Less known Functions

# bwulterode

> I'm in aeed of a matlab code to connect all white dots in a black image ie., connect two dots by drawing a line between them. can anyone help me?  
>  
> thanks,  
> siva  
  
---------------------------------------------------  
You can use bwulterode() to get the centroid, followed by imline.  
  
Here's the demo for bwulterode.  
  
originalBW = imread('circles.png');  
imshow(originalBW)  
ultimateErosion = bwulterode(originalBW);  
figure, imshow(ultimateErosion)  
  
  
  
Here's the demo for imline:  
  
%----- Burn line into image -----  
burnedImage = imread('pout.tif');  
% Create line mask, h, as an ROI object over the second image in the  
bottom row.  
subplot(2, 4, 6);  
hLine = imline(gca,[10 100],[10 100]); % Second argument defines line  
endpoints.  
% Create a binary image ("mask") from the ROI object.  
binaryImage2 = hLine.createMask();  
% Display the line mask.  
subplot(2, 4, 7);  
imshow(binaryImage2);  
title('Binary mask of the line');  
% Burn line into image by setting it to 255 wherever the mask is true.  
burnedImage(binaryImage2) = 255;  
% Display the image with the "burned in" line.  
subplot(2, 4, 8);  
imshow(burnedImage);  
title('New image with line burned into image');  
  
  
Combine those two to get what you want.  
Let me know if you can't figure it out.

# imclearborder

# [Steve on Image Processing](http://blogs.mathworks.com/steve)

## September 4th, 2007

## [Clearing border components](http://blogs.mathworks.com/steve/2007/09/04/clearing-border-components/" \o "Permanent Link to Clearing border components)

I saw an application recently where someone wanted to handle NaNs in an image differently depending on whether they were "interior." In other words, a set of connected NaN pixels surrounded by non-NaN pixels are handled one way, and a set of connected NaN pixels that touches the image border are handled another way.

Coincidentally, another person recently commented on the need in some measurement situations to remove objects that touch the image border.

So I thought it would be good time to show how to use the Image Processing Toolbox functionimclearborder. This function removes all connected components of a binary image that touch any image border. Here's an example. (To run the example, download aug31.png from [here](http://blogs.mathworks.com/images/steve/168/aug31.png).

BW = imread('aug31.png');

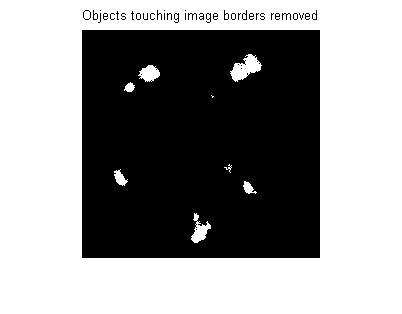
imshow(BW)



BW2 = imclearborder(BW);

imshow(BW2)

title('Objects touching image borders removed')



Now back to the NaN problem. Here's a small example:

f = magic(5);

f(2:5,1:2) = NaN;

f(3:4,4) = NaN

*f =*

*17 24 1 8 15*

*NaN NaN 7 14 16*

*NaN NaN 13 NaN 22*

*NaN NaN 19 NaN 3*

*NaN NaN 25 2 9*

The NaNs on the left make up a group touching the border. The NaNs in the fourth column are an interior group. We can distinguish between the two groups by using imclearboder and some logical operations.

nan\_mask = isnan(f)

*nan\_mask =*

*0 0 0 0 0*

*1 1 0 0 0*

*1 1 0 1 0*

*1 1 0 1 0*

*1 1 0 0 0*

border\_nans = nan\_mask & ~imclearborder(nan\_mask)

*border\_nans =*

*0 0 0 0 0*

*1 1 0 0 0*

*1 1 0 0 0*

*1 1 0 0 0*

*1 1 0 0 0*

interior\_nans = nan\_mask & ~border\_nans

*interior\_nans =*

*0 0 0 0 0*

*0 0 0 0 0*

*0 0 0 1 0*

*0 0 0 1 0*

*0 0 0 0 0*

You can use also imclearborder on gray scale images. As the documentation describes, the function "suppresses structures that are lighter than their surroundings and that are connected to the image border." You can see the effect using the rice.png image:

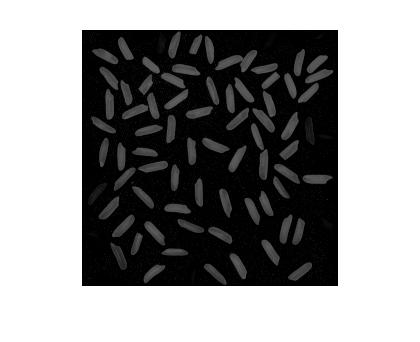
I = imread('rice.png');

imshow(I)



J = imclearborder(I);

imshow(J)



The modified image looks darker. As the documentation notes, for gray scale images imclearborder "tends to reduce the overall intensity level in addition to suppressing border structures." Using the display scaling syntax of imshow helps to see the result clearly:

imshow(J, [])



So what happened to those two dark grains (on the left and on the right)? If we zoom into the original image to look closely at the grain on the right, we can see that there are a couple of slightly darker gray pixels connecting the grain to its neighbor on the right.

imshow(I)

axis([200 256 81 137])



But that grain on the right touches the border. So when imclearborder removes it, it also has the affect of lowering the gray level of its neighbor.

imshow(J, [])

axis([200 256 81 137])



Have you found imclearborder to be useful? Let me know - I'd like to hear about it.