Project 4 solution

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

Figure 1. R component image

Figure 2. G component image

Figure 3. B component image

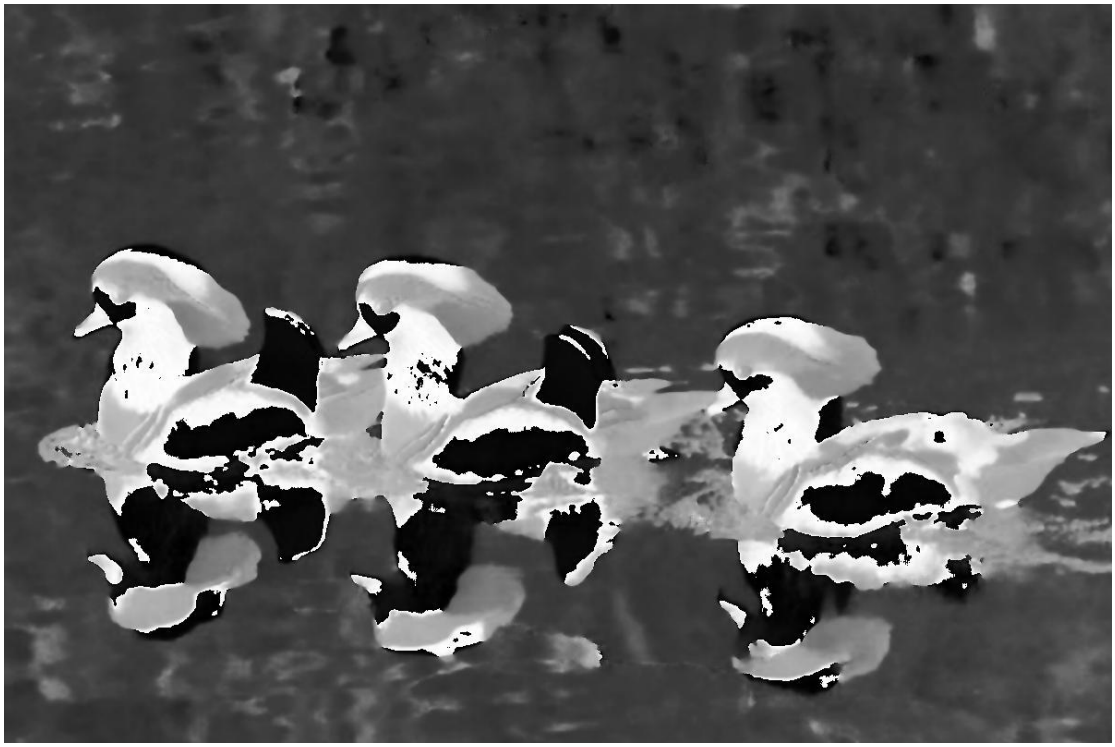


Figure 4. H component image

Figure 5. S component image

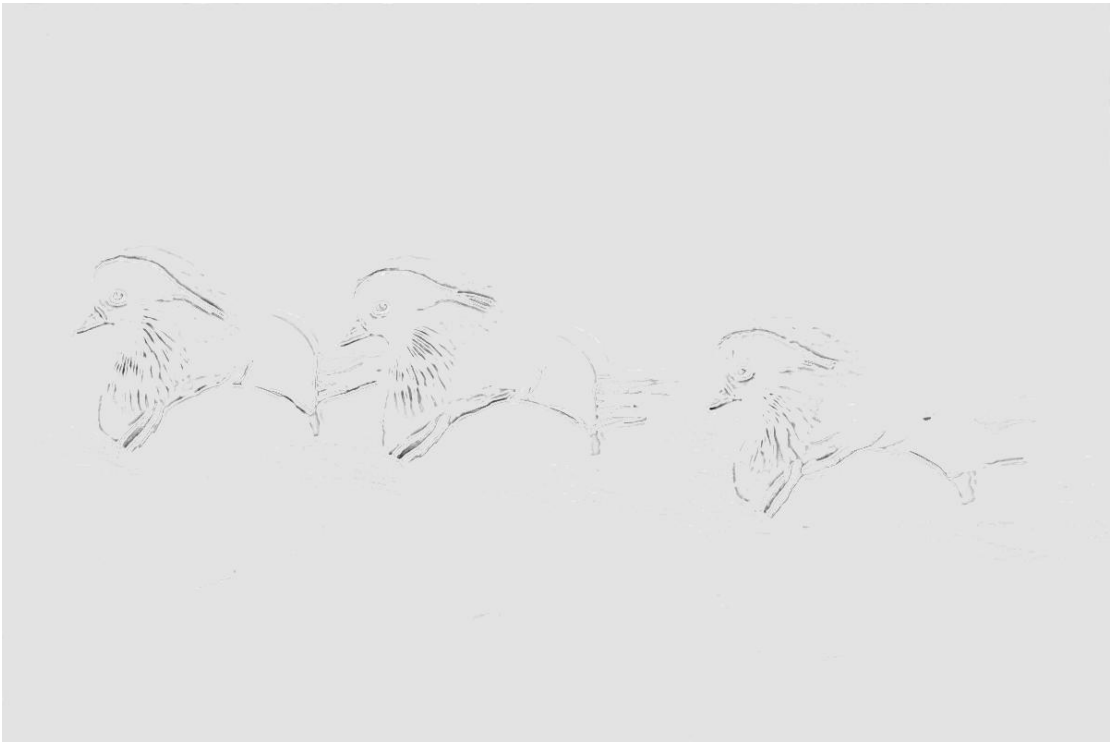
Figure 6. I component image

2. Figures of RGB-based (15%) and HSI-based (15%) sharpened images and their difference image (10%)

(1). RGB-based sharpened image:

(2). HSI-based sharpened image:

(3). Difference image (HSI-based - RGB-based, scaling the image between 0 and 1):



(4). Difference image (RGB-based - HSI-based, scaling the image between 0 and 1):

3. Source code:

```
%%
% Load the image, data type: uint8
img = imread('Bird 3 blurred.tif');

% Get the image size, and change data type to "double"
img1 = im2double(img);
[M,N,comp] = size(img1);

% Get R, G, B component
r = img1(:,:,1);
g = img1(:,:,2);
b = img1(:,:,3);

figure;
imshow(r)
title('R component image')
figure;
imshow(g)
title('G component image')
figure;
imshow(b)
title('B component image')

% H component
num = 0.5.*((r - g) + (r - b));
den = sqrt((r - g).^2 + (r - b).*(g - b));
den(den == 0) = eps;
theta = acosd(num ./ den);   %deg
theta1 = acos(num ./ den);   %rad

%H in rad
H = theta1;
H(b > g) = 2*pi - H(b > g);
H = H ./ (2*pi);
figure;
imshow(H)
title('H component image')
```

```
% S component
num_S = min(min(r, g), b);
den_S = r + g + b;

den_S(den_S == 0) = eps;
S = 1 - 3 .* (num_S ./ den_S);
figure;
imshow(S)
title('S component image')

% I component
I = (r + g + b) ./ 3;
figure;
imshow(I)
title('I component image')
%% Sharpening
kernel1 = [-1, -1, -1;
          -1, 8, -1;
          -1, -1, -1];

c = 1;

% Zero-padding for spatial filtering
r_o = zeros(M+2,N+2);
g_o = zeros(M+2,N+2);
b_o = zeros(M+2,N+2);
I_o = zeros(M+2,N+2);

r_s = zeros(M,N);
g_s = zeros(M,N);
b_s = zeros(M,N);
I_s = zeros(M,N);

r_o(2:M+1,2:N+1) = r;
g_o(2:M+1,2:N+1) = g;
b_o(2:M+1,2:N+1) = b;
I_o(2:M+1,2:N+1) = I;
```

```
 % Spatial filtering
for i = 2:M+1
    for j = 2:N+1
        r_s(i-1, j-1) = sum(sum(r_o(i-1:i+1, j-1:j+1) .* kernel1));
        g_s(i-1, j-1) = sum(sum(g_o(i-1:i+1, j-1:j+1) .* kernel1));
        b_s(i-1, j-1) = sum(sum(b_o(i-1:i+1, j-1:j+1) .* kernel1));
        I_s(i-1, j-1) = sum(sum(I_o(i-1:i+1, j-1:j+1) .* kernel1));
    end
end
RGB_s = cat(3, r + r_s, g + g_s, b + b_s);
RGB_s = max(min(RGB_s, 1), 0);
figure;
imshow(RGB_s)
title('RGB image')

HSI_s = cat(3, H, S, I + I_s);
%%
% Convert HSI to RGB
HSI_rgb = hsi2rgb(HSI_s);
figure;
imshow(HSI_rgb)
title('HSI image')

diff_r = -(RGB_s(:,:,1) - HSI_rgb(:,:,1));
diff_g = -(RGB_s(:,:,2) - HSI_rgb(:,:,2));
diff_b = -(RGB_s(:,:,3) - HSI_rgb(:,:,3));
diff_total = diff_r + diff_g + diff_b;
figure;
imshow(mat2gray(diff_total))
title('Difference image (HSI-RGB)')
%%
diff_r1 = (RGB_s(:,:,1) - HSI_rgb(:,:,1));
diff_g1 = (RGB_s(:,:,2) - HSI_rgb(:,:,2));
diff_b1 = (RGB_s(:,:,3) - HSI_rgb(:,:,3));
diff_total1 = diff_r1 + diff_g1 + diff_b1;
figure;
imshow(mat2gray(diff_total1))
title('Difference image (RGB-HSI)')
```

```
%%
function rgb = hsi2rgb(hsi)
% Extract the individual HSI component images.
H = hsi(:, :, 1) .* (2*pi);
S = hsi(:, :, 2);
I = hsi(:, :, 3);
% Implement the conversion equations.
R = zeros(size(hsi, 1), size(hsi, 2));
G = zeros(size(hsi, 1), size(hsi, 2));
B = zeros(size(hsi, 1), size(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));
rgb = cat(3, R, G, B);
rgb = max(min(rgb, 1), 0);
end
```