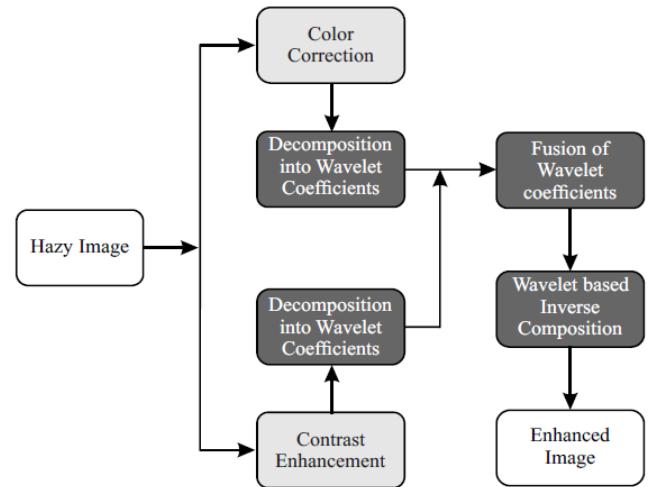


Under Water Image Processing using Wavelet Fusion

Taking reference from IEEE paper “Underwater Image Enhancement by Wavelet Based Fusion” a model to has been implemented to enhance an underwater video captured during the testing of “Autonomous Water Rover”. The flow of the process is as follows :-

- Step 1: Import image
- Step 2: Contrast Adjustment using CLAHE
- Step 3: Colour Correction using histogram stretching of V in HSV
- Step 4: Wavelet Fusion of images from step 2 and 3.
 - 2 Level decomposition
 - db2 wavelet
 - ‘max’ for approximate and ‘min’ for details
- Step 5: Get enhanced image



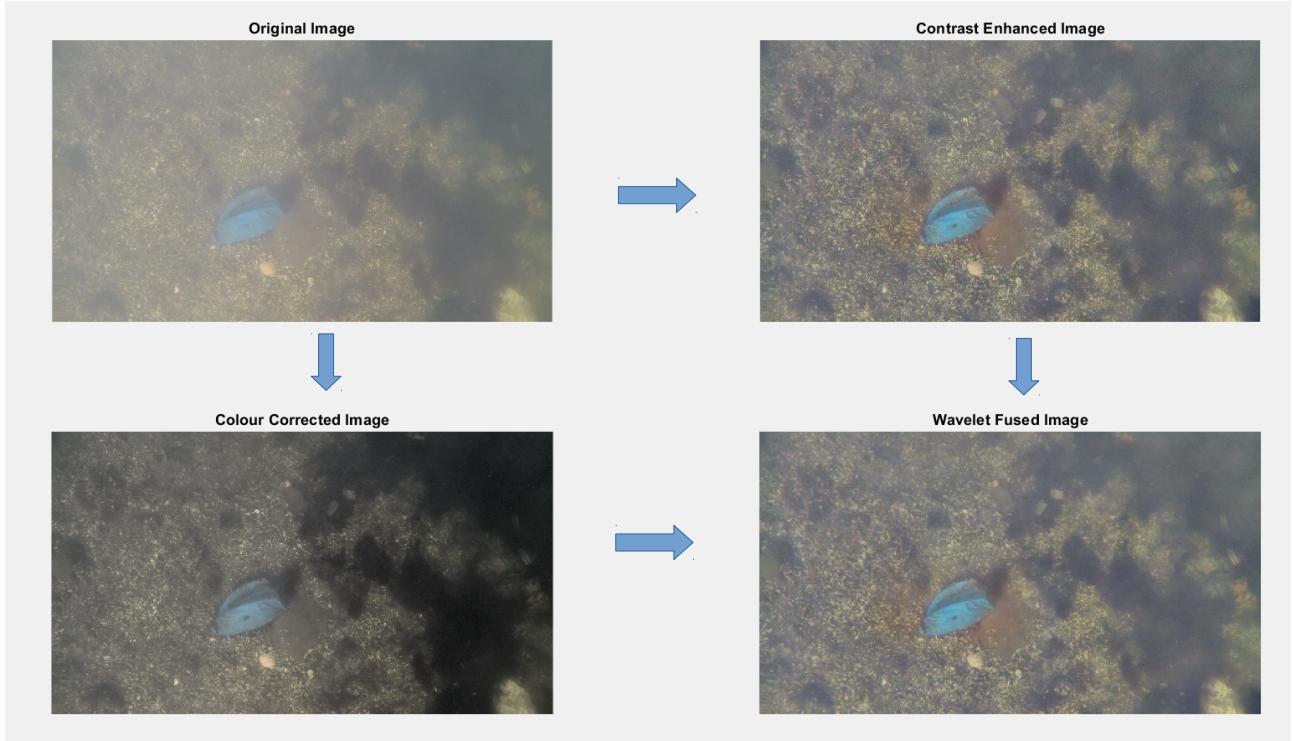
Results

Original



Processed





Code for Video input, enhancement and conversion to New Video

```

clc; % Clear the command window.
close all; % Close all figures (except those of imtool.)
imtool close all; % Close all imtool figures.
clear; % Erase all existing variables.
workspace; % Make sure the workspace panel is showing.
fontSize = 22;

% First get the folder that it lives in.
folder = 'C:\Users\Chirag\Desktop\ZEE24';
movieFullName = fullfile(folder, 'uw2.mp4');
% Check to see that it exists.
if ~exist(movieFullName, 'file')
    strErrorMessage = sprintf('File not found:\n%s\nYou can choose a new one, or cancel', movieFullName);
    response = questdlg(strErrorMessage, 'File not found', 'OK - choose a new movie.', 'Cancel', 'OK - choose a new movie.');
    if strcmpi(response, 'OK - choose a new movie.')
        [baseFileName, folderName, FilterIndex] = uigetfile('.avi');
        if ~isequal(baseFileName, 0)
            movieFullName = fullfile(folderName, baseFileName);
        else
            return;
        end
    else
        return;
    end
end

try
    videoObject = VideoReader(movieFullName)
    % Determine how many frames there are.
    numberOfframes = videoObject.NumberOfFrames;
    vidHeight = videoObject.Height;
    vidWidth = videoObject.Width;

    numberOfFramesWritten = 0;
    % Prepare a figure to show the images in the upper half of the screen.
    figure;
    % Enlarge figure to full screen.

```

```

set(gcf, 'units','normalized','outerposition',[0 0 1 1]);
% Ask user if they want to write the individual frames out to disk.
promptMessage = sprintf('Do you want to save the individual frames out to individual disk files?');
button = questdlg(promptMessage, 'Save individual frames?', 'Yes', 'No', 'Yes');
if strcmp(button, 'Yes')
    writeToDisk = true;
    % Extract out the various parts of the filename.
    [folder, baseFileName, extentions] = fileparts(movieFullFileName);
    % Make up a special new output subfolder for all the separate
    % movie frames that we're going to extract and save to disk.
    % (Don't worry - windows can handle forward slashes in the folder name.)
    folder = pwd; % Make it a subfolder of the folder where this m-file lives.
    outputFolder = sprintf('%s/Movie Frames from %s', folder, baseFileName);
    % Create the folder if it doesn't exist already.
    if ~exist(outputFolder, 'dir')
        mkdir(outputFolder);
    end
else
    writeToDisk = false;
end

% Loop through the movie, writing all frames out.
% Each frame will be in a separate file with unique name.
meanGrayLevels = zeros(numberOfFrames, 1);
meanRedLevels = zeros(numberOfFrames, 1);
meanGreenLevels = zeros(numberOfFrames, 1);
meanBlueLevels = zeros(numberOfFrames, 1);
for frame = 10:10:5300
    % Extract the frame from the movie structure.
    thisFrame = read(videoObject, frame);
    % Display it
    hImage = subplot(1, 2, 1);
    image(thisFrame);
    caption = sprintf('Frame %4d of %d.', frame/10, numberOfFrames/10);
    title(caption, 'FontSize', fontSize);
    drawnow; % Force it to refresh the window.
    % Write the image array to the output file, if requested.
    if writeToDisk
        % Construct an output image file name.
        outputBaseFileName = sprintf('%4.4d.png', frame/10);
        outputFullName = fullfile(outputFolder, outputBaseFileName);
        %Enhance the image
        thisFrame=uint8(uwProcess(thisFrame));
        imwrite(thisFrame, outputFullName, 'png');
    end
end
% Update user with the progress. Display in the command window.
if writeToDisk
    progressIndication = sprintf('Wrote frame %4d of %d.', frame, numberOfFrames);
else
    progressIndication = sprintf('Processed frame %4d of %d.', frame, numberOfFrames);
end
disp(progressIndication);
% Increment frame count (should eventually = numberOfFrames
% unless an error happens).
numberOfFramesWritten = numberOfFramesWritten + 1;
% Alert user that we're done.
if writeToDisk
    finishedMessage = sprintf('Done! It wrote %d frames to folder\n"%s"', numberOfFramesWritten, outputFolder);
else
    finishedMessage = sprintf('Done! It processed %d frames of\n"%s"', numberOfFramesWritten, movieFullFileName);
end
disp(finishedMessage); % Write to command window.
uiwait(msgbox(finishedMessage)); % Also pop up a message box.
% Exit if they didn't write any individual frames out to disk.
if ~writeToDisk

```

```

    return;
end
% Ask user if they want to read the individual frames from the disk,
% that they just wrote out, back into a movie and display it.
promptMessage = sprintf('Do you want to recall the individual frames\nback from disk into a movie?\n(This will take several seconds.)');
button = questdlg(promptMessage, 'Recall Movie?', 'Yes', 'No', 'Yes');
if strcmp(button, 'No')
    return;
end
% Create a VideoWriter object to write the video out to a new, different file.
writerObj = VideoWriter('ogiuw.avi');
%Slowdown the movie
writerObj.FrameRate=10;
open(writerObj);

% Read the frames back in from disk, and convert them to a movie.
% Preallocate recalledMovie, which will be an array of structures.
% First get a cell array with all the frames.
allTheFrames = cell(530,1);
allTheFrames(:) = {zeros(vidHeight, vidWidth, 3, 'uint8')};
% Next get a cell array with all the colormaps.
allTheColorMaps = cell(530,1);
allTheColorMaps(:) = {zeros(256, 3)};
% Now combine these to make the array of structures.
recalledMovie = struct('cdata', allTheFrames, 'colormap', allTheColorMaps)
for frame = 1 : 530
    % Construct an output image file name.
    outputBaseFileName = sprintf('%4.4d.png', frame);
    outputFullFileName = fullfile(outputFolder, outputBaseFileName);
    % Read the image in from disk.
    thisFrame = imread(outputFullFileName);
    % Convert the image into a "movie frame" structure.
    recalledMovie(frame) = im2frame(thisFrame);
    % Write this frame out to a new video file.
    writeVideo(writerObj, thisFrame);
end
close(writerObj);
% Create new axes for our movie.
subplot(1, 2, 2);
axis off; % Turn off axes numbers.
title('Movie recalled from disk', 'FontSize', fontSize);
% Play the movie in the axes.
movie(recalledMovie);
% Note: if you want to display graphics or text in the overlay
% as the movie plays back then you need to do it like I did at first
% (at the top of this file where you extract and imshow a frame at a time.)
msgbox('Done with this demo!');

```

catch ME

```

    % Some error happened if you get here.
    strErrorMessage = sprintf('Error extracting movie frames from:\n\n%s\n\nError: %s\n\n', movieFullFileName, ME.message);
    uiwait(msgbox(strErrorMessage));
end

```

Function For Wavelet enhancement

```

function imgnew=uwProcess(img)
%Convert Image to HSV for Value Stretching
imgHSV=rgb2hsv(img);
%Extract and do histogram stretching of Value plane Colour correction --> img1
v=imgHSV(:,:,3);
amin1=double(min(min(v,[],2)));
amax1=double(max(max(v,[],2)));
arange=amax1-amin1;
for i = 1:1:1080
    for j = 1:1:1920
        vnew(i,j)=((v(i,j)-amin1)*1)/arange;
    end

```

```

end
imgHSV(:,:,3)=vnew;
img1=hsv2rgb(imgHSV);
%Enhance contrast using CLAHE (contrast adjustment)--> img2
imgr=adapthisteq(img(:,:,1),'clipLimit',0.01,'Distribution','rayleigh');
imgg=adapthisteq(img(:,:,2),'clipLimit',0.01,'Distribution','rayleigh');
imgb=adapthisteq(img(:,:,3),'clipLimit',0.01,'Distribution','rayleigh');
img2(:,:,1)=imgr;
img2(:,:,2)=imgg;
img2(:,:,3)=imgb;

%Fuse CC and CA
inr = wfusimg(img1(:,:,1),img2(:,:,1),'db2',2,'max','min');
ing = wfusimg(img1(:,:,2),img2(:,:,2),'db2',2,'max','min');
inb = wfusimg(img1(:,:,3),img2(:,:,3),'db2',2,'max','min');
imgnew(:,:,1)=inr;
imgnew(:,:,2)=ing;
imgnew(:,:,3)=inb;

```