1 Technical description

(1) RGB

如果直接在RGB color space用 histogram equalization 會造成某些時候顏色產生偏誤，所以用power law直接調整RGB的明亮度， 讓過曝的照片暗一點，讓曝光度不足的照片亮一點

(2) HSI

將 RGB image 透過公式的運算轉成 HSI color space，在 HSI color space 上對components 上進行所需的影像處理。 HSI 各個 components 可以經由公式從RGB image 得到

H ：

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases}$$

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

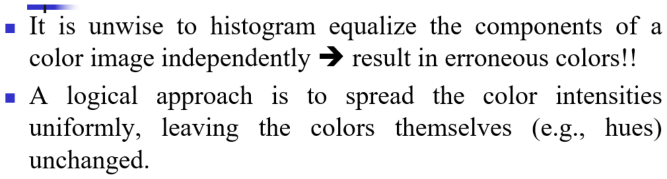
S ：

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

I ：

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

得到 HSI image 後，再做 historgram equalization。 根據投影片來看，直接對顏色部分的 component 做 histeq，會造成顏色錯誤，所以在 HSI color space 上做 histogram equalization 時，要讓 hue component 維持不變。



- It is unwise to histogram equalize the components of a color image independently ➔ result in erroneous colors!!
- A logical approach is to spread the color intensities uniformly, leaving the colors themselves (e.g., hues) unchanged.

從下圖可以了解到對intensity做處理是最有效的，所以就針對intensity做histogram equalization

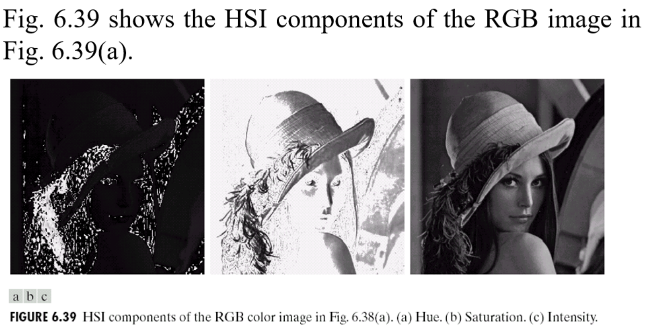
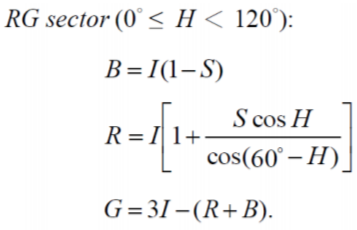


Fig. 6.39 shows the HSI components of the RGB image in Fig. 6.39(a).

FIGURE 6.39 HSI components of the RGB color image in Fig. 6.38(a). (a) Hue. (b) Saturation. (c) Intensity.

在HSI color space 做完處理後，接著要將圖片轉回RGB color space，可由以下公式得到RGB的components

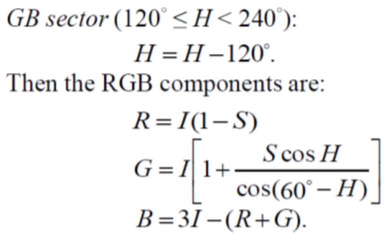


RG sector $(0^\circ \le H < 120^\circ)$:

$$B = I(1-S)$$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$$G = 3I - (R+B).$$



GB sector $(120^\circ \le H < 240^\circ)$:

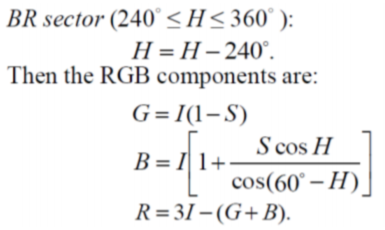
$$H = H - 120^\circ.$$

Then the RGB components are:

$$R = I(1-S)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$$B = 3I - (R+G).$$



BR sector $(240^\circ \le H \le 360^\circ)$:

$$H = H - 240^\circ.$$

Then the RGB components are:

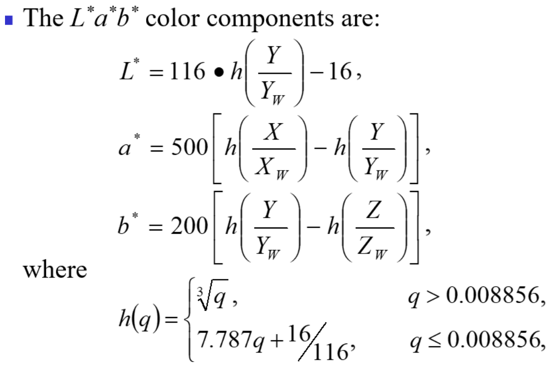
$$G = I(1-S)$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$$R = 3I - (G+B).$$

(3） L * a * b*

RGB無法直接轉成L * a * b*，所以先將RGB轉XYZ，再將XYZ轉到 L * a * b*，根據投影片，XYZ可由以下公式得到 L * a * b*的 components



- The $L^*a^*b^*$ color components are:

$$L^* = 116 \bullet h\left(\frac{Y}{Y_W}\right) - 16,$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right],$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right],$$

where

$$h(q) = \begin{cases} \sqrt[3]{q}, & q > 0.008856, \\ 7.787q + 16/116, & q \le 0.008856, \end{cases}$$

The L * a * b* system is an excellent decoupler of intensity (L * *) and color (a* = R-G,* b* *= G-B*)

針對L *，也就是intensity 的部分做histogram equalization，做完之後再由L * a * b*轉XYZ，XYZ轉RGB，之後在RGB做display。

程式碼：

```
% 初始化
img_array=cell(1,4);
img_array{1} = imread('aloe.jpg');
img_array{2} = imread('church.jpg');
img_array{3} = imread('house.jpg');
img_array{4} = imread('kitchen.jpg');
for i=1:4
    % RGB
    if i == 1
        I_RGB=5*(im2double(img_array{i}).^0.8);
    elseif i == 2
        I_RGB=2.5*(im2double(img_array{i}).^0.8);
    elseif i==3
```

```
        I_RGB=1*(im2double(img_array{i}).^1.5);
    else
        I_RGB=1*(im2double(img_array{i}).^1.1);
    end
    % HSI
    I_HSI=rgbtohsi(img_array{i});
    H=I_HSI(:,:,1);
    S=I_HSI(:,:,2);
    I=histeq(I_HSI(:,:,3));
    I_HSI=cat(3,H,S,I);
    I_HSI=hsitorgb(I_HSI);

    %L*a*b
    %L部分原本是0~100，拉伸到0~255做histeq後，再還原到原本的範圍
    I_LAB = rgbtolab(img_array{i});
    L=I_LAB(:,:,1);
    a=I_LAB(:,:,2);
    b=I_LAB(:,:,3);
    L=2.56 * double(L);
    L=histeq(uint8(L));
    L=double(L)/ 2.56;

    I_LAB=cat(3,L,a,b);
    I_LAB = labtorgb(I_LAB);
    figure;
    sgtitle('Color Image Enhancement');
    subplot(2,2,1),imshow(img_array{i});title('Original image');
    subplot(2,2,2),imshow(I_RGB);title('enhance RGB');
    subplot(2,2,3),imshow(I_HSI);title('enhance HSI');
    subplot(2,2,4),imshow(I_LAB);title('enhance LAB');
end
```

rgb轉hsi 以及hsi轉rgb，公式在 Technical description中

```
function I_HSI = rgbtohsi(I_RGB)
    I_RGB = im2double(I_RGB);
    R = I_RGB(:, :, 1);
    G = I_RGB(:, :, 2);
    B = I_RGB(:, :, 3);

    num = 0.5*((R - G) + (R - B));
    den = sqrt((R - G).^2 + (R - B).*(G - B));
    theta = acos(num./(den + eps));

    H = theta;
    H(B > G) = 2*pi - H(B > G);
    H = H/(2*pi);

    num = min(min(R, G), B);
    den = R + G + B;
    den(den == 0) = eps;
    S = 1 - 3.* num./den;
    H(S == 0) = 0;
    I = (R + G + B)/3;

    I_HSI = cat(3, H, S, I);
    I_HSI = im2uint8(I_HSI);
```

```
end

function I_RGB = hsitorgb(I_HSI)

I_HSI = im2double(I_HSI);
H = I_HSI(:, :, 1) * 2 * pi;
S = I_HSI(:, :, 2);
I = I_HSI(:, :, 3);

R = zeros(size(I_HSI, 1), size(I_HSI, 2));
G = zeros(size(I_HSI, 1), size(I_HSI, 2));
B = zeros(size(I_HSI, 1), size(I_HSI, 2));
% RG sector (0 <= H < 2*pi/3)
idx = find( (0 <= H) & (H < 2*pi/3));
B(idx) = I(idx) .* (1 - S(idx));
R(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx)) ./ cos(pi/3 - H(idx)));

G(idx) = 3*I(idx) - (R(idx) + B(idx));

% BG sector (2*pi/3 <= H < 4*pi/3).
idx = find( (2*pi/3 <= H) & (H < 4*pi/3) );
R(idx) = I(idx) .* (1 - S(idx));
G(idx) = I(idx) .* (1 + S(idx) .* (cos(H(idx) - 2*pi/3)) ./ (cos(pi - H(idx))));
B(idx) = 3*I(idx) - (R(idx) + G(idx));

% BR sector.
idx = find( (4*pi/3 <= H) & (H <= 2*pi));
G(idx) = I(idx) .* (1 - S(idx));
B(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx) - 4*pi/3) ./ cos(5*pi/3 - H(idx)))

R(idx) = 3*I(idx) - (G(idx) + B(idx));

I_RGB = cat(3, R, G, B);
I_RGB = max(min(I_RGB, 1), 0);
I_RGB = im2uint8(I_RGB);
end
```

rgb轉L * a * b* 以及L * a * b*轉rgb，公式在 Technical description中

```
function I_LAB = rgbtolab(I_RGB)
I_RGB = double(I_RGB);
%rgb to xyz
R = I_RGB(:,:,1)/255;
G = I_RGB(:,:,2)/255;
B = I_RGB(:,:,3)/255;
[i,j] = size(R);

for x=1:i
    for y=1:j
        var_R=R(x,y);
        var_G=G(x,y);
        var_B=B(x,y);
        if  var_R > 0.04045
            var_R = ( ( var_R + 0.055 ) / 1.055 ) ^ 2.4;
        else
            var_R = var_R / 12.92;
        end
        if  var_G > 0.04045
            var_G = ( ( var_G + 0.055 ) / 1.055 ) ^ 2.4;
        else
            var_G = var_G / 12.92;
        end
        if  var_B > 0.04045
            var_B = ( ( var_B + 0.055 ) / 1.055 ) ^ 2.4;
        else
            var_B = var_B / 12.92;
        end
        R(x,y)=var_R;
        G(x,y)=var_G;
        B(x,y)=var_B;
    end
end
R = R * 100;
G = G * 100;
B = B * 100;

X = R * 0.4124 + G * 0.3576 + B * 0.1805;
Y = R * 0.2126 + G * 0.7152 + B * 0.0722;
Z = R * 0.0193 + G * 0.1192 + B * 0.9505;

%xyz to L * a * b*
X = X / 95.047 ;
Y = Y / 100.000;
Z = Z / 108.883;
for z=1:i
    for w=1:j
        var_X=X(z,w);
        var_Y=Y(z,w);
        var_Z=Z(z,w);
        if  var_X > 0.008856
            var_X = var_X ^  (1/3) ;
        else
            var_X = ( 7.787 * var_X ) + ( 16 / 116 );
        end
        if  var_Y > 0.008856
            var_Y = var_Y ^ (1/3) ;
        else
            var_Y = ( 7.787 * var_Y ) + ( 16 / 116 );
        end

        if  var_Z > 0.008856
            var_Z = var_Z ^  (1/3) ;
        else
            var_Z = ( 7.787 * var_Z ) + ( 16 / 116 );
        end
        X(z,w)=var_X;
        Y(z,w)=var_Y;
        Z(z,w)=var_Z;
    end
end
L = ( 116 * Y ) - 16;
a = 500 * ( X - Y );
b = 200 * ( Y - Z );
I_LAB=cat(3,L,a,b);
end
```

```
function I_RGB = labtorgb(I_LAB)
I_LAB=double(I_LAB);
L = I_LAB(:,:,1);
a = I_LAB(:,:,2);
b = I_LAB(:,:,3);

[i,j] = size(L);
% L * a * b* XYZ
Y = ( L + 16 ) / 116;
X = a / 500 + Y;
Z = Y - b / 200;

for z=1:i
    for w=1:j
        var_X=X(z,w);
        var_Y=Y(z,w);
        var_Z=Z(z,w);
        if  var_Y^3  > 0.008856
            var_Y = var_Y^3;
        else
            var_Y = ( var_Y - 16 / 116 ) / 7.787;
        end
        if  var_X^3  > 0.008856
            var_X = var_X^3;
        else
            var_X = ( var_X - 16 / 116 ) / 7.787;
        end
        if var_Z^3  > 0.008856
            var_Z = var_Z^3;
        else
            var_Z = ( var_Z - 16 / 116 ) / 7.787;
        end
        X(z,w)=var_X;
        Y(z,w)=var_Y;
        Z(z,w)=var_Z;
    end
end
X = X * 95.047 ;
Y = Y * 100.000;
Z = Z * 108.883;

R = X *  3.2406 + Y * -1.5372 + Z * -0.4986;
G = X * -0.9689 + Y *  1.8758 + Z *  0.0415;
B = X *  0.0557 + Y * -0.2040 + Z *  1.0570;

% L * a * b* XYZ
X = X / 100;
Y = Y / 100;
Z = Z / 100;
for x=1:i
    for y=1:j
        var_R =R(x,y);
        var_G =G(x,y);
        var_B =B(x,y);
        if  var_R > 0.0031308
            var_R = 1.055 * ( var_R ^ ( 1 / 2.4 ) ) - 0.055;
        else
```

```
            var_R = 12.92 * var_R;
        end
        if  var_G > 0.0031308
            var_G = 1.055 * ( var_G ^ ( 1 / 2.4 ) ) - 0.055;
        else
            var_G = 12.92 * var_G;
        end
        if  var_B > 0.0031308
            var_B = 1.055 * ( var_B ^ ( 1 / 2.4 ) ) - 0.055;
        else
            var_B = 12.92 * var_B;
        end
        R(x,y)=var_R;
        G(x,y)=var_G;
        B(x,y)=var_B;
    end
end
I_RGB=cat(3,R,G,B);
end
```

2 Experimental results

3 Discussions

對影像處理時，如果能針對intensity進行平衡或調整，可以減少影像出現顏色跑掉的情形，並提高整體的對比，而轉到HSI和L * a * b* 能將顏色和intensity分開，能更精確處理圖片的intensity。從實驗結果來看，在每一個color space做影像處理，看起來會有些微的不同，效果也有好有壞，所以可以視影像的情況進行處理。

aloe和church在經過處理之後，看起來效果比較好，雖然兩張照片原來都光線不足，但是因為整張照片亮度都差不多，在調整intensity後，效果都算很不錯，house和kitchen因為過度曝光，即使經過histogram equalization，還是無法讓整張照片看起來更加平衡

4 References and Appendix

老師的投影片

http://www.easyrgb.com/en/math.php

https://zh.wikipedia.org/wiki/Lab%E8%89%B2%E5%BD%A9%E7%A9%BA%E9%97%B4

https://www.imageeprocessing.com/2013/06/convert-hsi-image-to-rgb-image.html