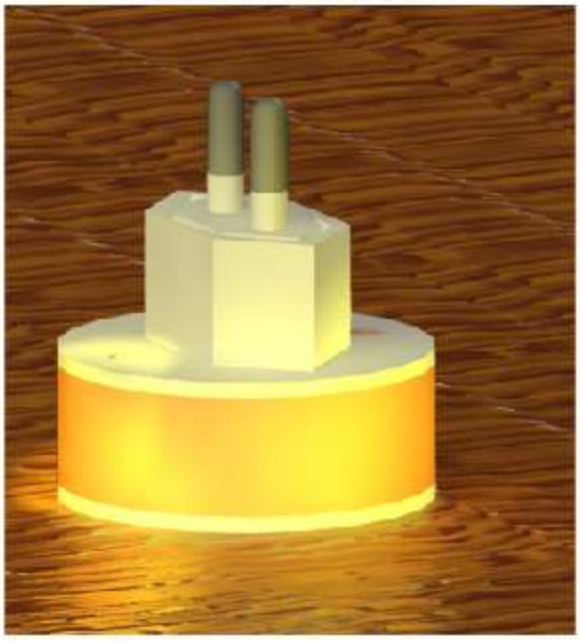
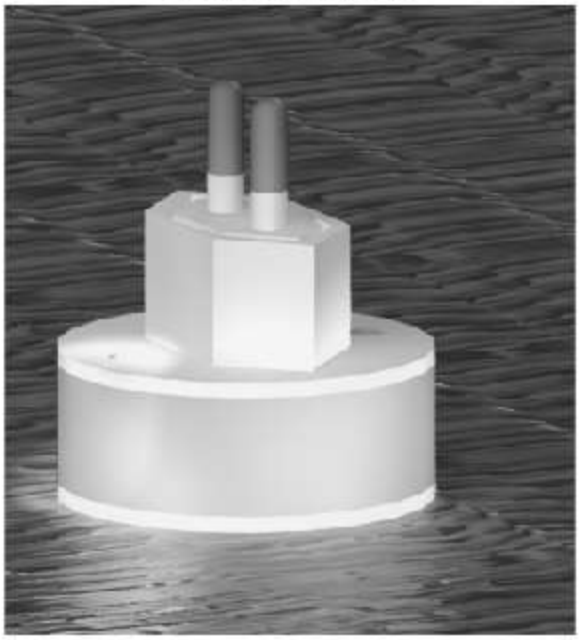
```
% mat=randi([0 255],8,8);
% I= imread("bulb_image.png");
% imshow(I);
% figure;
% Ig=rgb2gray(I);
% imshow(Ig);
% Ir=I;
% Ir(:,:,2)=0;
% Ir(:,:,3)=0;
% figure;
% imshow(Ir);
% Ib_w=Ig>70;
% Ib_w2=uint8(Ib_w);
% figure;
% imshow(Ib_w);
% Ieq=histeq(Ig);
% figure;
% imshow(Ieq);

% Generate an 8x8 matrix with random intensity values between 0 and 255
% This simulates a small grayscale image
mat = randi([0 255], 8, 8);

% Read the input color image from file
I = imread("bulb_image.png");

% Display the original RGB image
imshow(I);
title('Original RGB Image');

figure;

% Convert the RGB image to grayscale
Ig = rgb2gray(I);

% Display the grayscale image
imshow(Ig);
title('Grayscale Image');

% Create a copy of the original image for color channel manipulation
Ir = I;

% Remove green channel by setting it to zero
Ir(:,:,2) = 0;

% Remove blue channel by setting it to zero
Ir(:,:,3) = 0;

% Display the red channel extracted image
figure;
imshow(Ir);
title('Red Channel Image');

% Convert grayscale image into a binary image using threshold value 70
Ib_w = Ig > 70;

% Convert logical image to uint8 format for display
Ib_w2 = uint8(Ib_w);
```

```
% Display the binary image
figure;
imshow(Ib_w);
title('Binary Image using Thresholding');

% Perform histogram equalization on grayscale image
Ieq = histeq(Ig);

% Display the histogram equalized image
figure;
imshow(Ieq);
title('Histogram Equalized Image');
```

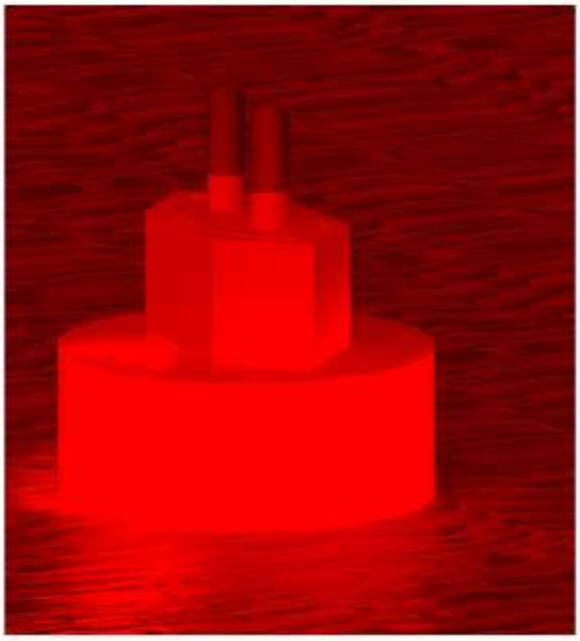
Original RGB Image



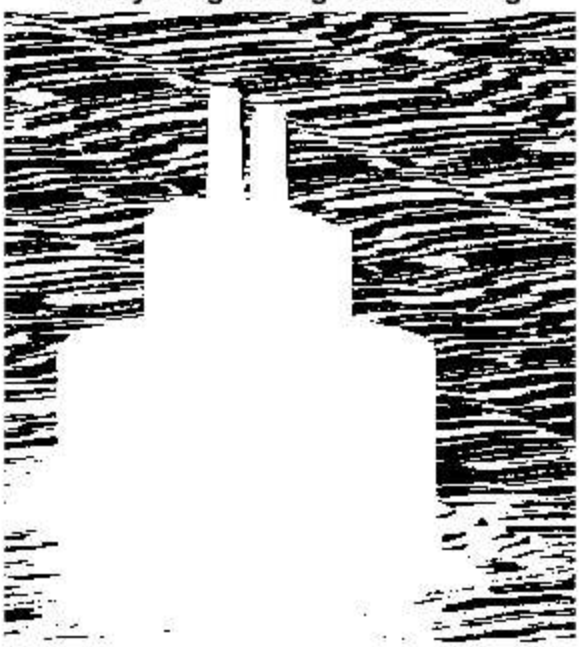
Grayscale Image

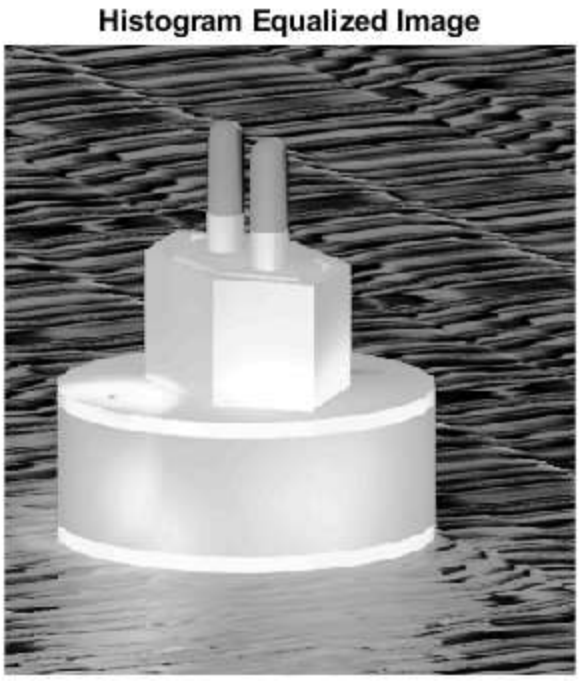


Red Channel Image



Binary Image using Thresholding





Published with MATLAB® R2021a