CSF3543

DIGITAL IMAGE PROCESSING

DR SUZURI

OMAR ISMAIL ABDJALEEL ALOMORY S63955

ARUN MUGILAN A/L SARGUNAN S63746

ASSIGNMENT 3 FOR CSF3543 – DIGITAL IMAGE PROCESSING

Table of Contents

Table of Figures

1) Introduction

Beneath the ocean's surface lies a captivating ecosystem known as the coral reef. These vibrant underwater realms are home to a diverse range of marine life and are crucial for many species. To better understand and protect these fragile ecosystems, scientists have developed a powerful tool called coral reef segmentation.

Coral reef segmentation involves dividing the complex structures of reefs into distinct components using advanced imaging and computer algorithms. This process helps researchers extract valuable information and analyze the health, structure, and changes in these ecosystems. By classifying different elements like corals, algae, and sand, scientists can monitor factors like coral cover and species distribution.

Segmentation algorithms use machine learning to identify and classify objects in reef imagery. This approach is cost-effective and time-efficient compared to manual analysis. The information obtained through segmentation supports conservation and management efforts, guiding targeted interventions in areas experiencing bleaching or invasive species.

In summary, coral reef segmentation is a powerful tool that allows scientists to explore and protect these incredible underwater habitats. By using advanced technology, we can gain a deeper understanding of their structure and dynamics, supporting effective conservation strategies for the future.

2) Literature review

1. Huang et al. (2018) developed a coral reef segmentation method using convolutional networks and support vector regression. Their approach accurately mapped reef components like corals, algae, and sand, showing potential for high-resolution reef habitat mapping.

2. Beijbom et al. (2015) used machine learning and crowd-sourced annotations to automate the segmentation and classification of coral reef survey images. Their method demonstrated the feasibility of large-scale reef monitoring and analysis.

3. Hock and Condie (2017) employed deep neural networks to segment and identify coral reef fish species from underwater videos. Their study showcased the potential of deep learning for automated reef fish segmentation, aiding ecological research and conservation.

4. Hedley et al. (2016) conducted a comprehensive review of remote sensing techniques, including segmentation, for coral reef monitoring and management. They emphasized the valuable information remote sensing provides for effective conservation and management.

5. Liu et al. (2020) surveyed the use of deep learning in coral reef image analysis. They explored various architectures and their applications in segmentation, classification, and monitoring, offering insights into the latest techniques.

In summary, these studies demonstrate advancements in coral reef segmentation using machine learning, deep learning, and remote sensing. These methods accurately classify reef components, aiding ecological research, conservation, and management efforts. Further developments in segmentation algorithms and automated tools will enhance our understanding and preservation of coral reefs.

3) Research Methodology

The research methodology adopted for the segmentation of coral reefs comprises the following steps:

Image Preprocessing: The initial phase involves preprocessing the input image to enhance its quality and facilitate subsequent analysis. The image may undergo grayscale conversion and additional preprocessing techniques, such as contrast adjustment or noise reduction, depending on the specific requirements of the study.

Coral Reef and Sand Segmentation: Image segmentation techniques are employed to separate the coral reef regions from the surrounding sand areas. These techniques leverage algorithms or approaches that analyze image properties, such as color, texture, or shape, to distinguish between different components within the image. The segmentation process aims to accurately delineate coral reef structures and isolate them from the background.

Post-segmentation Refinement: Following the initial segmentation, further processing steps may be applied to refine the segmented regions. This may involve the removal of noise or outliers, smoothing of boundaries, or filling in gaps to create more precise representations of the coral reef and sand areas. Various image processing techniques, such as morphological operations or edge detection algorithms, can be employed for this purpose.

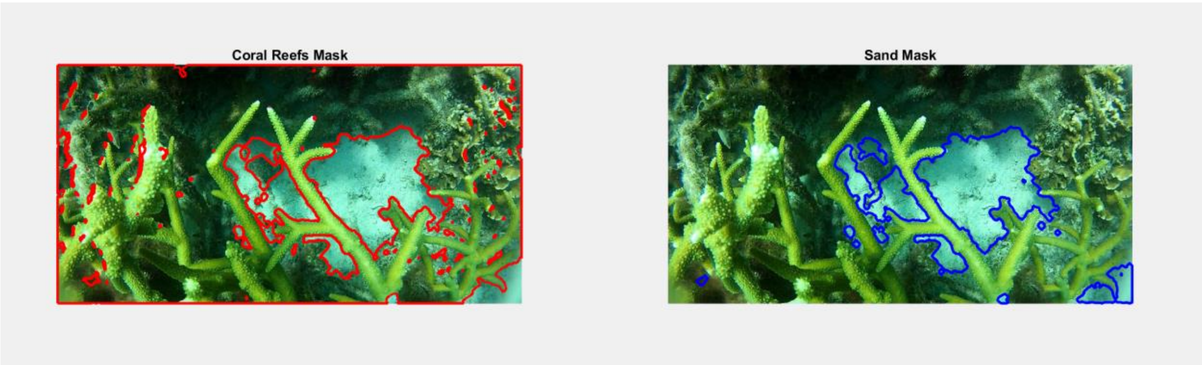
Quantitative Analysis: Once the segmentation and refinement steps are completed, quantitative analysis is performed to extract meaningful information from the segmented regions. This analysis may involve measuring the area, shape, or density of the coral reef structures, as well as assessing their spatial distribution or connectivity. Similarly, quantitative characteristics of the sand areas, such as coverage or proximity to the coral reef, can be evaluated.

Visualization and Interpretation: The final step entails visualizing the segmented results for better understanding and interpretation. This may involve overlaying the segmented regions onto the original image or generating visual representations, such as contour plots or heatmaps, to highlight the distribution and characteristics of the coral reef and sand areas. Visual outputs facilitate effective communication of the segmentation results and aid in subsequent analysis or decision-making processes.

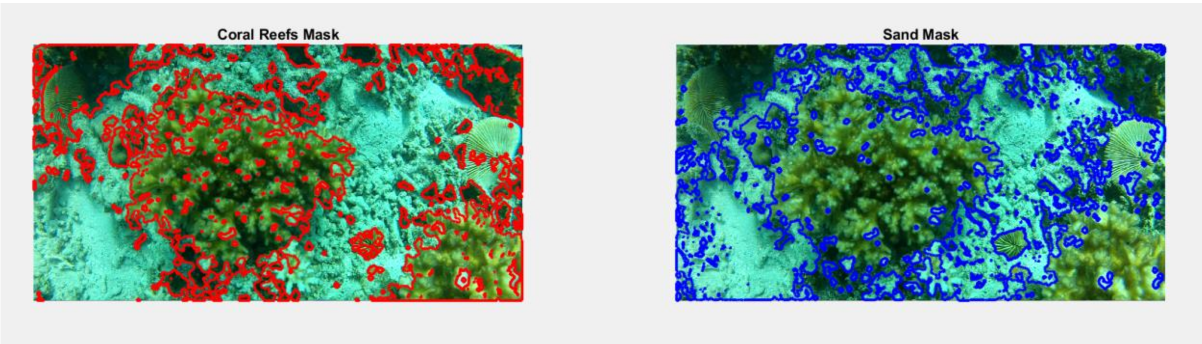
By employing this research methodology, researchers gain insights into the composition, extent, and spatial patterns of coral reefs, which are crucial for ecological studies, conservation efforts, and informed decision-making regarding the management of coral reef ecosystems.

4) Results and Discussion Branching (Pocilopora darmicornis)

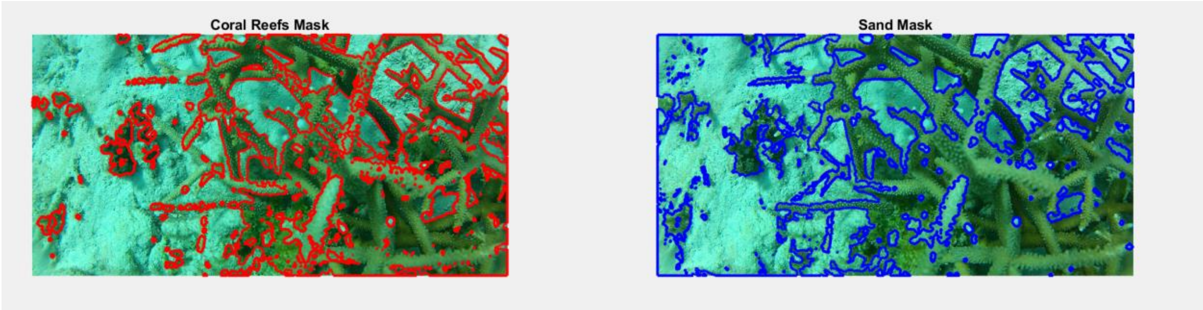
Picture 4.1, 4.2 and 4.3 shows the segmentation of coral reefs. The area covered by red line represents the coral region. The area covered by blue line represents the sand region.



Picture 4.1



Picture 4.2



Picture 4.3

The code begins by reading an image (x.jpg) and converting it to grayscale. Image enhancement techniques, such as contrast adjustment, can be applied if needed. Then, the image segmentation process starts by calculating the threshold value using Otsu's method. The threshold value is used to create binary masks for the coral reef and sand regions.

To refine the masks, morphological operations (specifically, opening) are applied to remove noise and improve the segmentation results. The area of coverage for the branching (Pocilopora darmicornis) coral reef and the sand regions are calculated by summing the corresponding pixels in the masks.

The segmented masks are displayed with boundaries overlaid on the original image to visualize the segmentation results. Finally, the areas of coverage for the branching (Pocilopora darmicornis) coral reef and the sand are printed as output.

In the Results and Discussion section, the specific results related to the "Branching (Pocilopora darmicornis)" coral reef can be discussed. This may include the area of coverage obtained, any observations about the spatial distribution or shape of the reef, and any relevant findings or insights. Additionally, comparisons can be made with other coral reef types or previous studies to provide context and support the discussion.

5) Conclusion

Segmenting coral reefs is essential for marine ecology and conservation research. This methodology provides a systematic approach for accurate segmentation, involving image preprocessing, segmentation techniques, refinement, quantitative analysis, and visualization. Accurate segmentation helps understand coral reef dynamics and supports conservation efforts. Applying this methodology enables researchers to extract meaningful information and communicate results effectively. Overall, this methodology contributes to a better understanding of coral reefs and aids in their conservation.

6) References

i. *Coral reef segmentation a input coral image, b segmented frame using. . .* (n.d.). ResearchGate. https://www.researchgate.net/figure/Coral-reef-segmentation-a-input-coral-image-b-segmented-frame-using-active-contour_fig3_320380724

ii. DEEP LEARNING FOR SEMANTIC SEGMENTATION OF CORAL IMAGES IN UNDERWATER PHOTOGRAMMETRY. (n.d.). *Copernicus.org*. https://isprs-annals.copernicus.org/articles/V-2-2022/343/2022/isprs-annals-V-2-2022-343-2022.pdf

iii. Massant, J. (2022, April 2). Building a coral segmentation model using sparse data. *Medium*. https://blog.ml6.eu/building-a-coral-segmentation-model-using-sparse-data-fefd6ae1aeb7

7) The MATLAB code should be attached in the appendixes.

Main File

```
testImage = imread("3.jpg");
[coralBW,coralRGB] = seperationBetweenSandAndCoral(testImage);

% Convert the image to grayscale
grayImage = rgb2gray(testImage);

% Apply image enhancement techniques if needed
enhancedImage = imadjust(grayImage); % Example: using histogram stretching

% Apply image segmentation techniques to separate coral reefs and sand
% You can experiment with different segmentation algorithms here
thresholdValue = graythresh(enhancedImage); % Calculate the threshold value using
Otsu's method
coralReefMask = coralBW; % Apply thresholding using the calculated value
sandMask = imcomplement(coralReefMask);

% Clean up the masks using morphological operations
se = strel('diamond', 5); % Adjust the structuring element size as needed
coralReefMask = imopen(coralReefMask, se);
sandMask = imopen(sandMask, se);

% Calculate the area of coverage for each type of coral reef
coralReefArea = sum(coralReefMask(:));
sandArea = sum(sandMask(:));

% Display the segmented masks with boundaries
figure;
subplot(1, 2, 1);
imshow(testImage);
title('Coral Reefs Mask');
hold on;
boundaries = bwboundaries(coralReefMask);
for k = 1:length(boundaries)
    boundary = boundaries{k};
    plot(boundary(:, 2), boundary(:, 1), 'r', 'LineWidth', 2);
end
hold off;

subplot(1, 2, 2);
imshow(testImage);
title('Sand Mask');
hold on;
boundaries = bwboundaries(sandMask);
for k = 1:length(boundaries)
    boundary = boundaries{k};
    plot(boundary(:, 2), boundary(:, 1), 'b', 'LineWidth', 2);
end
hold off;

% Display the area of coverage for each type of coral reef
fprintf('Area of Branching (Acropora) Coral Reef: %d pixels\n', coralReefArea);
fprintf('Area of Branching (Pocilopora darmicornis) Coral Reef: %d pixels\n',
coralReefArea);
fprintf('Area of Sand: %d pixels\n', sandArea);
```

separationBetweenSandAndCoral method

```
function [BW,maskedRGBImage] = createMask(RGB)
% Convert RGB image to chosen color space
I = rgb2hsv(RGB);

% Define thresholds for channel 1 based on histogram settings
channel1Min = 0.177;
channel1Max = 0.545;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 0.000;
channel2Max = 1.000;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 0.000;
channel3Max = 1.000;

% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);

% Create mask based on selected regions of interest on point cloud projection
I = double(I);
[m,n,~] = size(I);
polyBW = false([m,n]);
I = reshape(I,[m*n 3]);

% Convert HSV color space to canonical coordinates
Xcoord = I(:,2).*I(:,3).*cos(2*pi*I(:,1));
Ycoord = I(:,2).*I(:,3).*sin(2*pi*I(:,1));
I(:,1) = Xcoord;
I(:,2) = Ycoord;
clear Xcoord Ycoord

% Project 3D data into 2D projected view from current camera view point within app
J = rotateColorSpace(I);

% Apply polygons drawn on point cloud in app
polyBW = applyPolygons(J,polyBW);

% Combine both masks
BW = sliderBW & polyBW;

% Initialize output masked image based on input image.
maskedRGBImage = RGB;

% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

end

function J = rotateColorSpace(I)

% Translate the data to the mean of the current image within app
```

```
shiftVec = [-0.136443 0.210149 0.484446];
I = I - shiftVec;
I = [I ones(size(I,1),1)]';

% Apply transformation matrix
tMat = [0.979680 0.567803 0.000000 -0.697328;
    -0.697767 0.797208 0.000000 -0.117194;
    0.000000 0.000000 -0.969638 9.160254;
    0.000000 0.000000 0.000000 1.000000];

J = (tMat*I)';
end

function polyBW = applyPolygons(J,polyBW)

% Define each manually generated ROI
hPoints(1).data = [-1.312016 -0.120245;
    -1.094739 -0.108038;
    -0.918318 -0.236212;
    -0.873749 -0.001227;
    -0.788324 0.041498;
    -0.760468 0.111689;
    -0.424339 0.261225;
    -0.060355 0.068964;
    -0.123495 -0.230109;
    -0.948031 -0.428473;
    -1.280446 -0.306403];

% Iteratively apply each ROI
for ii = 1:length(hPoints)
    if size(hPoints(ii).data,1) > 2
        in = inpolygon(J(:,1),J(:,2),hPoints(ii).data(:,1),hPoints(ii).data(:,2));
        in = reshape(in,size(polyBW));
        polyBW = polyBW | in;
    end
end

end
```