Project #4

309605004 歐思辰

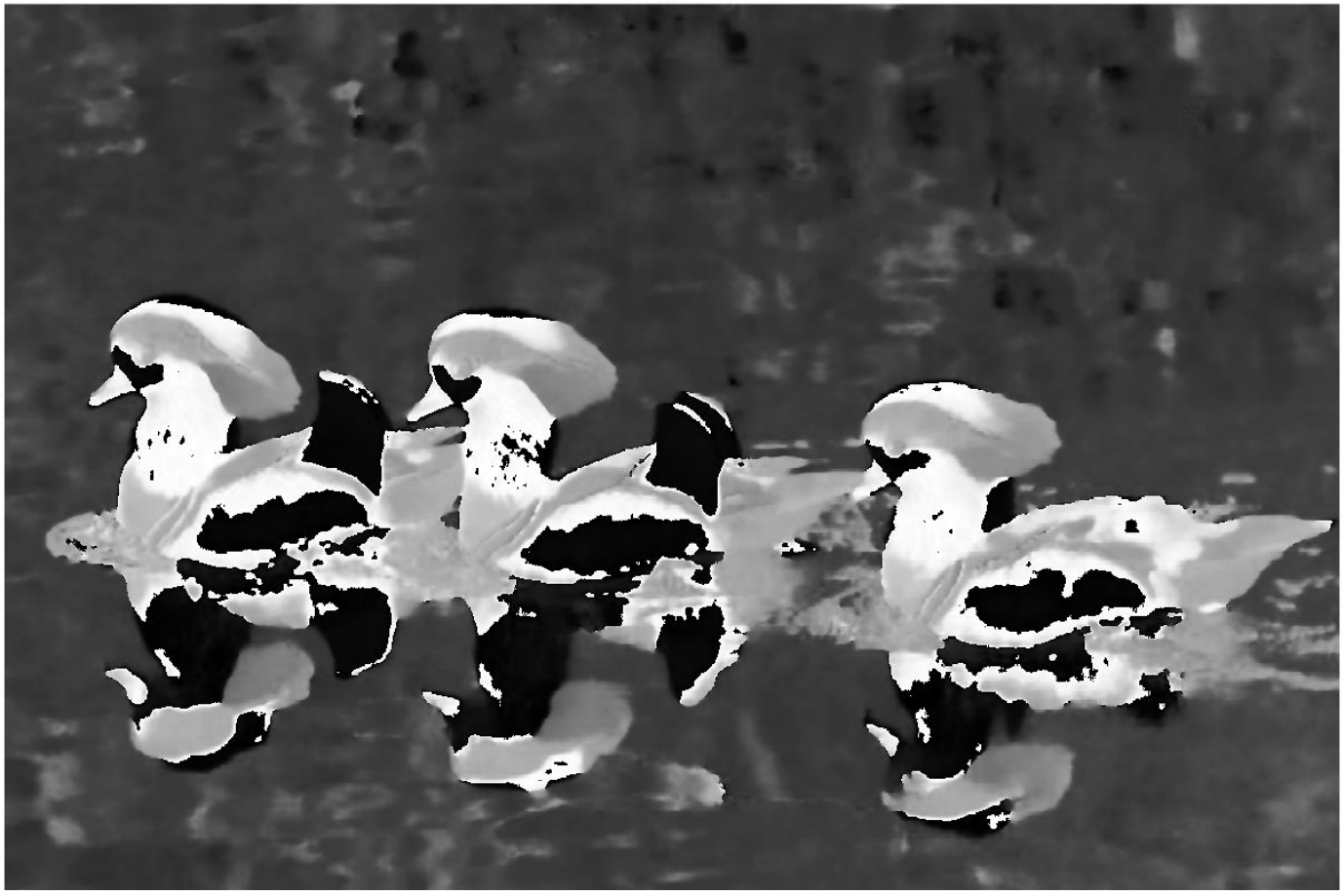
- R, G, B, H, S and I component images

Red

Green

Blue

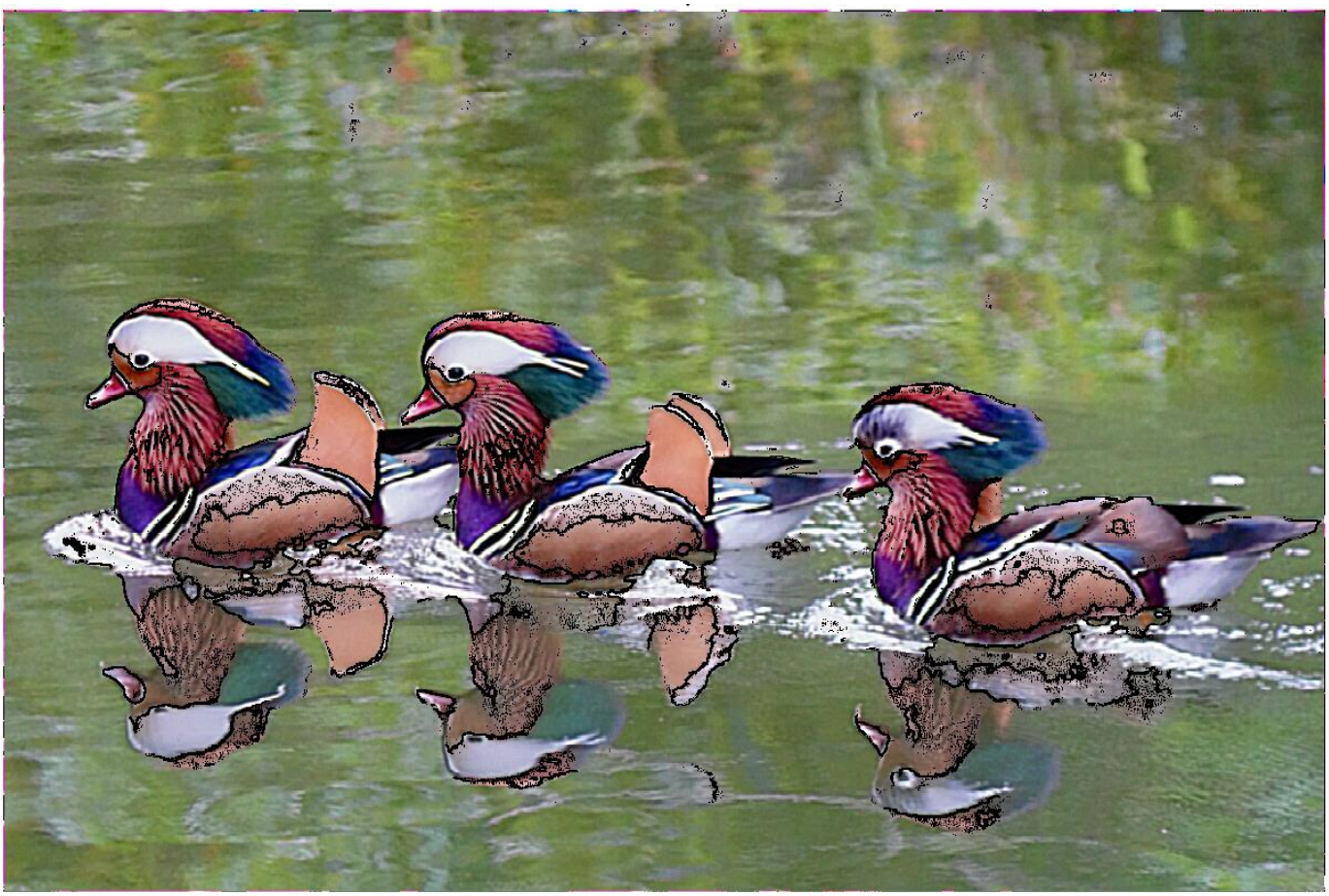
H



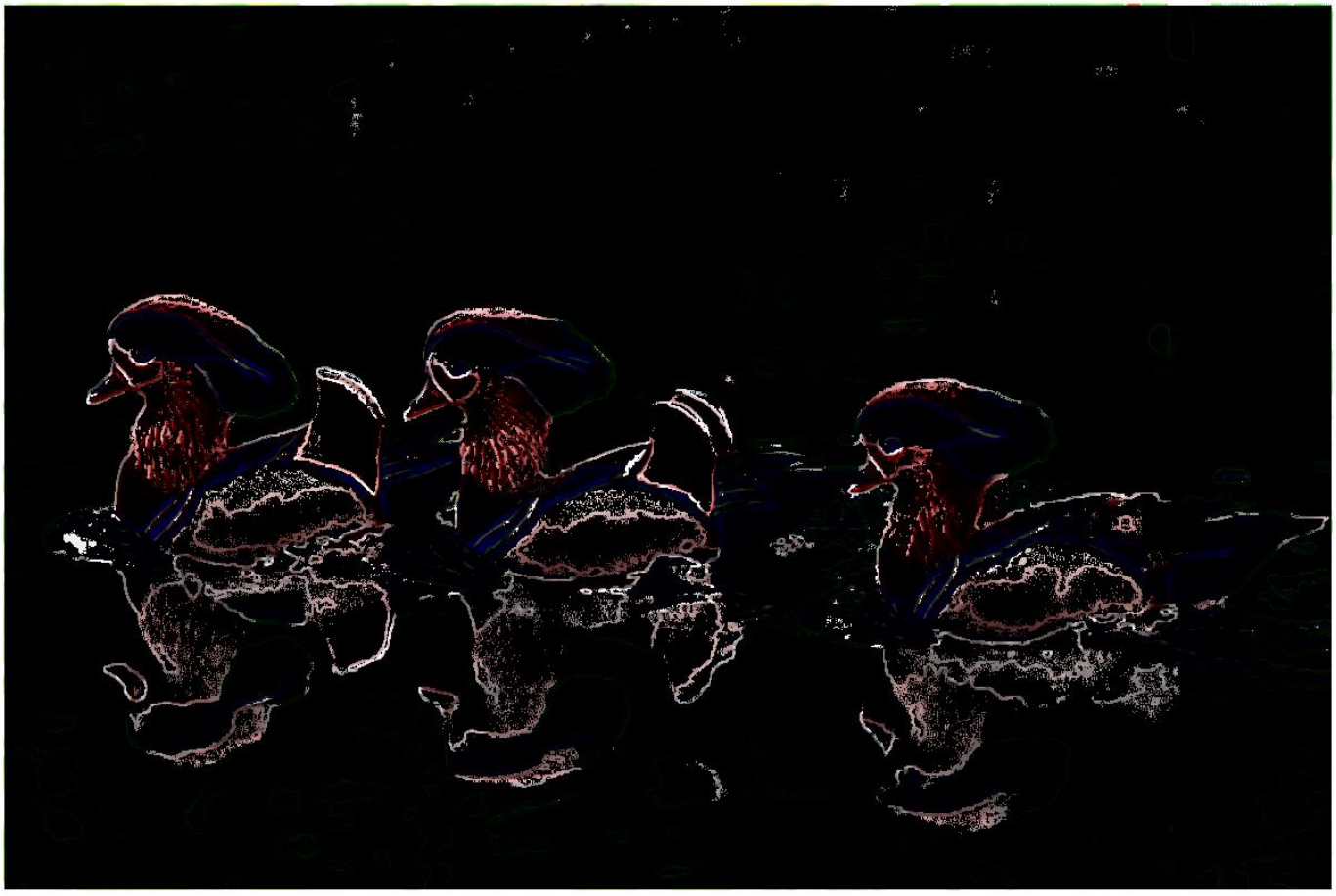
S

- Figures of RGB-based and HSI-based sharpened images and their difference image
  - RGB-based sharpened images

- HSI -based sharpened images



- Sharpened difference image



➢ Source codes

```
clc
clear
ori = imread('Bird 3 blurred.tif');
[m,n,d] = size(ori);
%% rbg hsi component image
ori = im2double(ori);
titleList = { 'Red' 'Green' 'Blue' 'H' 'S' 'I' };
r = ori(:,:,1);
g = ori(:,:,2);
b = ori(:,:,3);
for k = 1:3
  figure;
  imshow(ori(:,:,k));
  title(titleList{k});
end
HSI = zeros(m,n,3);
th=acos((0.5.*((r-g)+(r-b)))./((sqrt((r-g).^2+(r-b).*(g-b)))));
H = th;
H(b>g)=2*pi-H(b>g);
HSI(:,:,1)=H/(2*pi);
HSI(:,:,2)=1-3.*(min(min(r,g),b))./(r+g+b);
HSI(:,:,3)=(r+g+b)/3;
b = ori(:,:,3);
for k = 1:3
  figure;
  imshow(HSI(:,:,k));
  title(titleList{k+3});
end
%% rbg hsi component image
rgb_sharpen = zeros(m,n,3);
hsi_sharpen = zeros(m,n,3);
sharpen_diff = zeros(m,n,3);

kernel = [-1 -1 -1 ; -1 8 -1; -1 -1 -1];
for k = 1:3
  rgb_sharpen(:,:,k) = conv2(ori(:,:,k),kernel,'same') +
ori(:,:,k);
  hsi_sharpen(:,:,k) = conv2(HSI(:,:,k),kernel,'same') +
HSI(:,:,k);
end
hsi_sharpen = hsi2rgb(hsi_sharpen);
```

```
sharpen_diff = rgb_sharpen - hsi_sharpen;
figure ; imshow(rgb_sharpen);title("rgb sharpen");
figure ; imshow(hsi_sharpen);title("hsi sharpen");
figure ; imshow(sharpen_diff);title("sharpen difference");

%% hsi to rgb
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));

 % Combine all three results into an RGB image.  Clip to [0, 1] to
 % compensate for floating-point arithmetic rounding effects.
 rgb = cat(3, R, G, B);
 rgb = max(min(rgb, 1), 0);
end
```