* **Fusion of Infrared and Visible Image**

In this paper we propose a method for fusion of Infrared and Visible Image by preserving the thermal targets from infrared image and background structure from visible image.

Here is the step used in fusion of infrared and visible image:

1. **Displaying Infrared and Visible Image.**

* **Code**

IR = imread("manWalkIR.jpg");

VIS = imread("manWalkVB.jpg");

* **Output**



1. **Generating Histogram of Infrared image.**

Converting Infrared image to gray scale for thresholding and then generating its histogram.

* **Code**

% Convert IR to grayscale only for thresholding (safe for both RGB or gray)

grayIR = im2gray(IR); % replaces rgb2gray safely

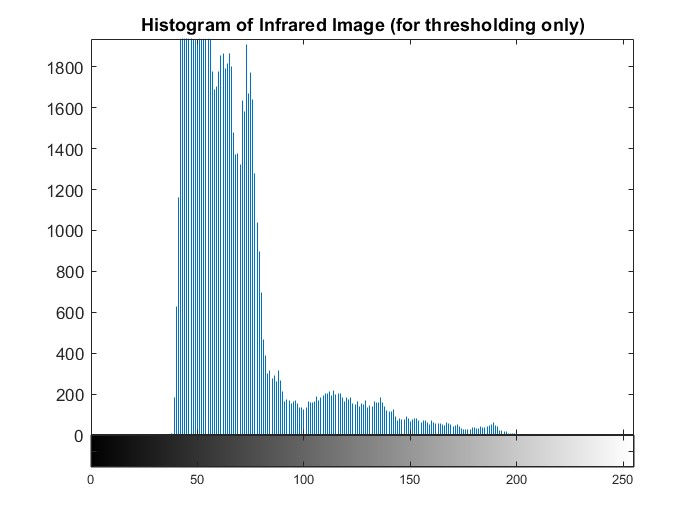
% Display histogram

figure(2)

imhist(grayIR);

title('Histogram of Infrared Image (for thresholding only)');

* **Output**



1. **Gaussian Filter and Otsu Threshold**

Here we apply the gaussian filter to the infrared image to reduce the noise in infrared image. Then we computed the Otus Threshold which helped in creating a binary mask. This binary mask is applied to the infrared image so that the target object in the infrared image is separated from the original infrared image.

* **Code**

% Apply Gaussian smoothing to reduce noise (on grayIR)

smoothedIR = imgaussfilt(grayIR, 2); % sigma = 2

% Compute Otsu threshold

level = graythresh(smoothedIR); % returns normalized threshold [0,1]

threshold = round(level \* 255); % scale to [0,255]

fprintf('Computed Otsu Threshold: %d\n', threshold);

% Create binary mask using threshold

binaryMask = smoothedIR > threshold;

% Morphological closing to fill gaps

binaryMask = imclose(binaryMask, strel('disk', 5)); % fill small holes

% Remove small fragments

binaryMask = bwareaopen(binaryMask, 100); % remove objects < 100 pixels

% Apply mask to IR image

if size(IR,3) == 3

maskedIR = IR;

maskedIR(repmat(~binaryMask, [1 1 3])) = 0;

else

maskedIR = IR;

maskedIR(~binaryMask) = 0;

end

% Display masked IR image

figure(3)

imshow(maskedIR);

title('Masked IR Image (Auto ROI)');

* **Output**

