Introduction

Exploring, understanding and investigating underwater activities of images are gaining importance for the last few years. However, the area is still lacking in image processing analysis techniques and methods that could be used to improve the quality of underwater images.

In order to improve the perception of underwater images, an underwater image enhancement method is proposed on the basis of local contrast correction (LCC) and fusion. The characteristics of color and details of underwater images are also considered in the proposed method. First, the white balance method based on red channel compensation is used to correct the image color. Then, two image input versions, LCC and sharpening, are introduced. Finally, the weight is calculated, and multi-scale fusion is performed following the obtained weight.

The results show that the proposed method can be applied to water degradation images in different environments without using the image formation model. Color distortion and unobvious details of underwater images are effectively solved, and the local contrast effect is improved.

Results of the software (MATLAB) are presented in the report.

LITERATURE REVIEW

The underwater image processing area has received considerable attention within the last decades, showing important achievements. In underwater situations, clarity of images is degraded by light absorption and scattering. This causes one colour to dominate the image. The images we are interested on can suffer of one or more of the following problems: limited range visibility, low contrast, non-uniform lighting, blurring, bright artifacts, color diminished (bluish appearance) and noise.

The image processing can be addressed from two different points of view: as an image restoration technique or as an image enhancement method. The image restoration aims to recover a degraded image using a model of the degradation and of the original image formation. Image enhancement uses qualitative subjective criteria to produce a more visually pleasing image and they do not rely on any physical model for the image formation.

Methodology

Given the underwater image formation mechanism and the attenuation of light propagation in the water, an improved LCC method with a multi-scale image fusion strategy is proposed in this study. The underwater image is compensated using the red channel, and the color compensated image is processed by a white balance. Then, two versions of the image input are generated: the LCC image and the sharpened image. Next, the Laplacian contrast weight, saliency weight, and saturation weight of the LCC and sharpened images are calculated, and the two groups of weights are normalized. Finally, LCC images and sharpened images and their corresponding normalized weights are fused. The multi-scale fusion method is also adopted to avoid artifacts. The algorithm flow is shown in Figure 1.

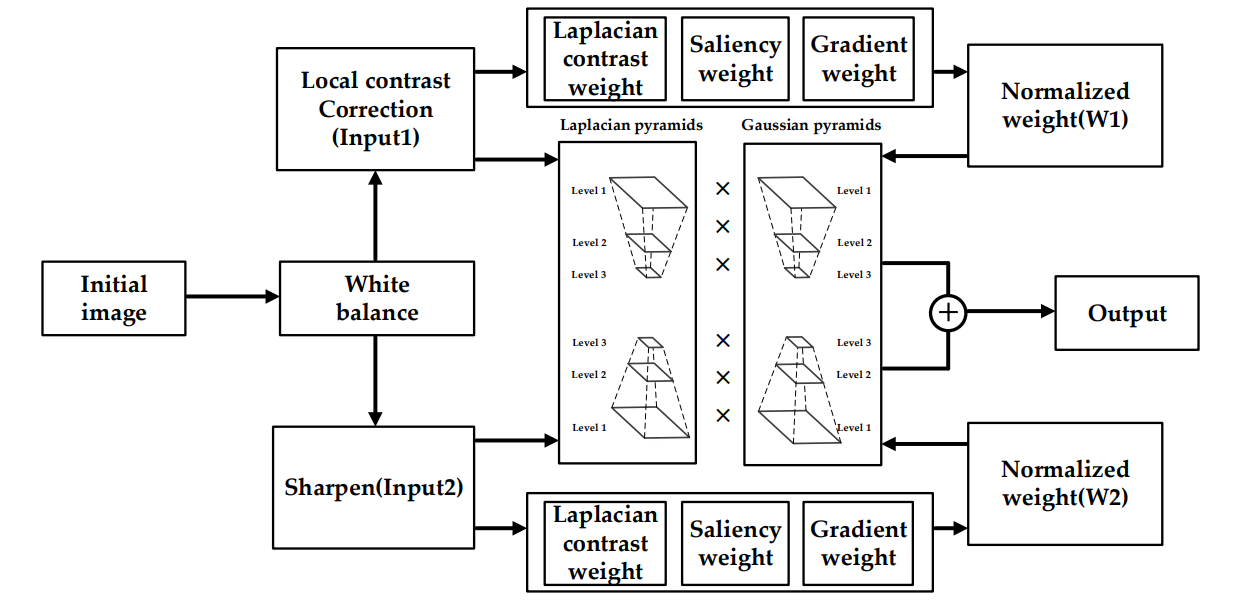


Figure 1. Details of the proposed method. In Details of the proposed method. Input1 and Input2 represent local contrast correction (LCC) and image sharping, Input1 and Input2 represent local contrast correction and image sharp respectively. These two images are used as inputs of the fusion process. Then, the normalized weight maps are obtained, and multi-scale fusion is carried out on this basis.

1. Underwater Image White Balance Based on Red Channel Compensation

Given the physical characteristics of light propagation in water, red light is absorbed first, and underwater images are mainly blue and green. White balance is an effective way to improve the tone of an image. It eliminates unwanted colors created by various lighting or light attenuation characteristics. The Gray-World algorithm is an effective white balance processing method for outdoor images. However, due to its characteristics, the red channel will be overcompensated in the underwater environment where the red attenuation is severe, resulting in red artifacts in the image. The red channel compensation method is used to solve this problem.

1. Improved Local Contrast Correction Method

In addition to color aberration and blurring due to their characteristics, underwater images are usually interfered with by natural light or artificial light, which makes local areas of the image too bright. The white balance processing of the image will also lead to excessive brightness, so introducing the contrast correction method is necessary to solve this problem. The contrast reflects a measurement of the brightness level between the brightest and darkest areas of an image. Gamma correction is widely used as a global contrast correction method. It changes the image brightness by changing the value of a constant index γ. The figure 2 illustrates this relationship. The three transformation curves show how values are mapped when gamma is less than, equal to, and greater than 1.

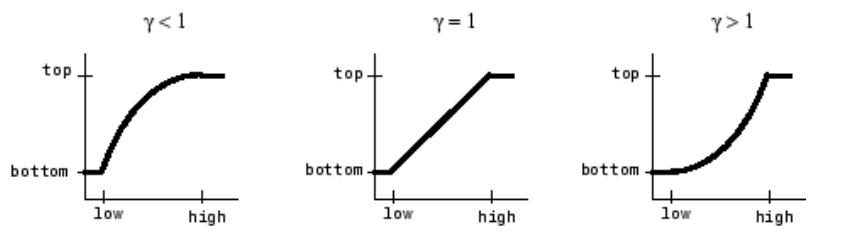


Figure 2. Plots Showing Three Different Gamma Correction Settings. (In each graph, the x-axis represents the intensity values in the input image, and the y-axis represents the intensity values in the output image.)

1. Image Sharpening

Contrast correction can improve overexposure and underexposure of the image. It can also repair the missing color area. However, the underwater image is usually fuzzy, and the details are not obvious. Thus, the sharpening version of underwater images is introduced using Gaussian Filter method. It is typically used before edge detection and aims to reduce the level of noise in the image.

1. Multi-Scale Fusion

Using a weight graph in the fusion process can highlight the pixels with high weight

value in the result. For the selection of weight image, the Laplacian contrast, saliency, and

saturation features of the image are used in the solution.

Laplacian contrast (WL) weight estimates the global contrast. It calculates the absolute

value of the Laplacian filter applied to each luminance channel. This filter can be used to

extend the depth of field of the image because it can ensure that the edges and textures

of the image have high values. However, this weight is not enough to restore contrast

because it cannot distinguish between the ramp and flat regions. Therefore, saliency weight

is also used to overcome this problem.

Saliency weight (WSal) can highlight objects and regions that lose saliency in underwater scenes. The regional contrast-based salient object detection algorithm can produce a full resolution saliency map.

Saturation weight (WSat) makes the fusion algorithm adapt to the chromatic information through a high saturation region.

After the weight estimates of two different input versions are obtained, the three weight estimates of each input version are combined into one weight in the following way: for each input version n, the resulting WL, WSal, and WSat are linearly superimposed to obtain the integrated weight. Then, N aggregated maps are normalized on a pixel-per-pixel basis. The weight of each pixel in each map is divided by the overall weight of the same pixels.

However, this simple fusion will lead to artifacts in the resulting images. Thus, the fusion method based on multi-scale Laplacian pyramid decomposition is adopted. The Laplace operator is applied to get the first layer of the pyramid for the input image version. Then, the second layer image is obtained by down sampling the layer, and so on. A three-tier pyramid is set up. Similarly, the normalized weight version Wn, corresponding to each layer of the Laplacian pyramid, filters the input image using the low-pass Gaussian filter kernel function to obtain the Gaussian pyramid of the normalized weight image.

Implementation/ Program Code

% Method described on Color Balance and Fusion for Underwater Image

close all;

clear all;

clc;

%%% Underwater White Balance %%%

%% Load the image and split channels.

rgbImage=double(imread('sample\_4.jpg'))/255;

grayImage = rgb2gray(rgbImage);

Ir = rgbImage(:,:,1);

Ig = rgbImage(:,:,2);

Ib = rgbImage(:,:,3);

Ir\_mean = mean(Ir, 'all');

Ig\_mean = mean(Ig, 'all');

Ib\_mean = mean(Ib, 'all');

%% Color compensation

alpha = 0.1;

Irc = Ir + alpha\*(Ig\_mean - Ir\_mean);

alpha = 0; % 0 does not compensates blue channel.

Ibc = Ib + alpha\*(Ig\_mean - Ib\_mean);

%% White Balance

I = cat(3, Irc, Ig, Ibc);

I\_lin = rgb2lin(I);

percentiles = 5;

illuminant = illumgray(I\_lin,percentiles);

I\_lin = chromadapt(I\_lin,illuminant,'ColorSpace','linear-rgb');

Iwb = lin2rgb(I\_lin);

figure('name', 'Underwater White Balance');

imshow([rgbImage, I, Iwb])

%%% Multi-Scale fusion.

%% Gamma Correction

Igamma = imadjust(Iwb,[],[],2);

figure('name', 'Gamma Correction');

imshow([Iwb, Igamma]);

%% Image sharpening

sigma = 20;

Igauss = Iwb;

N = 30;

for iter=1: N

Igauss = imgaussfilt(Igauss,sigma);

Igauss = min(Iwb, Igauss);

end

gain = 1;

Norm = (Iwb-gain\*Igauss);

for n = 1:3

Norm(:,:,n) = histeq(Norm(:,:,n));

end

Isharp = (Iwb + Norm)/2;

figure('name', 'image sharpening');

imshow([Iwb,Igauss,Norm, Isharp])

%% weights calculation

% Lapacian contrast weight

Isharp\_lab = rgb2lab(Isharp);

Igamma\_lab = rgb2lab(Igamma);

% input1

R1 = double(Isharp\_lab(:, :, 1)) / 255;

% calculate laplacian contrast weight

WC1 = sqrt((((Isharp(:,:,1)) - (R1)).^2 + ...

((Isharp(:,:,2)) - (R1)).^2 + ...

((Isharp(:,:,3)) - (R1)).^2) / 3);

% calculate the saliency weight

WS1 = saliency\_detection(Isharp);

WS1 = WS1/max(WS1,[],'all');

% calculate the saturation weight

WSAT1 = sqrt(1/3\*((Isharp(:,:,1)-R1).^2+(Isharp(:,:,2)-R1).^2+(Isharp(:,:,3)-R1).^2));

figure('name', 'Image 1 weights');

imshow([WC1 , WS1, WSAT1]);

% input2

R2 = double(Igamma\_lab(:, :, 1)) / 255;

% calculate laplacian contrast weight

WC2 = sqrt((((Igamma(:,:,1)) - (R2)).^2 + ...

((Igamma(:,:,2)) - (R2)).^2 + ...

((Igamma(:,:,3)) - (R2)).^2) / 3);

imshow(WC2);

% calculate the saliency weight

WS2 = saliency\_detection(Igamma);

WS2 = WS2/max(WS2,[],'all');

imshow(WS2);

% calculate the saturation weight

WSAT2 = sqrt(1/3\*((Igamma(:,:,1)-R1).^2+(Igamma(:,:,2)-R1).^2+(Igamma(:,:,3)-R1).^2));

imshow(WSAT2);

figure('name', 'Image 2 weights');

imshow([WC2 , WS2, WSAT2]);

% calculate the normalized weight

W1 = (WC1 + WS1 + WSAT1+0.1) ./ ...

(WC1 + WS1 + WSAT1 + WC2 + WS2 + WSAT2+0.2);

W2 = (WC2 + WS2 + WSAT2+0.1) ./ ...

(WC1 + WS1 + WSAT1 + WC2 + WS2 + WSAT2+0.2);

%% Naive fusion

R = W1.\*Isharp+W2.\*Igamma;

figure('name', 'Naive Fusion');

imshow([I, Iwb, Isharp, Igamma, R]);

%% Multi scale fusion.

img1 = Isharp;

img2 = Igamma;

% calculate the gaussian pyramid

level = 10;

Weight1 = gaussian\_pyramid(W1, level);

Weight2 = gaussian\_pyramid(W2, level);

% calculate the laplacian pyramid

% input1

R1 = laplacian\_pyramid(Isharp(:, :, 1), level);

G1 = laplacian\_pyramid(Isharp(:, :, 2), level);

B1 = laplacian\_pyramid(Isharp(:, :, 3), level);

% input2

R2 = laplacian\_pyramid(Igamma(:, :, 1), level);

G2 = laplacian\_pyramid(Igamma(:, :, 2), level);

B2 = laplacian\_pyramid(Igamma(:, :, 3), level);

% fusion

for k = 1 : level

Rr{k} = Weight1{k} .\* R1{k} + Weight2{k} .\* R2{k};

Rg{k} = Weight1{k} .\* G1{k} + Weight2{k} .\* G2{k};

Rb{k} = Weight1{k} .\* B1{k} + Weight2{k} .\* B2{k};

end

% reconstruct & output

R = pyramid\_reconstruct(Rr);

G = pyramid\_reconstruct(Rg);

B = pyramid\_reconstruct(Rb);

fusion = cat(3, R, G, B);

figure('name', 'Multi scale fusion');

imshow([I, fusion]);