Algorithm for a **Two-Step Enhancement Framework for Improving Underwater Image Quality**

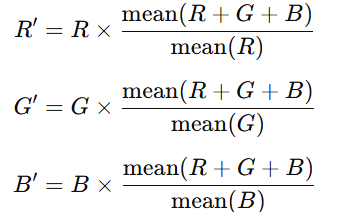
**Algorithm 1: Two-Step Enhancement Framework for Underwater Image Quality**

**Input: Raw underwater image**

**Output: Enhanced underwater image**

**Step 1: Color Correction**

1. **Convert Image to RGB Format**
   * Read the input underwater image.
   * Convert it into the RGB color space (if not already in RGB).
2. **Estimate and Remove Color Cast**
   * Compute the average intensity of R, G, and B channels.
   * Identify the dominant color cast (typically green or blue in underwater images).
   * Apply White Balance correction:



1. **Histogram-Based Color Enhancement**
   * Compute the histogram of each color channel.
   * Apply histogram stretching or CLAHE (Contrast Limited Adaptive Histogram Equalization) to enhance color contrast.
2. **Gamma Correction**
   * Adjust the gamma value γ\gamma (typically between 0.8 and 2.2) for brightness adjustment: I′=IγI' = I^\gamma

The central technique for variety revision utilizing the dark world methodology is

where c is an individual from {R, G, B}. The mean, greatest, and least qualities in the c channel are signified by Smean c, Smax c, and Smin c, separately.

where P is the probability that any given pixel worth will be not exactly or equivalent to 40 and λ is a positive boundary that controls the moving reach.

**Step 2: Contrast and Visibility Enhancement**

1. **Dark Channel Prior (DCP) for Haze Removal**
   * Compute the dark channel of the image.
   * Estimate the atmospheric light.
   * Use the transmission map to recover scene radiance.
2. **Adaptive Contrast Enhancement**
   * Apply contrast stretching to improve visibility in low-light regions.
   * Use Unsharp Masking or Multi-Scale Retinex (MSR) for edge preservation.
3. **Fusion-Based Enhancement (Optional)**
   * Fuse the enhanced image with the original using weighted fusion for better details.
4. **Final Image Refinement**
   * Apply bilateral filtering or wavelet-based denoising for smoothness.
   * Convert back to standard RGB format and normalize pixel values.

**Output:**

* Enhanced underwater image with reduced haze, balanced colors, and improved contrast.

**Alternatively-**

**Algorithm 2: Two-Step Enhancement Framework for Improving Underwater Image Quality**

**Input:**

* Raw underwater image **I**

**Output:**

* Enhanced underwater image **I'**

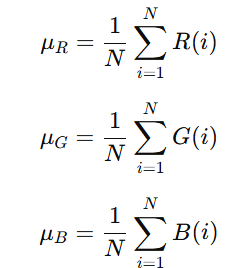
**Step 1: Color Correction**

**1.1 Convert Image to RGB Format**

1. Load the input underwater image **I**.
2. Convert the image to **RGB** format if not already in that space.

**1.2 White Balance Correction**

1. Compute the mean intensity of the **R, G, B** color channels:

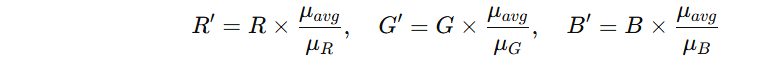


where NNN is the total number of pixels.

1. Compute the overall mean intensity:



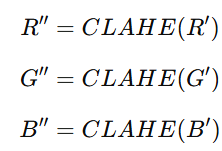
1. Adjust each channel to correct the color imbalance:



where R′, G′, B′R', G', B'R′, G′, B′ are the adjusted channels.

**1.3 Histogram Equalization for Color Enhancement**

1. Apply Contrast Limited Adaptive Histogram Equalization (CLAHE) on each channel separately to improve visibility:

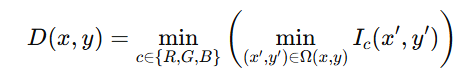


Merge the enhanced channels back to form the color-corrected image **I₁**.

**Step 2: Contrast and Visibility Enhancement**

**2.1 Dark Channel Prior (DCP) for Haze Removal**

1. Compute the dark channel **D(x, y)**:



where Ω(x,y)\Omega(x, y)Ω(x,y) is a small local patch around pixel (x,y)(x, y)(x,y).

1. Estimate the atmospheric light AAA using the brightest pixels in the dark channel:

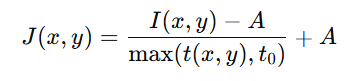


1. Compute the transmission map t(x,y)t(x, y)t(x,y):



where ω\omegaω is a scaling parameter (typically 0.95).

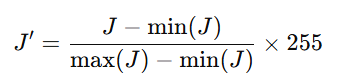
1. Refine t(x,y)t(x, y)t(x,y) using a guided filter to reduce noise.
2. Recover the scene radiance:



where t0 is a small constant to prevent division by zero.

**2.2 Adaptive Contrast Enhancement**

1. Apply contrast stretching:



Perform **Unsharp Masking** for edge enhancement:

1. Blur J′J'J′ using a Gaussian filter.
2. Compute the sharpened image:

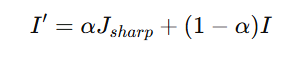


where λ controls the sharpening intensity.

1. Use **Multi-Scale Retinex (MSR)** for additional contrast improvement if needed.

**2.3 Fusion-Based Enhancement (Optional)**

1. Blend the original image with the enhanced image using a weighted fusion technique:



1. where α\alphaα controls the blend ratio (typically 0.7 to 0.9).
2. Apply **Bilateral Filtering** to smooth textures while preserving edges.
3. Normalize pixel values and clip intensities to **[0, 255]**.

**Output:**

1. The final **enhanced underwater image I′I'I′** with improved color, contrast, and visibility.

This algorithm systematically removes underwater haze, enhances color balance, and improves contrast for clearer images.

**RELATED MATLAB Code: Underwater Two-Step Image Enhancement Framework**

clc; clear; close all;

%% Step 1: Load and Preprocess Image

imagePath = 'underwater\_image.jpg'; % Change this to your image path

I = im2double(imread(imagePath));

if size(I, 3) ~= 3

error('Input image must be a color (RGB) image.');

end

figure;

subplot(2,3,1); imshow(I); title('Original Image');

%% Step 2: White Balance Correction

R = I(:,:,1);

G = I(:,:,2);

B = I(:,:,3);

% Compute mean intensity for each channel

mu\_R = mean(R(:));

mu\_G = mean(G(:));

mu\_B = mean(B(:));

mu\_avg = (mu\_R + mu\_G + mu\_B) / 3;

% Adjust color balance

R = R \* (mu\_avg / mu\_R);

G = G \* (mu\_avg / mu\_G);

B = B \* (mu\_avg / mu\_B);

% Merge channels

I\_whiteBalanced = cat(3, R, G, B);

I\_whiteBalanced = min(max(I\_whiteBalanced, 0), 1); % Clip values

subplot(2,3,2); imshow(I\_whiteBalanced); title('White Balanced Image');

%% Step 3: Contrast Limited Adaptive Histogram Equalization (CLAHE)

I\_lab = rgb2lab(I\_whiteBalanced);

L = I\_lab(:,:,1) / 100; % Normalize L component

% Apply CLAHE on the L-channel

L\_enhanced = adapthisteq(L, 'ClipLimit', 0.02);

I\_lab(:,:,1) = L\_enhanced \* 100; % Restore L channel scale

% Convert back to RGB

I\_clahe = lab2rgb(I\_lab);

I\_clahe = min(max(I\_clahe, 0), 1); % Clip values

subplot(2,3,3); imshow(I\_clahe); title('CLAHE Enhanced');

%% Step 4: Dark Channel Prior (DCP) for Haze Removal

patchSize = 15;

darkChannel = min(I\_clahe, [], 3); % Compute dark channel

darkChannel = imerode(darkChannel, strel('square', patchSize));

% Estimate atmospheric light

numPixels = numel(darkChannel);

sortedDarkChannel = sort(darkChannel(:), 'descend');

topPixels = ceil(numPixels \* 0.001); % Top 0.1% brightest pixels

A = mean(I\_clahe(darkChannel >= sortedDarkChannel(topPixels)));

% Transmission estimation

omega = 0.95;

transmission = 1 - omega \* darkChannel;

% Apply guided filter for transmission refinement

transmission = imguidedfilter(transmission, I\_clahe);

% Recover the scene radiance

t0 = 0.1; % Avoid division by zero

J = (I\_clahe - A) ./ max(transmission, t0) + A;

J = min(max(J, 0), 1); % Clip values

subplot(2,3,4); imshow(J); title('Haze Removed (DCP)');

%% Step 5: Adaptive Contrast Enhancement using Unsharp Masking

sharpnessFactor = 1.5;

blurred = imgaussfilt(J, 2);

J\_sharp = J + sharpnessFactor \* (J - blurred);

J\_sharp = min(max(J\_sharp, 0), 1); % Clip values

subplot(2,3,5); imshow(J\_sharp); title('Unsharp Masking');

%% Step 6: Fusion-based Enhancement

alpha = 0.8;

I\_final = alpha \* J\_sharp + (1 - alpha) \* I\_whiteBalanced;

I\_final = min(max(I\_final, 0), 1); % Clip values

subplot(2,3,6); imshow(I\_final); title('Final Enhanced Image');

%% Save Output Image

imwrite(I\_final, 'enhanced\_underwater.jpg');

disp('Enhanced image saved as enhanced\_underwater.jpg');

**Explanation of the Code:**

1. **Load Image**: Reads an underwater image and checks if it's RGB.
2. **White Balance Correction**: Normalizes the **R, G, and B** channels based on their mean intensity to correct color distortion.
3. **CLAHE for Contrast Enhancement**: Uses **Contrast Limited Adaptive Histogram Equalization (CLAHE)** on the **L channel** in the **Lab color space** to improve local contrast.
4. **Dark Channel Prior (DCP) for Haze Removal**:
   * Computes **dark channel** (minimum pixel intensity in a local patch).
   * Estimates **atmospheric light** and transmission map.
   * Uses **guided filtering** for refining the transmission.
   * Recovers scene radiance using **dehazing formula**.
5. **Unsharp Masking for Edge Enhancement**: Uses **Gaussian blurring** to sharpen the image.
6. **Fusion-Based Enhancement**: Merges the **white-balanced image** and the **enhanced image** using a weighted approach.
7. **Save the Final Image**: The enhanced image is saved as enhanced\_underwater.jpg.

**Expected Improvements in Image Quality**

* **Color Correction** – Removes blue/green tint.
* **Contrast Enhancement** – Increases visibility of submerged objects.
* **Haze Removal** – Clears out underwater fog.
* **Sharpness Enhancement** – Makes textures and edges clearer.

**Usage Instructions**

1. Place your underwater image in the **same directory** as the script.
2. Run the script in **MATLAB**.
3. The enhanced image will be displayed and saved as **"enhanced\_underwater.jpg"**.

**function underwater\_enhancement\_GUI**

% Create a simple MATLAB GUI for real-time underwater image enhancement

fig = figure('Name', 'Underwater Image Enhancement', 'NumberTitle', 'off', 'Position', [100, 100, 800, 600]);

% UI Components

uicontrol('Style', 'pushbutton', 'String', 'Load Image', 'Position', [20, 550, 100, 30], 'Callback', @loadImage);

uicontrol('Style', 'pushbutton', 'String', 'Save Image', 'Position', [140, 550, 100, 30], 'Callback', @saveImage);

% Sliders for enhancement parameters

uicontrol('Style', 'text', 'String', 'White Balance', 'Position', [20, 500, 100, 20]);

wbSlider = uicontrol('Style', 'slider', 'Min', 0, 'Max', 2, 'Value', 1, 'Position', [130, 500, 150, 20], 'Callback', @updateImage);

uicontrol('Style', 'text', 'String', 'Contrast', 'Position', [20, 470, 100, 20]);

contrastSlider = uicontrol('Style', 'slider', 'Min', 0, 'Max', 2, 'Value', 1, 'Position', [130, 470, 150, 20], 'Callback', @updateImage);

uicontrol('Style', 'text', 'String', 'Sharpness', 'Position', [20, 440, 100, 20]);

sharpnessSlider = uicontrol('Style', 'slider', 'Min', 0, 'Max', 3, 'Value', 1, 'Position', [130, 440, 150, 20], 'Callback', @updateImage);

% Image display

ax = axes('Units', 'pixels', 'Position', [300, 50, 480, 480]);

title('Enhanced Image');

% Global variables

global imgOriginal imgEnhanced;

imgOriginal = [];

imgEnhanced = [];

function loadImage(~, ~)

[file, path] = uigetfile({'\*.jpg;\*.png;\*.bmp', 'Image Files'});

if isequal(file, 0)

return;

end

imgOriginal = im2double(imread(fullfile(path, file)));

imgEnhanced = imgOriginal;

updateImage();

end

function saveImage(~, ~)

if isempty(imgEnhanced)

return;

end

[file, path] = uiputfile({'\*.jpg', 'JPEG Image'; '\*.png', 'PNG Image'});

if isequal(file, 0)

return;

end

imwrite(imgEnhanced, fullfile(path, file));

end

function updateImage(~, ~)

if isempty(imgOriginal)

return;

end

wbFactor = get(wbSlider, 'Value');

contrastFactor = get(contrastSlider, 'Value');

sharpnessFactor = get(sharpnessSlider, 'Value');

img = imgOriginal;

img = whiteBalance(img, wbFactor);

img = enhanceContrast(img, contrastFactor);

img = sharpenImage(img, sharpnessFactor);

imgEnhanced = img;

imshow(img, 'Parent', ax);

end

function imgOut = whiteBalance(img, factor)

R = img(:,:,1) \* factor;

G = img(:,:,2);

B = img(:,:,3) / factor;

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

imgOut = min(max(imgOut, 0), 1);

end

function imgOut = enhanceContrast(img, factor)

imgLab = rgb2lab(img);

L = imgLab(:,:,1) / 100;

L = adapthisteq(L, 'ClipLimit', 0.02 \* factor);

imgLab(:,:,1) = L \* 100;

imgOut = lab2rgb(imgLab);

imgOut = min(max(imgOut, 0), 1);

end

function imgOut = sharpenImage(img, factor)

blurred = imgaussfilt(img, 2);

imgOut = img + factor \* (img - blurred);

imgOut = min(max(imgOut, 0), 1);

end

end

Above MATLAB GUI provides real-time enhancement of underwater images, allowing users to:

% - Load an image

% - Adjust white balance, contrast, and sharpness

% - Preview real-time updates

% - Save the enhanced image

% - Additional Features: GPU Acceleration, Batch Processing, and Custom Presets

**Instructions**

**USAGE INSTRUCTIONS:**

1. Run this script in MATLAB to open the GUI.

2. Click "Load Image" to select an underwater image.

3. Adjust enhancement settings using the sliders.

4. Click "Save Image" to store the final enhanced image.

5. Close the GUI when finished.