Color Balance and Fusion for Underwater Image Enhancement

* **Abstract**

We introduce an effective technique to enhance the images captured underwater and degraded due to the medium scattering and absorption. Our method is a single image approach that does not require specialized hardware or knowledge about the underwater conditions or scene structure. It builds on the blending of two images that are directly derived from a color compensated and white-balanced version of the original degraded image. The two images to fusion, as well as their associated weight maps, are defined to promote the transfer of edges and color contrast to the output image. To avoid that the sharp weight map transitions create artifacts in the low frequency components of the reconstructed image, we also adapt a multiscale fusion strategy. Our extensive qualitative and quantitative evaluation reveals that our enhanced images and videos are characterized by better exposedness of the dark regions, improved global contrast, and edges sharpness. Our validation also proves that our algorithm

is reasonably independent of the camera settings, and improves the accuracy of several image processing applications, such as image segmentation and key point matching.

Diagram

Description automatically generated

* **USE:**

Underwater imaging has also been an important source of interest in different branches of technology and scientific research

1. Inspection of underwater infrastructures
2. Marine cables
3. Detection of man made objects
4. Control of underwater vehicles
5. Marine biology research
6. Archeology

**Our paper Approach:**

Our approach builds on the fusion of multiple inputs, but derives the two inputs to combine by correcting the contrast and by sharpening a white-balanced version of a single native input image. The white balancing stage aims at removing the color cast induced by underwater light scattering, so as to produce a natural appearance of the sub-sea images. The multi-scale implementation of the fusion process results in an artifact-free blending.

**Background Knowledge and Previous Work**

1. **Light Propagation in Underwater**

First, the amount of light available under water, depends on several factors. The interaction between the sun light and the sea surface is affected by the time of the day (which influences the light incidence angle), and by the shape of the interface between air and water (rough vs. calm sea). The diving location also directly impacts the available light, due to a location-specific color cast: deeper seas and oceans induce green and blue casts, tropical waters appear cyan, while protected reefs are characterized by high visibility. Other than the light goes several hundred times denser in water than in air.

1. **Related Work**

**1: Using Specialized Hardware**

the divergent-beam underwater Lidar imaging (UWLI) system [10] uses an optical/laser-sensing technique to capture turbid underwater images. Unfortunately, these complex acquisition systems are very expensive, and power consuming.

**2: Polarization-based Methods**

These approaches use several images of the same scene captured with different degrees of polarization, as obtained by rotating a polarizing filter fixed to the camera. polarization techniques are not applicable to video acquisition.

**3: Multiple images or a rough approximation of the scene model**

*Deep Photo* system [17] is able to restore images by employing the existing georeferenced digital terrain and urban 3D models. Since this additional information (images and depth approximation) is generally not available, these methods are impractical for common users.

**4: Similarities between light propagation in fog and under water**

**5: Dark Channel Prior (DCP)**

Recently, several algorithms that specifically restore underwater images based on Dark Channel Prior (DCP) have been introduced. The DCP has initially been proposed for outdoor scenes dehazing.

* **UNDERWATER WHITE BALANCE :**

In our approach, white balancing aims at compensating for the color cast caused by the selective absorption of colors with depth, while image fusion is considered to enhance the edges and details of the scene, to mitigate the loss of contrast resulting from back scattering.

**Explanation of White-Balancing:**

White-balancing aims at improving the image aspect, primarily by removing the undesired color castings due to various illumination or medium attenuation properties. important problem is the green-bluish appearance that needs to be rectified.

After this we see in water images there is an error in red pixels and the image is more bluish.

Graphical user interface, application

Description automatically generated

* **Gray world algorithm :**

For this we use grey world algorithm:

Gray world algorithm assumes that the average reflectance in the scene is achromatic. Hence, the illuminant color distribution is simply estimated by averaging each channel independently.

To compensate for the loss of red channel, we build on the

four following observations/principles:

1. The green channel is relatively well preserved under water, compared to the red and blue ones.
2. The green channel is the one that contains opponent color information compared to the red channel, and it is thus especially important to compensate for the stronger attenuation induced on red, compared to green.
3. The compensation should be proportional to the difference between the mean green and the mean red values because, under the Gray world assumption (all channels have the same mean value before attenuation), this difference reflects the disparity/unbalance between red and green attenuation.
4. To avoid saturation of the red channel during the Gray World step that follows the red loss compensation, the enhancement of red should primarily affect the pixels with small red channel values, and should not change pixels that already include a significant red component.

**Steps**:

**1: Converting image into grey :**

**Code:**

We take a RGB image as input and convert it to grayscale and store it in another variable

% , so we can get the mean luminance (the intensity of light

%emitted from a surface per unit area in a given direction)

% (im2double) function is used for persesion by converting image from unit08

% into double format.

% (rgb2gray) is used to convert rgb into greyscale

rgbImage=imread('7.jpg');

rgbImage=im2double(rgbImage);

grayImage = rgb2gray(rgbImage);

**2: Extract the individual red, green, and blue color channels.**

**%% White Balancing**

**Code:**

% (rgbImage(:, :, 1)) function is used for extracting RGB channels

redChannel = rgbImage(:, :, 1);

greenChannel = rgbImage(:, :, 2);

blueChannel = rgbImage(:, :, 3);

**3: % Taking mean of R,G,B and Grey channel**

**Code:**

% function (mean2) is used for (taking mean of an image)

meanR = mean2(redChannel);

meanG = mean2(greenChannel);

meanB = mean2(blueChannel);

meanGray = mean2(grayImage);

**4: % Make all channels have the same mean**

**Code:**

% (double) function is used to clear any round-off errors

redChannel = (double(redChannel) \* meanGray / meanR);

greenChannel = (double(greenChannel) \* meanGray / meanG);

blueChannel = (double(blueChannel) \* meanGray / meanB);

**5: % redChannel and blueChannel Correction**

**Code:**

% **equation : *Irc(x)* = *Ir(x)* - *α.(* ¯*Ig* − ¯*Ir) .Ig(x).(*1 − *Ir(x))***

% a = Constant peremeter = 0.3 , ¯Ig = mean of Ig , ¯Ir = mean of Ir

% Irc = compensated red channel

% **equation : *Ibc(x)* = *Ib(x)* + *α.(* ¯*Ig* − ¯*Ib) .Ig(x).(*1 − *Ib(x))***

% a = Constant peremeter , ¯Ig = mean of Ig , ¯Ib = mean of Ib

% Ibc = compensated blue channel

redChannel=redChannel-0.3\*(meanG-meanR).\*greenChannel.\*(1-redChannel);

blueChannel=blueChannel+0.3\*(meanG-meanB).\*greenChannel.\*(1-blueChannel);

**6: % Recombine separate color channels into a single, true color RGB image :**

**Code:**

% (cat) function is used to combine all r,g,b channels into one.

% (figure) function creates a new figure object using default property values.

% (subplot) function is used to plot image in frame

% (title) is used to give the name to figure object

figure('PropertyName',PropertyValue,...)

rgbImage\_white\_balance = cat(3, redChannel, greenChannel, blueChannel);

figure('Name','Color Enhancement');

subplot(221);

imshow(redChannel);

title('Suppressed Red Channel');

subplot(222);

imshow(blueChannel);

title('Enhanced Blue Channel');

subplot(223);

imshow(greenChannel);

title('Green Channel');

subplot(224);

imshow(rgbImage\_white\_balance);

title('After White balance');

* **Gamma Correction :**

Gamma correction aims at correcting the global contrast and is relevant since, in general, white balanced underwater images tend to appear too bright. This correction increases the difference between darker/lighter regions at the cost of a loss of details in the under-/over-exposed regions.

**Code:**

% (iamadjust ) function is to Adjust the contrast, specifying a gamma value

% of less than 1 (0.5). Notice that in the call to imadjust, the example

% specifies the data ranges of the input and output images as empty matrices.

% When you specify an empty matrix, imadjust uses the default range of [0,1].

% In the example, both ranges are left empty. This means that gamma correction

% is applied without any other adjustment of the data.

I = imadjust(rgbImage\_white\_balance,[],[],0.5);

* **Sharpening using un sharp masking principle :**

we follow the unsharp masking principle, in the sense that we blend a blurred or unsharp (here Gaussian filtered) version of the image with the image to sharpen. The typical formula for unsharp masking defines the sharpened image **S as S = I + β(I − G ∗ I),** where *I* is the image to sharpen (in our case the white balanced image), *G* ∗ *I* denotes the Gaussian filtered version of *I*, and *β* is a parameter. In practice, the selection of *β* is not trivial. A small *β* fails to sharpen *I*, but a too large *β* results in over-saturated regions, with brighter highlights and darker shadows.

**Code:**

% (imsharpen) function is used to sharpen the combined channel image

J= imsharpen (rgbImage\_white\_balance); (Easy )

OR

**Formula = S as S = I + β(I − G ∗ I),**

%J=(rgbImage\_white\_balance+(rgbImage\_white\_balance - imgaussfilt(rgbImage\_white\_balance))); (According to research paper )

figure('Name','Step I-III');

subplot(221);

imshow(rgbImage);

title('Original Image');

subplot(222);

imshow(rgbImage\_white\_balance);

title('I. White Balance');

subplot(223);

imshow(I);

title('II. Gamma Corrected');

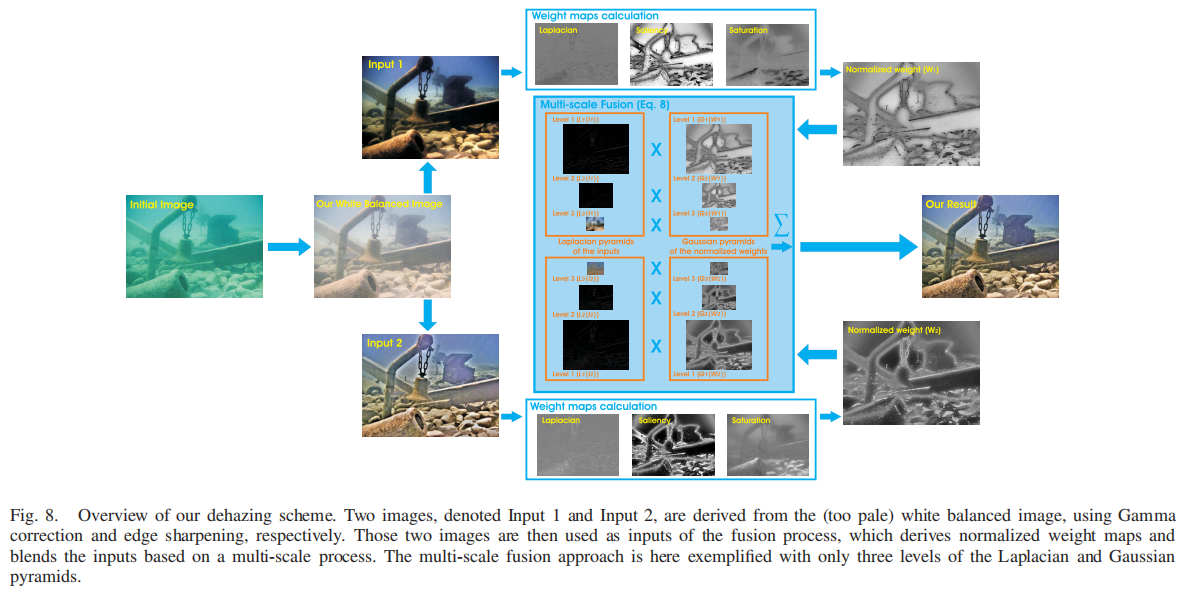
subplot(224);

imshow(J);

title('III. Sharpened');

This second input (sharpening) primarily helps in reducing the degradation caused by scattering.

* **Multi-Scale Fusion :**

****

**Code :**

%% Image Fusion using wavelet transform

% function =>xfus = wfusimg(x1,x2,wname,level,afusmeth,dfusmeth)

% The principle of image fusion using wavelets is to merge the wavelet

% decompositions of the two original images using fusion methods applied to

% approximations coefficients and details coefficients.

XFUS = wfusimg(I,J,'sym4',3,'max','max');

figure('Name','Final Comparison');

subplot(121);

imshow(rgbImage);

title('Original');

subplot(122);

imshow((histeq(XFUS)));

title('IV. Wavelet fusion');

* **Out Put:**

**Graphical user interface, application, PowerPoint

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

Graphical user interface, application, PowerPoint

Description automatically generated

* **Conclusion :**

We have presented an alternative approach to enhance underwater videos and images. Our strategy builds on the fusion principle and does not require additional information than the single original image. We have shown in our experiments that our approach is able to enhance a wide range of underwater images (e.g. different cameras, depths, light conditions) with high accuracy, being able to recover important faded features and edges. Moreover, for the first time, we demonstrate the utility and relevance of the proposed image enhancement technique for several challenging underwater

computer vision applications.