Keith Mikoleit

UW# 1468441

Abstract

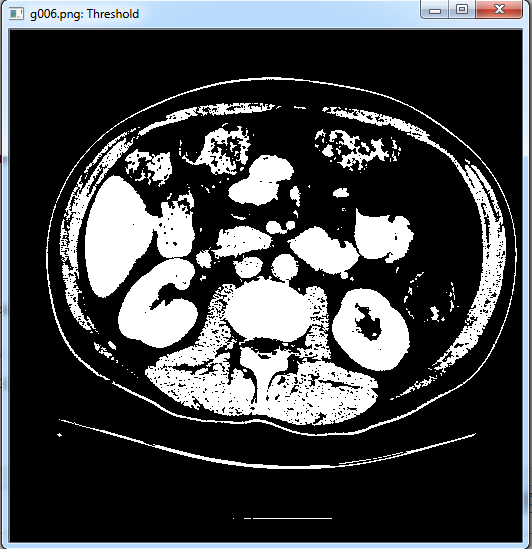
In this assignment I used the OpenCV package for Python to find the connected components of several MRI generated images of the liver and kidneys. All functions referenced are from the OpenCV and Numpy Packages.

EE596 Homework 1

# 1st Image: g006.png

For the first image I created a binary from the original my thresholding with a threshold of 128 and a max value of 255. I then opened the image with an ellipse of size (17,17), and then closed it with an ellipse of size (1,1). Results shown below:

## Original and Threshold

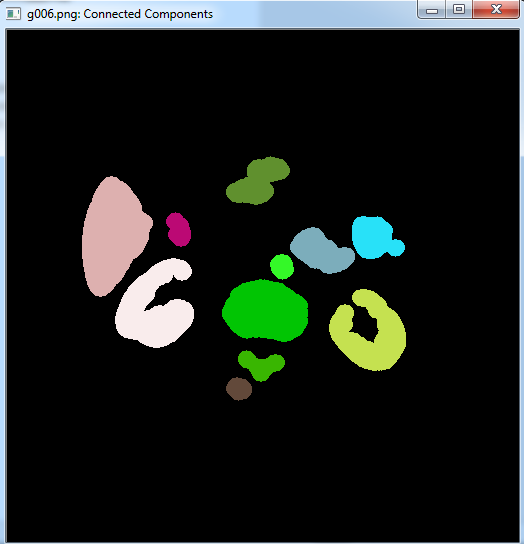


## Opening and Closing



## Connected Components

Finally, I used findCountours to get the extreme outer contours and save all points for drawing later. I then created an empty array of zeros of the same size of the original image using the numpy function zeros and then filled this array using drawContours. The function randint was used to generate random colors for the different connected components.



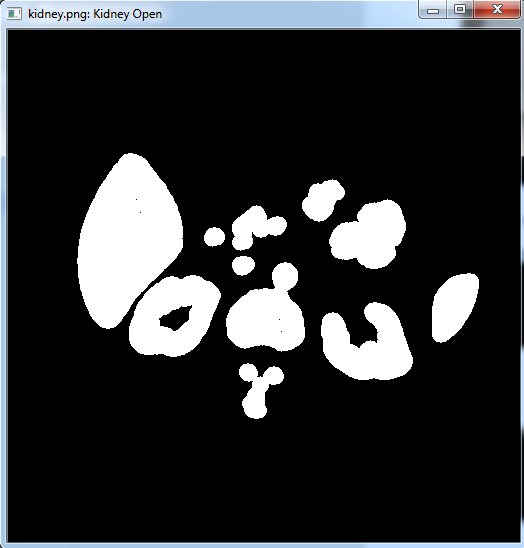
# 2nd Image: kidney.png

For the second image I created a binary from the original my thresholding with a threshold of 128 and a max value of 255. I then opened the image with an ellipse of size (17,17), and then closed it with an ellipse of size (2,2). Results shown below:

## Original and Threshold

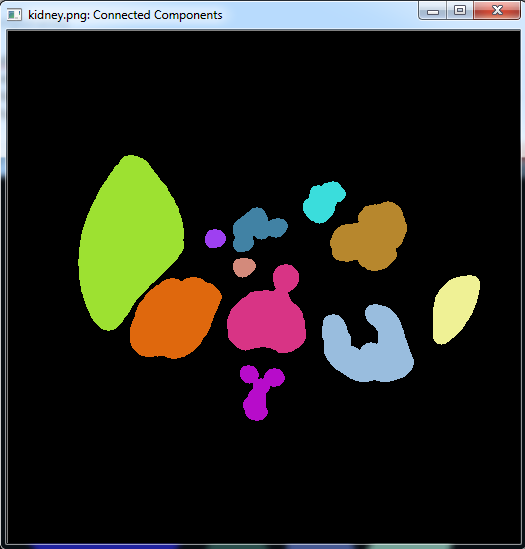


## Opening and Closing



## Connected Components

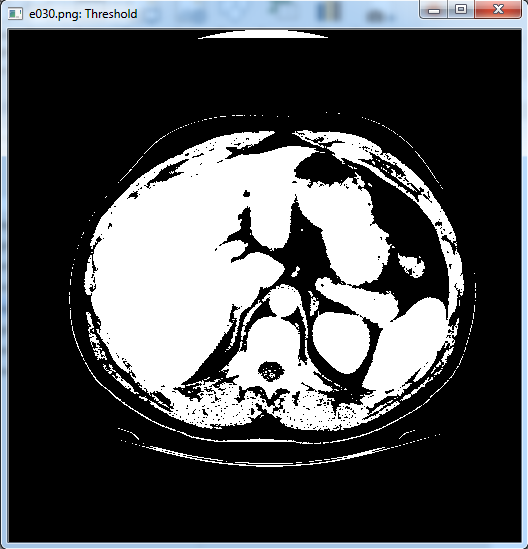
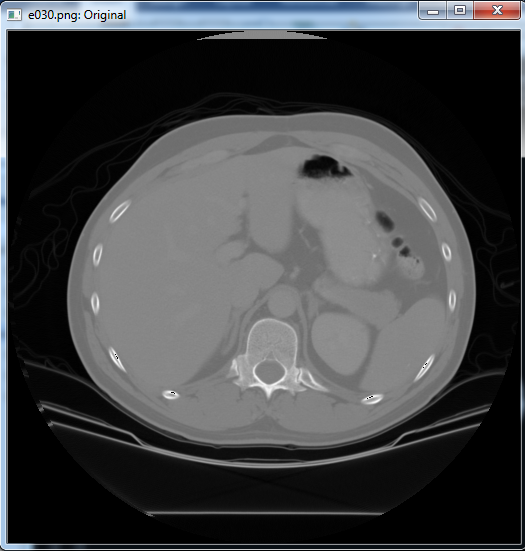
Finally, I used findCountours to get the extreme outer contours and save all points for drawing later. I then created an empty array of zeros of the same size of the original image using the numpy function zeros and then filled this array using drawContours. The function randint was used to generate random colors for the different connected components.



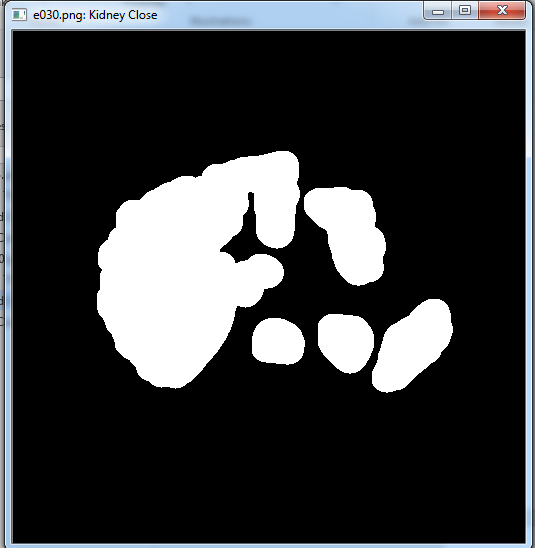
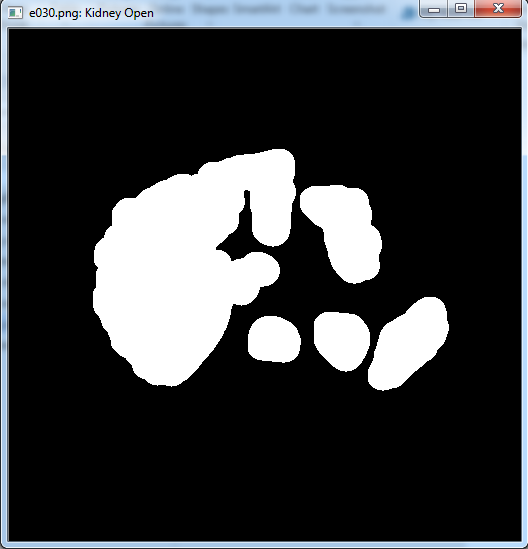
# 3rd Image: e030.png

For the second image I created a binary from the original my thresholding with a threshold of 128 and a max value of 255. I then opened the image with an ellipse of size (30,30), and then closed it with an ellipse of size (1,1). Results shown below:

## Original and Threshold



## Opening and Closing



## Connected Components

Finally, I used findCountours to get the extreme outer contours and save all points for drawing later. I then created an empty array of zeros of the same size of the original image using the numpy function zeros and then filled this array using drawContours. The function randint was used to generate random colors for the different connected components.

