

Project 8: Morphology

Contents

- Course No: ECE 5256
- Due Date: 4/9/2021
- In this project please describe your steps, and submit your code and results for the following. Use the image Dots.gif. Assume all particles are the same size. Write an algorithm that produces three images that consist of:
- Reading Dots.gif image
- For part (a): Only particles that have merged with the boundary of the image
- For part (b): Only overlapping particles
- Labelling each dot in the image
- For part (c): Only non-overlapping particles

Course No: ECE 5256

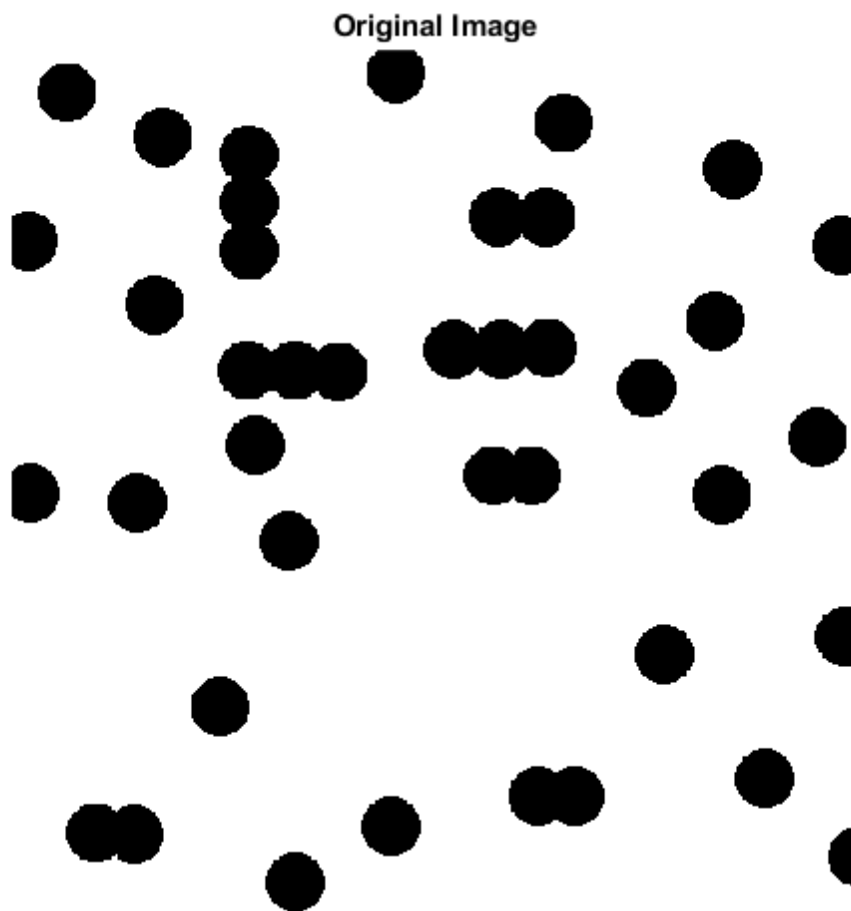
Due Date: 4/9/2021

In this project please describe your steps, and submit your code and results for the following. Use the image Dots.gif. Assume all particles are the same size. Write an algorithm that produces three images that consist of:

```
%(a) Only particles that have merged with the boundary of the image
%(b) Only overlapping particles
%(c) Only non-overlapping particles
% %Note that you should invert the image as white is considered the foreground and black the
% background. The maximum value of the image is 251 and the minimum is 0.
% For part (a), one approach is to make the border pixels the same as the foreground, then
% apply a connected component algorithm.
% For part (b), one approach is to eliminate the dots merged with the border and measure
% the size of a single dot. Then retain or discard dots of that size. Finding the difference
% between this image and remainder will allow you to get the two images in part (b) and
% (c).
```

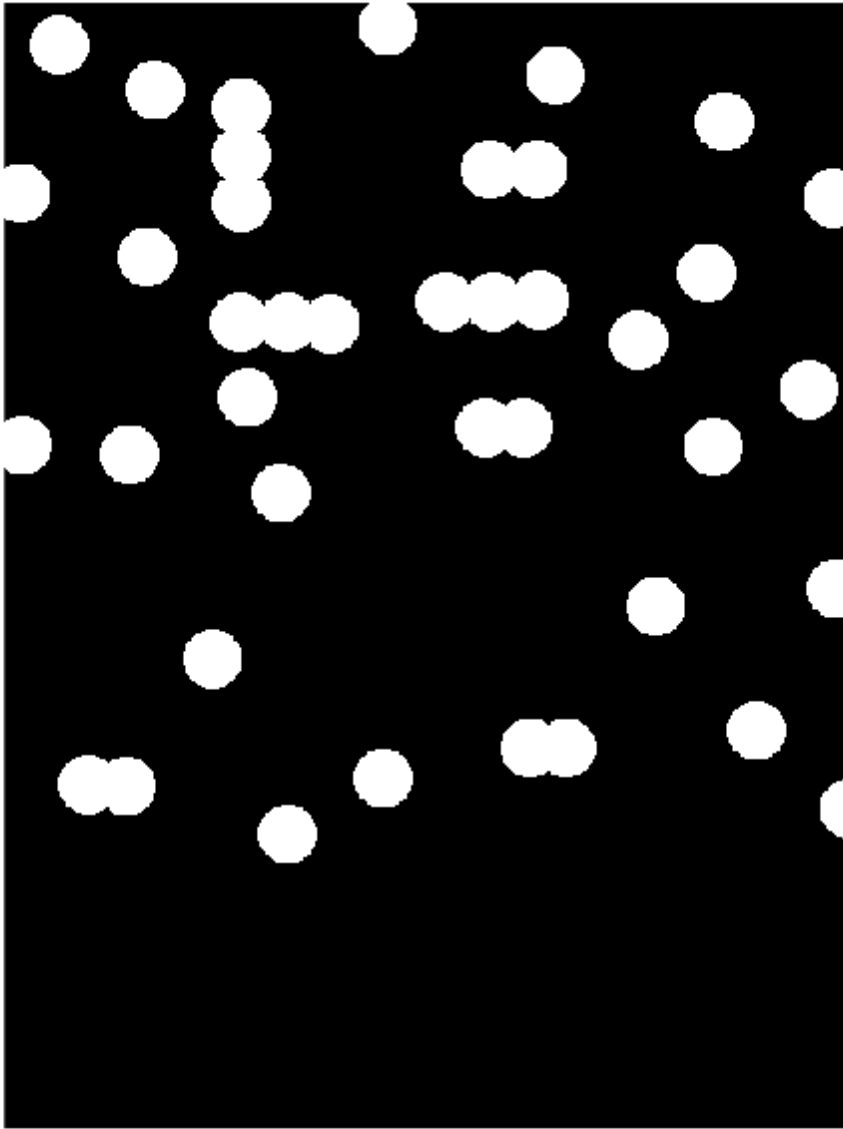
Reading Dots.gif image

```
I=double(imbinarize(imread("Dots.gif")));
imshow(I);
title("Original Image");
```



```
J = imcomplement(I); % Inverting the image
imshow(J);
title("Inverted Original Image");
```

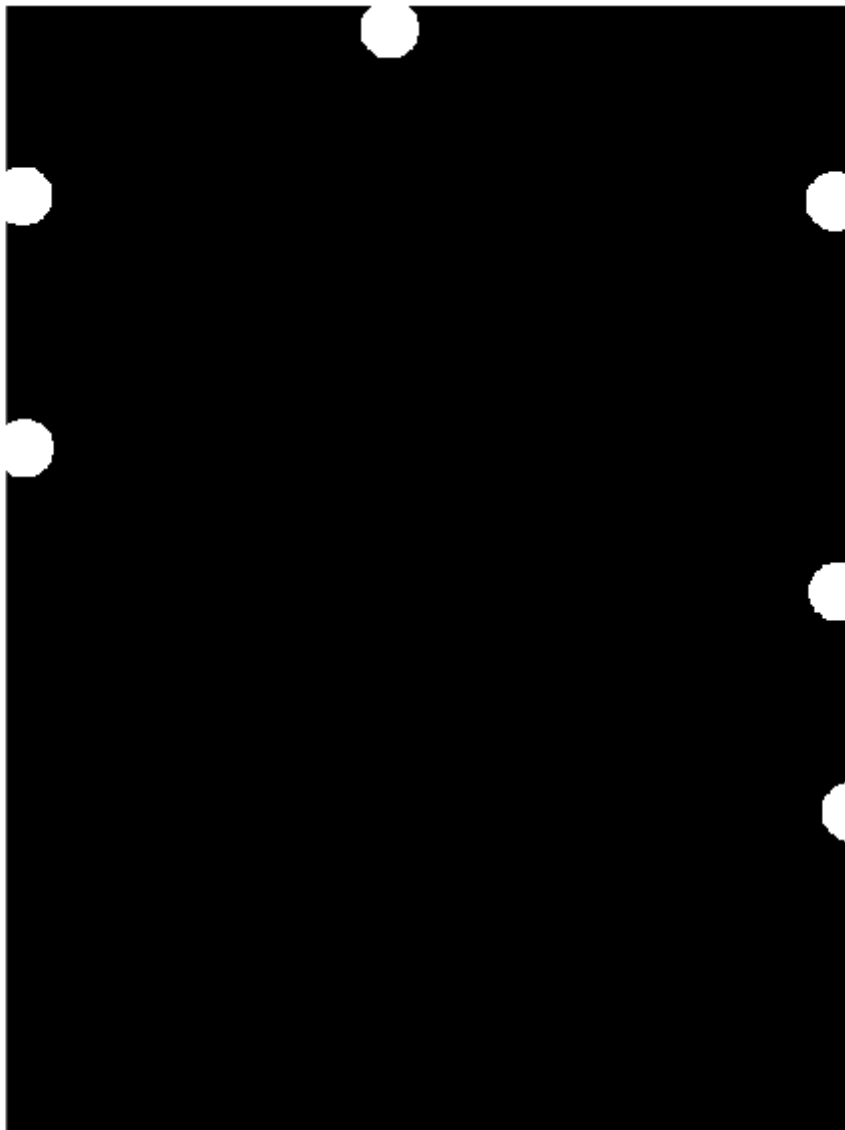
Inverted Original Image



For part (a): Only particles that have merged with the boundary of the image

```
[r c]=size(J);
output=zeros(r,c);
for i=1:r
    for j=1:c
        if(i==r || j==c || i==1 || j==1)
            output(i,j)=J(i,j);
        end
    end
end
output=double(output);
imshow(output);
title("Masked Image");
es=imreconstruct(output,J); % Apply a connected component algorithm.
imshow(es);
title("Only particles that have merged with the boundary of the image");
```

Only particles that have merged with the boundary of the image

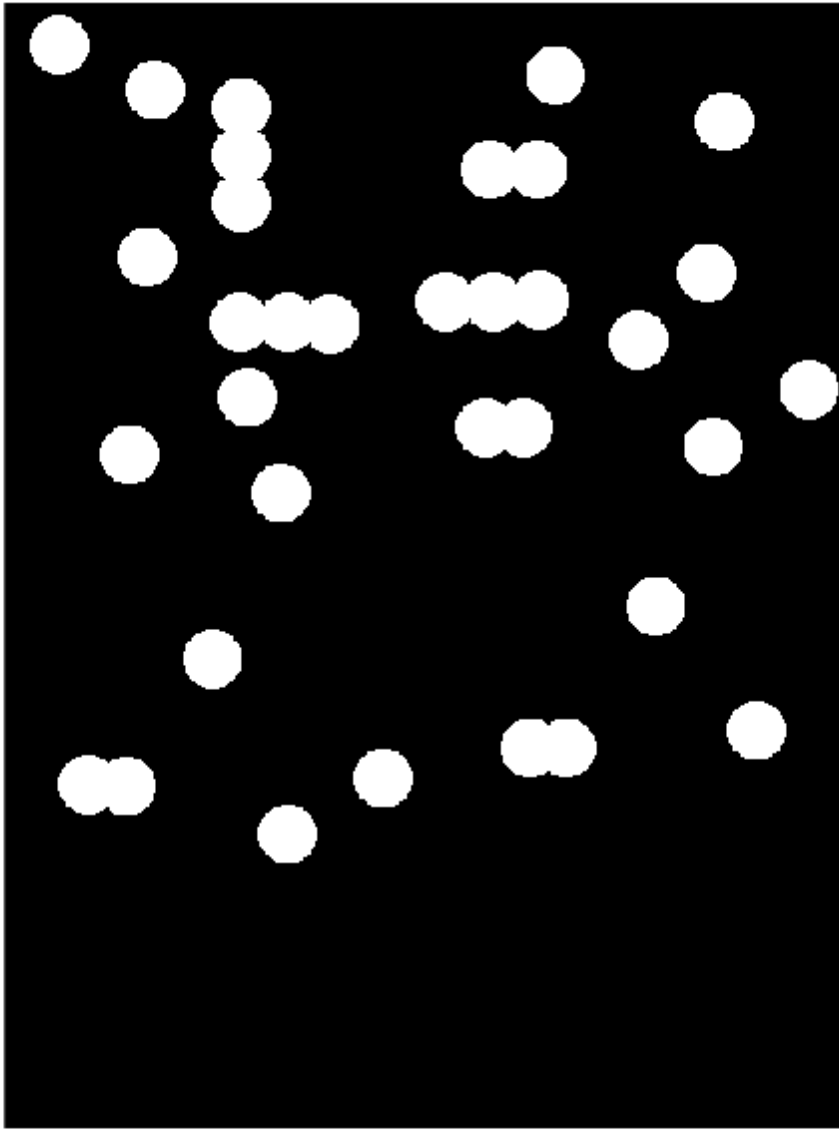


For part (b): Only overlapping particles

Lets clear the circles at the border

```
Z = J-es;  
imshow(Z);  
title('Eliminate the dots merged with the border');
```

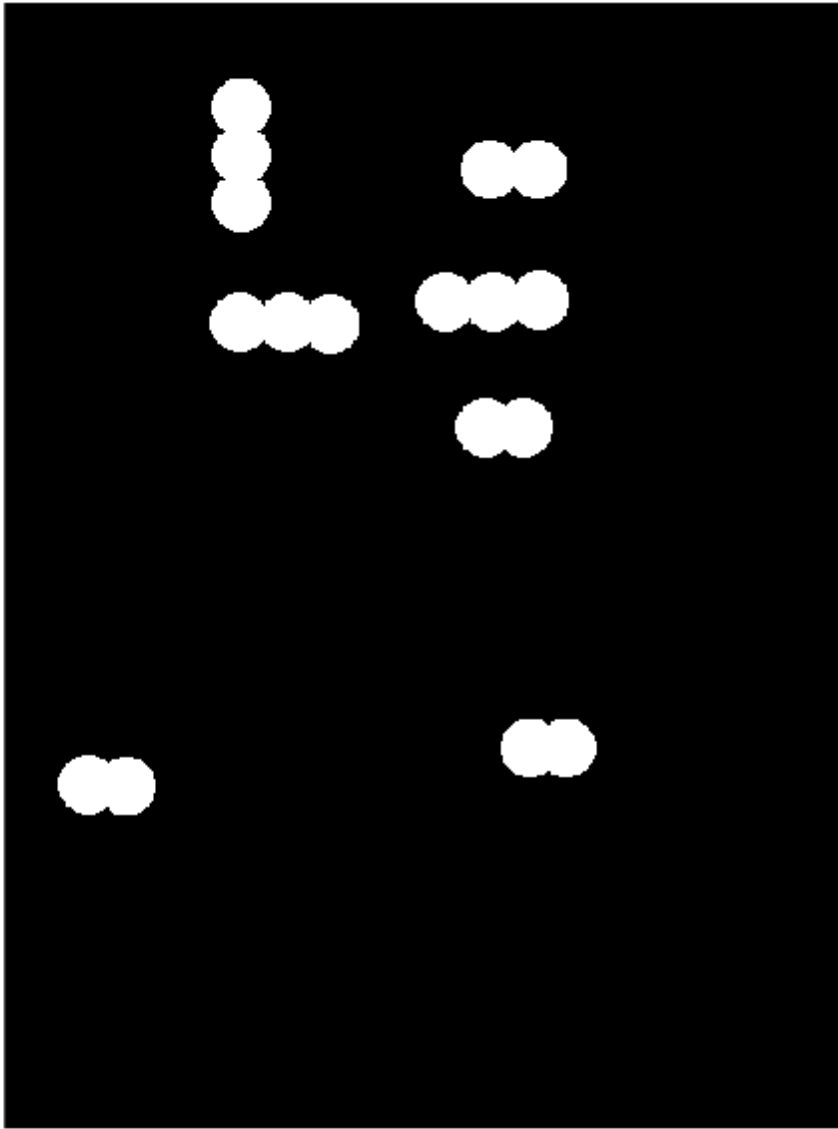
Eliminate the dots merged with the border



Labelling each dot in the image

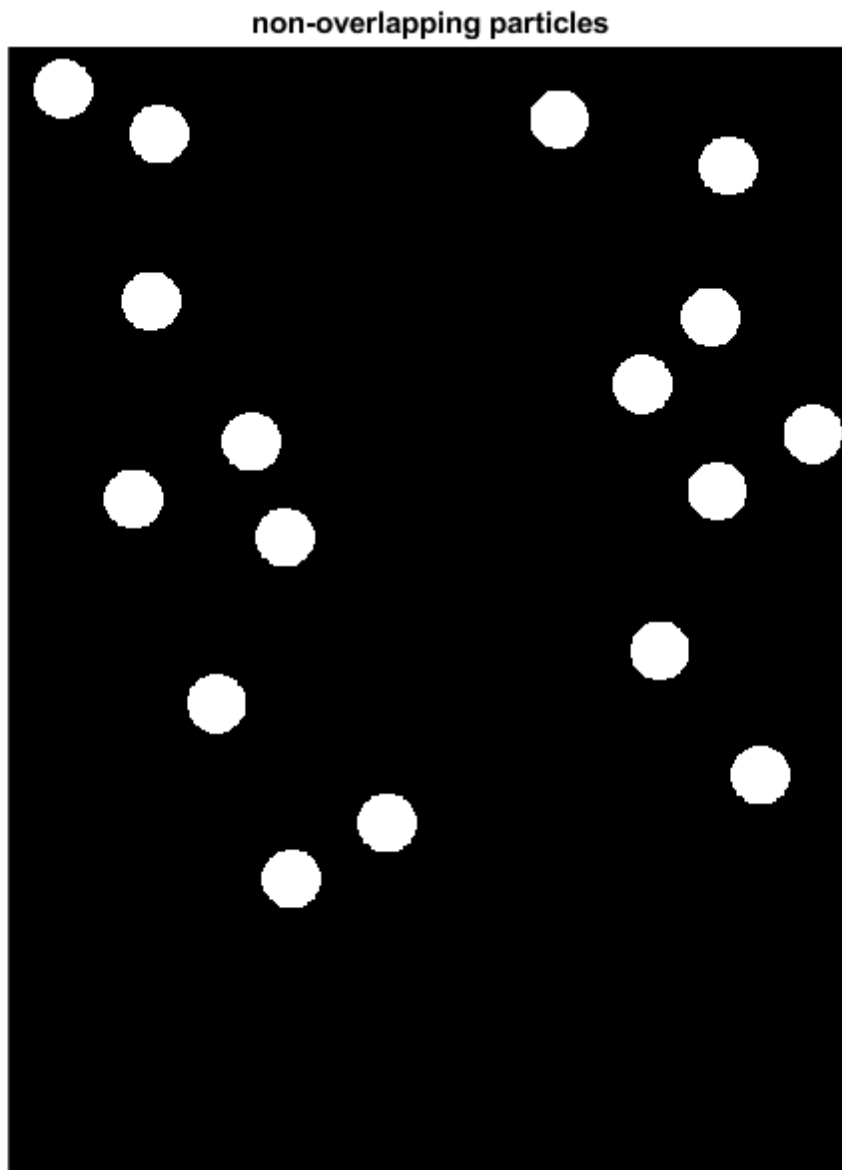
Now I have labelled each dot respectively. Based on these labels, the overlapped and single dots are classified.

```
label=bwlabel(J);
max(max(label));
im1 = (label==4);
im2 = (label==9);
im3 = (label==10);
im4 = (label==16);
im5 = (label==17);
im6 = (label==18);
im7 = (label==19);
out_overlapped = im1+im2+im3+im4+im5+im6+im7;
imshow(out_overlapped);
title("overlapping particles");
```

overlapping particles**For part (c): Only non-overlapping particles**

```
im8 = (label==3);  
im9 = (label==5);  
im10 = (label==6);  
im11 = (label==7);  
im12 = (label==8);  
im13 = (label==11);  
im14 = (label==12);  
im15 = (label==13);  
im16 = (label==14);  
im17 = (label==20);  
im18 = (label==21);  
im19 = (label==22);  
im20 = (label==23);  
im21 = (label==24);  
im22 = (label==25);  
im23 = (label==26);
```

```
im24 = (label==27);  
out_single = im8+im9+im10+im11+im12+im13+im14+im15+im16+im17+im18+im19+im20+im21+im22+im23+im24;  
imshow(out_single);  
title("non-overlapping particles");
```



Published with MATLAB® R2019a