
```
% have the user freehand draw an irregular shape over
% a gray scale image, have it extract only that part to a new image,
% and to calculate the mean intensity value of the image within that
% shape.
% Also calculates the perimeter, centroid, and center of mass
% (weighted centroid).
% Change the current folder to the folder of this m-file.
clc; % Clear command window.
clear; % Delete all variables.
close all; % Close all figure windows except those created by imtool.
imtool close all; % Close all figure windows created by imtool.
workspace; % Make sure the workspace panel is showing.
fontSize = 16;

fullFileName = 'AxialTest.tif';

grayImage = imread(fullFileName);
imshow(grayImage, []);
axis on;
title('Original Grayscale Image', 'FontSize', fontSize);
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.
message = sprintf('Left click and hold to begin drawing.\nSimply lift
the mouse button to finish');
uiwait(msgbox(message));
hFH = imfreehand();
% Create a binary image ("mask") from the ROI object.
binaryImage = hFH.createMask();
xy = hFH.getPosition;
% Now make it smaller so we can show more images.
subplot(2, 3, 1);
imshow(grayImage, []);
axis on;
title('Original Grayscale Image', 'FontSize', fontSize);
% Display the freehand mask.
subplot(2, 3, 2);
imshow(binaryImage);
axis on;
title('Binary mask of the region', 'FontSize', fontSize);
% Label the binary image and computer the centroid and center of mass.
labeledImage = bwlabel(binaryImage);
measurements = regionprops(binaryImage, grayImage, ...
    'area', 'Centroid', 'WeightedCentroid', 'Perimeter');
area = measurements.Area
centroid = measurements.Centroid
centerOfMass = measurements.WeightedCentroid
perimeter = measurements.Perimeter
% Calculate the area, in pixels, that they drew.
numberOfPixels1 = sum(binaryImage(:))
% Another way to calculate it that takes fractional pixels into
% account.
numberOfPixels2 = bwarea(binaryImage)
% Get coordinates of the boundary of the freehand drawn region.
```

```

structBoundaries = bwboundaries(binaryImage);
xy=structBoundaries{1}; % Get n by 2 array of x,y coordinates.
x = xy(:, 2); % Columns.
y = xy(:, 1); % Rows.
subplot(2, 3, 1); % Plot over original image.
hold on; % Don't blow away the image.
plot(x, y, 'LineWidth', 2);
drawnow; % Force it to draw immediately.
% Burn line into image by setting it to 255 wherever the mask is true.
burnedImage = grayImage;
burnedImage(binaryImage) = 255;
% Display the image with the mask "burned in."
subplot(2, 3, 3);
imshow(burnedImage);
axis on;
caption = sprintf('New image with\nmask burned into image');
title(caption, 'FontSize', fontSize);
% Mask the image and display it.
% Will keep only the part of the image that's inside the mask, zero
% outside mask.
blackMaskedImage = grayImage;
blackMaskedImage(~binaryImage) = 0;
subplot(2, 3, 4);
imshow(blackMaskedImage);
axis on;
title('Masked Outside Region', 'FontSize', fontSize);
% Calculate the mean
meanGL = mean(blackMaskedImage(binaryImage));
% Put up crosses at the centroid and center of mass
hold on;
plot(centroid(1), centroid(2), 'r+', 'MarkerSize', 30, 'LineWidth',
    2);
plot(centerOfMass(1), centerOfMass(2), 'g+', 'MarkerSize',
    20, 'LineWidth', 2);
% Now do the same but blacken inside the region.
insideMasked = grayImage;
insideMasked(binaryImage) = 0;
subplot(2, 3, 5);
imshow(insideMasked);
axis on;
title('Masked Inside Region', 'FontSize', fontSize);
% Now crop the image.
leftColumn = min(x);
rightColumn = max(x);
topLine = min(y);
bottomLine = max(y);
width = rightColumn - leftColumn + 1;
height = bottomLine - topLine + 1;
croppedImage = imcrop(blackMaskedImage, [leftColumn, topLine, width,
    height]);
% Display cropped image.
subplot(2, 3, 6);
imshow(croppedImage);
axis on;

```

```
title('Cropped Image', 'FontSize', fontSize);
% Put up crosses at the centriod and center of mass
hold on;
plot(centroid(1)-leftColumn, centroid(2)-topLine, 'r+', 'MarkerSize',
    30, 'LineWidth', 2);
plot(centerOfMass(1)-leftColumn, centerOfMass(2)-topLine, 'g
+', 'MarkerSize', 20, 'LineWidth', 2);
% Report results.
message = sprintf('Mean value within drawn area = %.3f\nNumber of
    pixels = %d\nArea in pixels = %.2f\nperimeter = %.2f\nCentroid at
    (x,y) = (%.1f, %.1f)\nCenter of Mass at (x,y) = (%.1f, %.1f)\nRed
    crosshairs at centroid.\nGreen crosshairs at center of mass.', ...
    meanGL, numberOfPixels1, numberOfPixels2, perimeter, ...
    centroid(1), centroid(2), centerOfMass(1), centerOfMass(2));
msgbox(message);
```

Published with MATLAB® R2016b