ECE 5256

Project 9: Morphology

Keefe Kamp

April 15, 2023

1 Part 1

This project set three goals for me to complete: (1) Identify the dots that touch the edge, (2) Identity the single dots, and (3) Identify the points that are touching. The preproccesing of this image is to invert the colors so the dots are white (foreground) and the rest is black (background). as well as covert it into a binary image. The way I chose to approach the goals is to first pad the input image with a white boarder. This allowed me to run the connected comments algorithm in MATLAB , `bwconncomp()` from this I pulled the largest component and separated it out. These are the ones that touch the boarder. From this I pull the mode of the connected components because the the size that is the most common would be a dot that is not touching anything becasue the overlap between dots is unlikely to be the same for two different sets of overlapping dots or the amount that one is cut off by the border will not be the same. This results in the most common size as the full unconnected dot. After pulling the boarder dots and the single dots I could subtract them from the main image to get the connected dots.

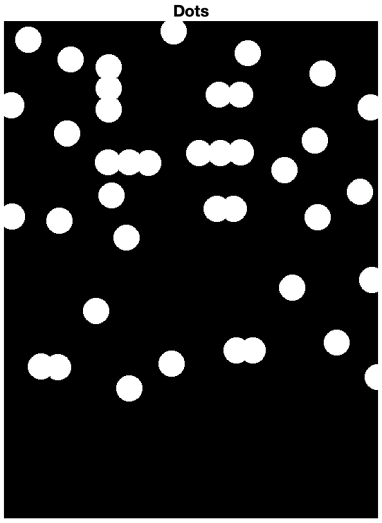


Figure 1: Input image after pre-processing

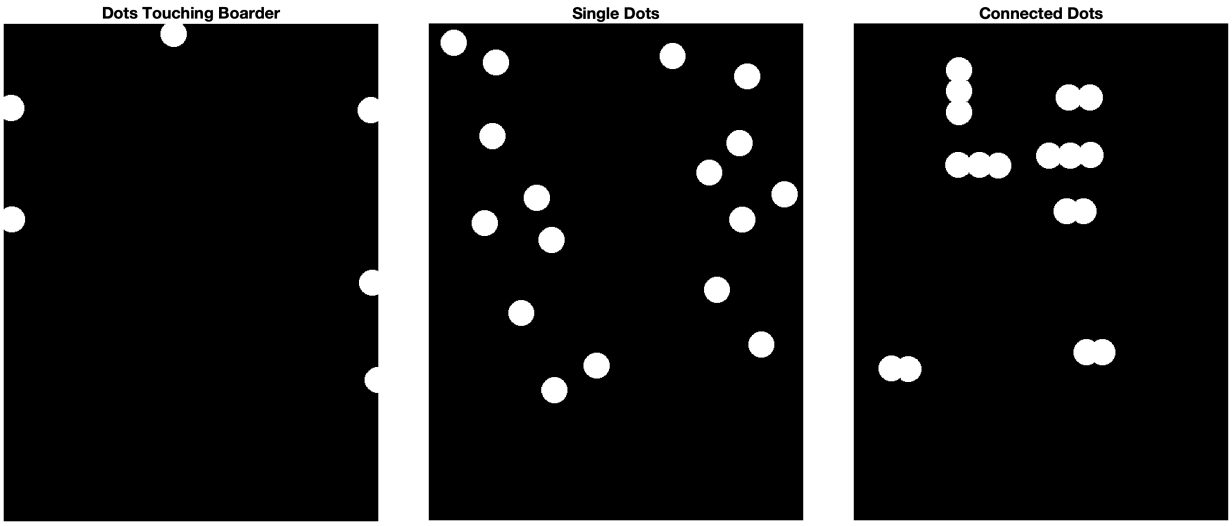
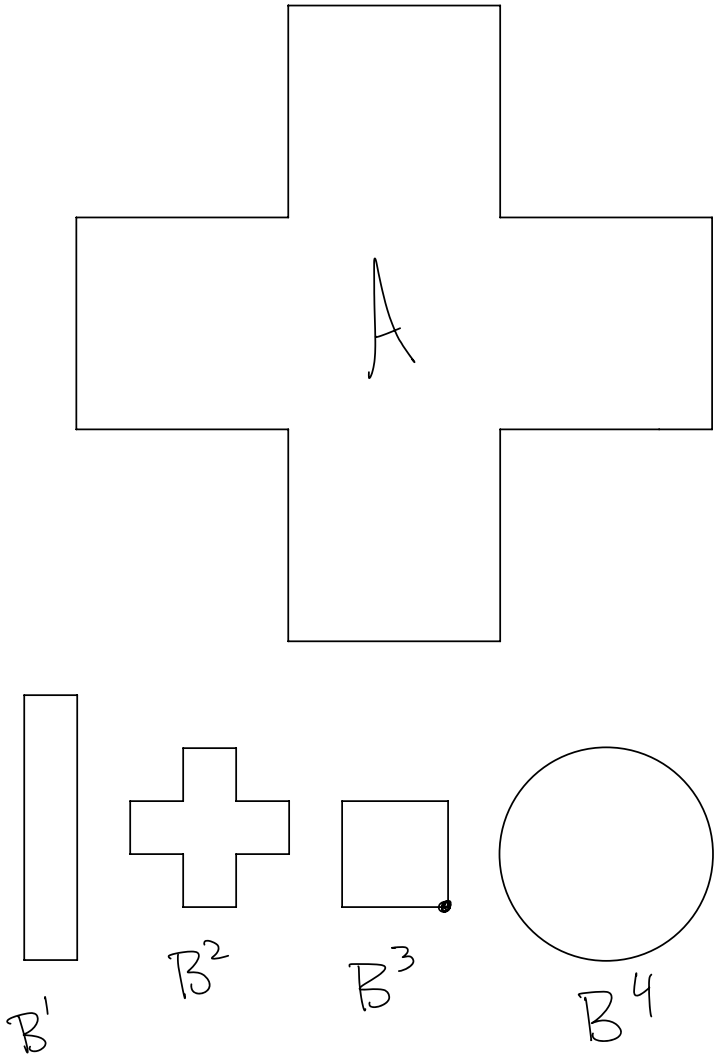


Figure 2: Separated images from left to right [Edge dots, Single dots, connected dots]

2 Book Problems

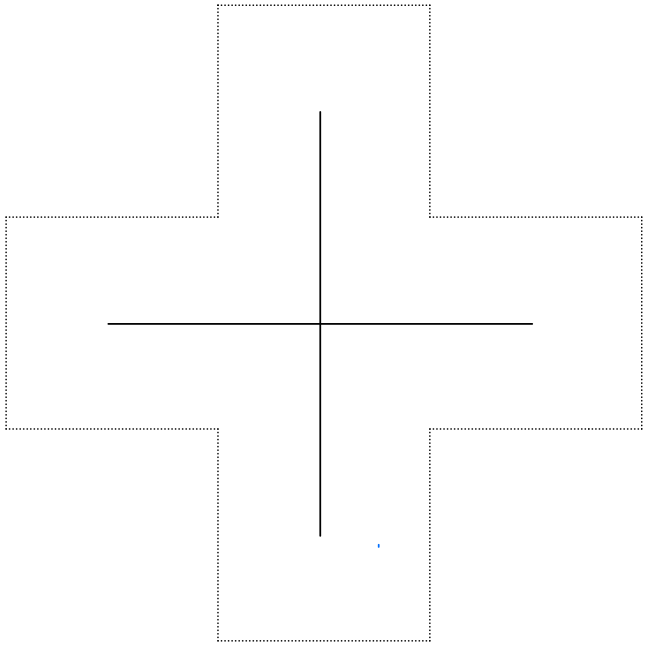
Book problems done on table are on the following pages.

9.9

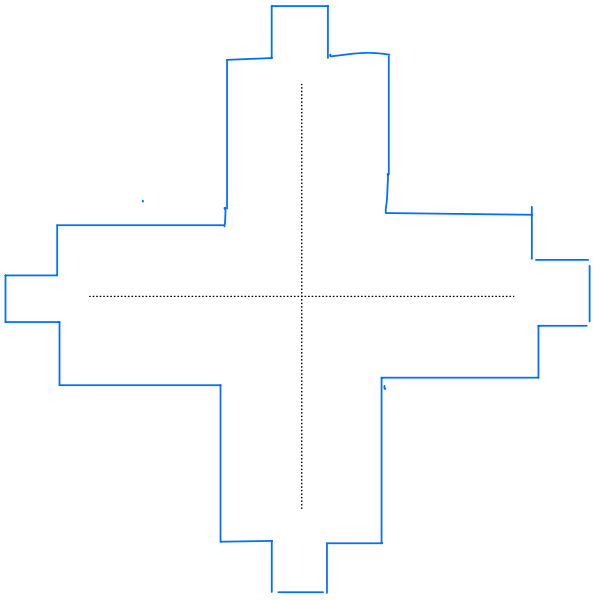


I did this on a grid but removed the grid to be able to see better

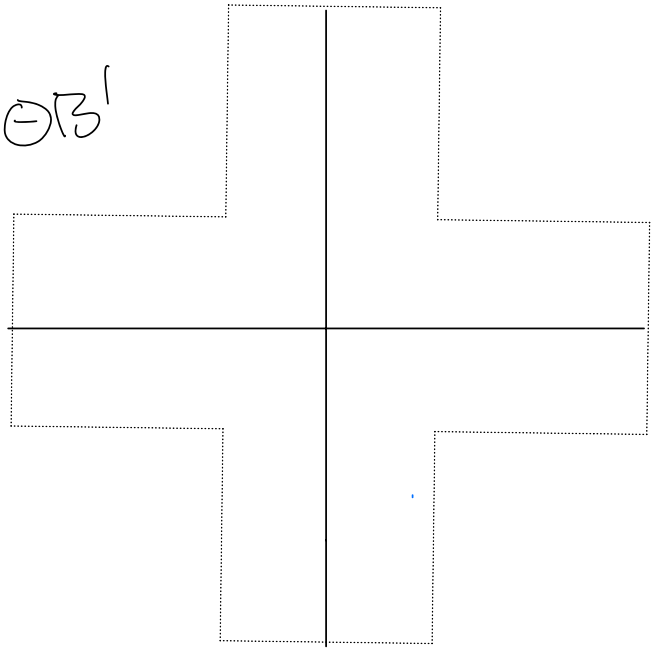
a) $A \ominus B^4 =$



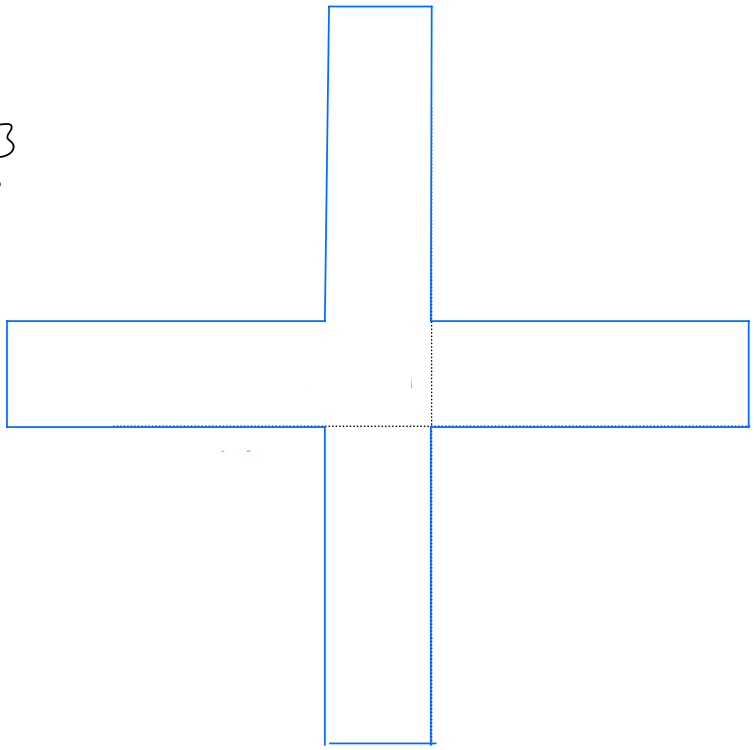
b) $\ominus B^2$



b) $A \ominus B^1$

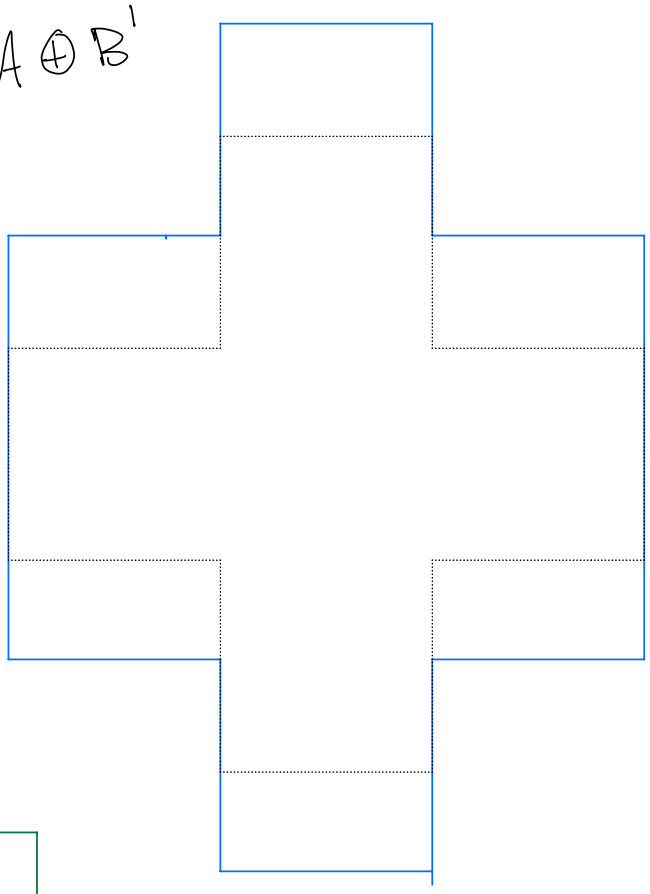


④ B^3



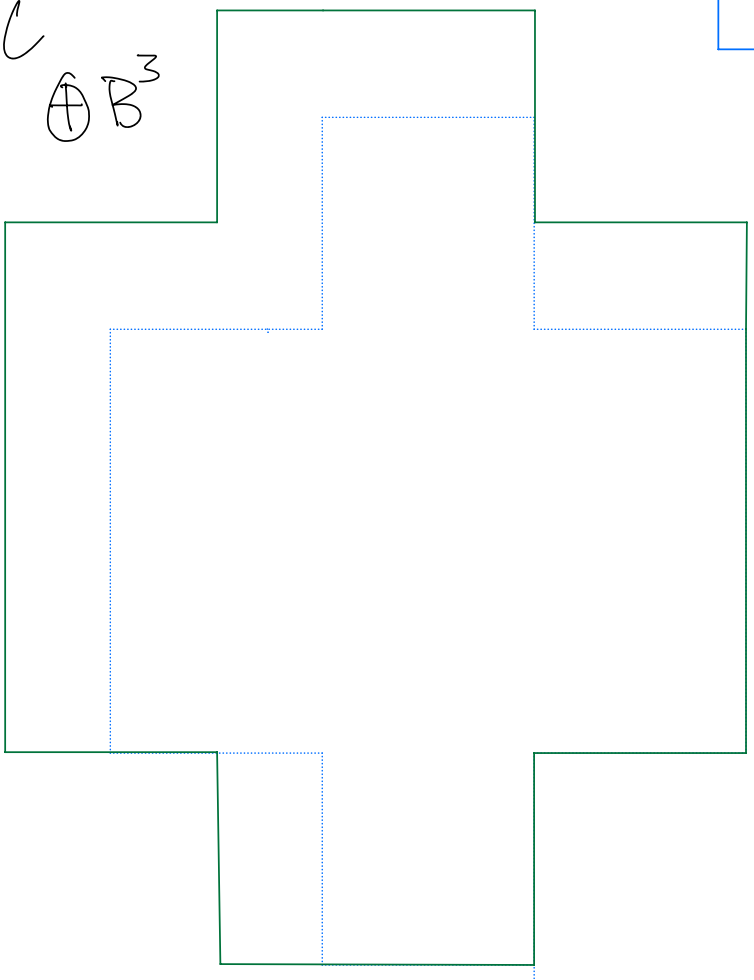
C

$A \oplus B^1$



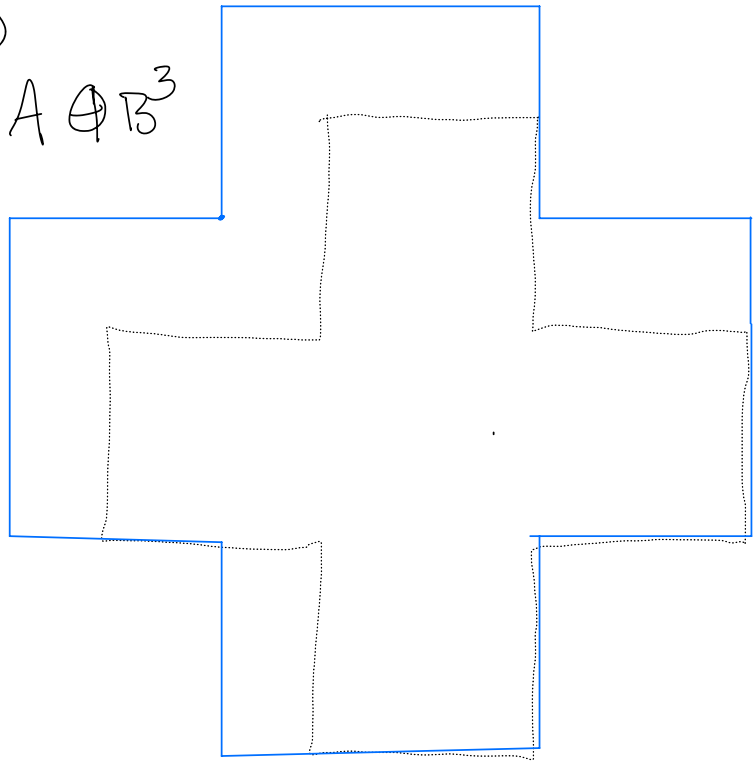
C

$\oplus B^3$

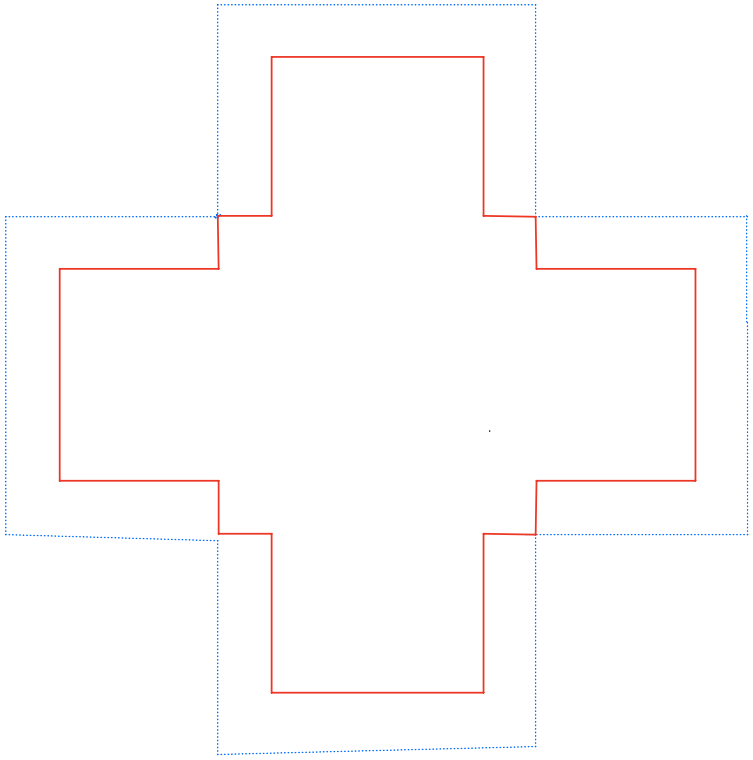


D

$A \oplus B^3$



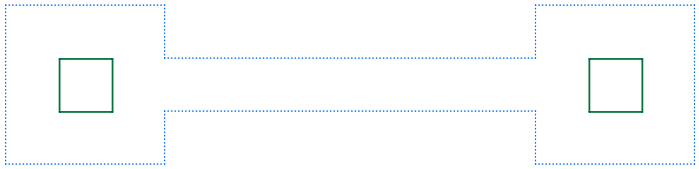
$\ominus B^2$



9.18

opening $(A \ominus B) \oplus B$

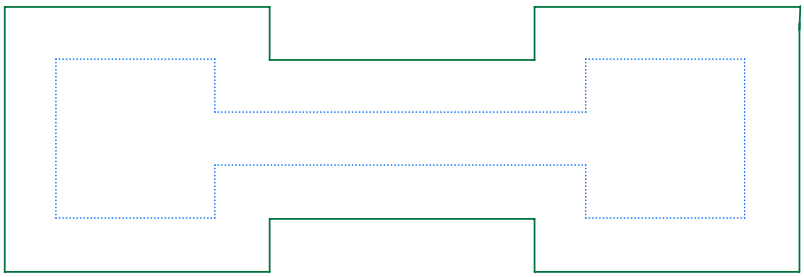
$A \circ B$



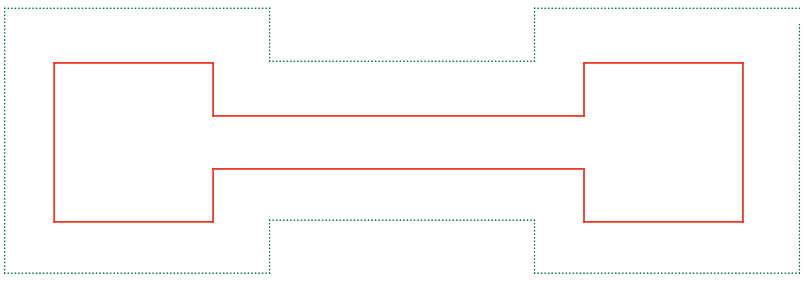
$\oplus B$

Closing $(A \oplus B) \ominus B$

$A \oplus B$



$\ominus B$



A MATLAB Code

This is the code used for the coding portions of this project.

```
%Housekeeping commands
clear all
close all

Dots=imread('Dots.gif'); %reading in image
Dots=im2gray(Dots); %convering to grayscale
%making image binary
Dots=Dots./max(Dots);
Dots=cast(Dots==0,'uint8');

DotsB=padarray(Dots,[1 1], 1); %padding array to have white boarder

CC=bwconncomp(DotsB); %proforming connected component algorithm
%finding size ofg each connected component
numPixels = cellfun(@numel,CC.PixelIdxList);
%finding the size and index of the largest connected component
[biggest,idx] = max(numPixels);
Boarder=zeros(size(DotsB)); %alloacting memory for new image
Boarder(CC.PixelIdxList{1,idx})=1; %"printing" boarder dots on new image



%displaying and saving boarder image
figure
imagesc(Boarder)
colormap('gray')
axis off image
title('Dots Touching Boarder')
exportgraphics(gcf,'Boarder.png','Resolution',300)

%finding most common size of connected compoent
M=mode(numPixels);

%finding which compentents are that size
```

```
idxs=find(numPixels==M);

single=zeros(size(DotsB)); %allocating memory for image picture
for i=1:length(idxs) %itterating though single dots and "printing them
    single(CC.PixelIdxList{1,idxs(i)})=1;
end

%displaying and saving single dots image
figure
imagesc(single)
colormap('gray')
axis off image
title('Single Dots')
exportgraphics(gcf,'Single.png','Resolution',300)

%subtracting two other groupds to ger connected dots
connected=DotsB-uint8(Boarder+single);

%Dispalying and saving connected dots
figure
imagesc(connected)
colormap('gray')
axis off image
title('Connected Dots')
exportgraphics(gcf,'Connected.png','Resolution',300)
```