');
    subplot(1,3,2), imshow(output1, []),title('color slicing at (134, 51, 143)');
    subplot(1,3,3), imshow(output2, []),title('color slicing at (131, 132, 4)');
```