Al Albi

Sean Hong

CSE 4095

April 24, 2018

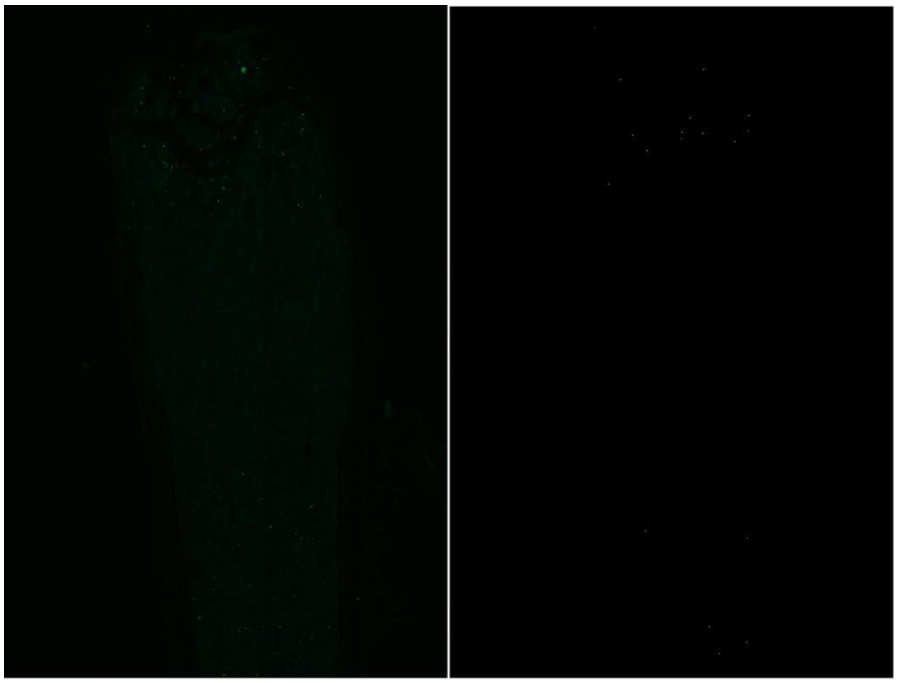
Project 2

1. Registration
   b. In my program, I implemented find_circles.m to implement the circle-finding code. Using a threshold of 0.6 and a margin of 0.2, I identified the beads. Below are the beads for the two pairs of images.

      First Pair

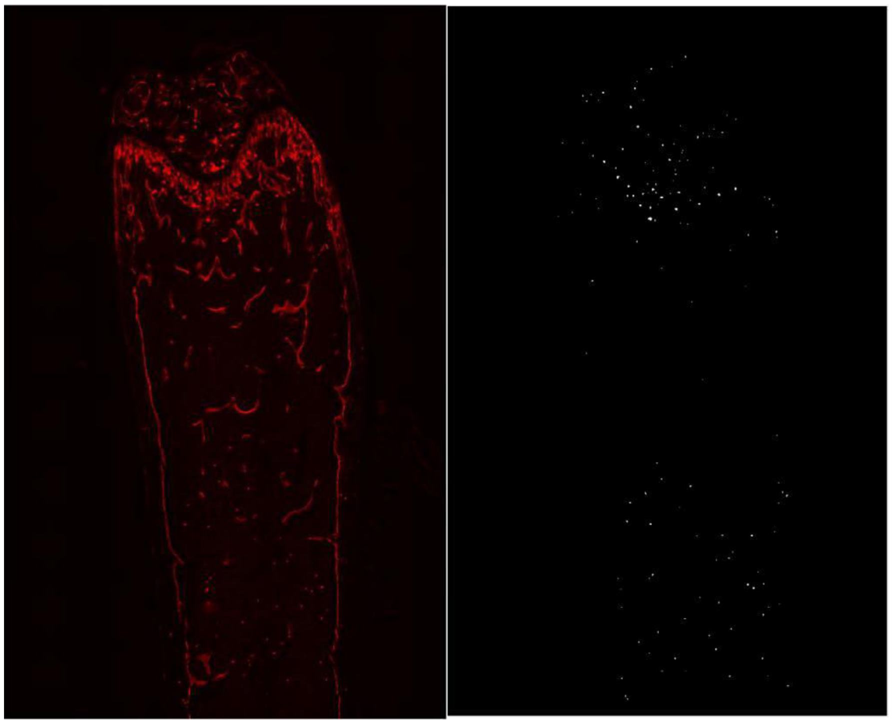
      Mineral 2

Osteoclast 1

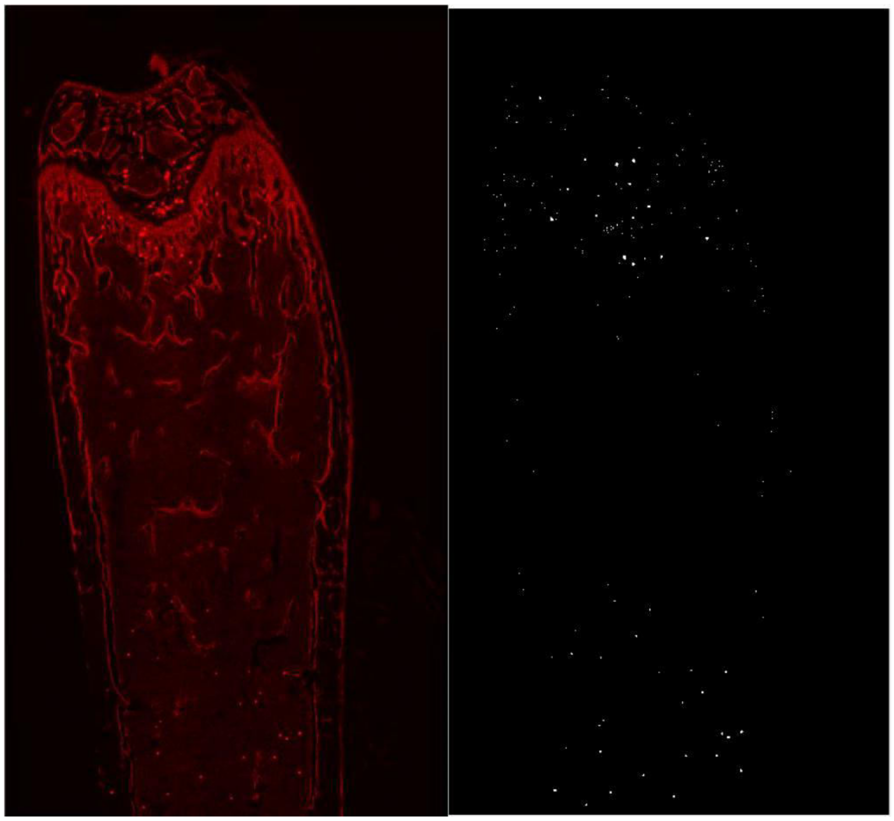


Second Pair

Mineral 1



Osteoblast 1



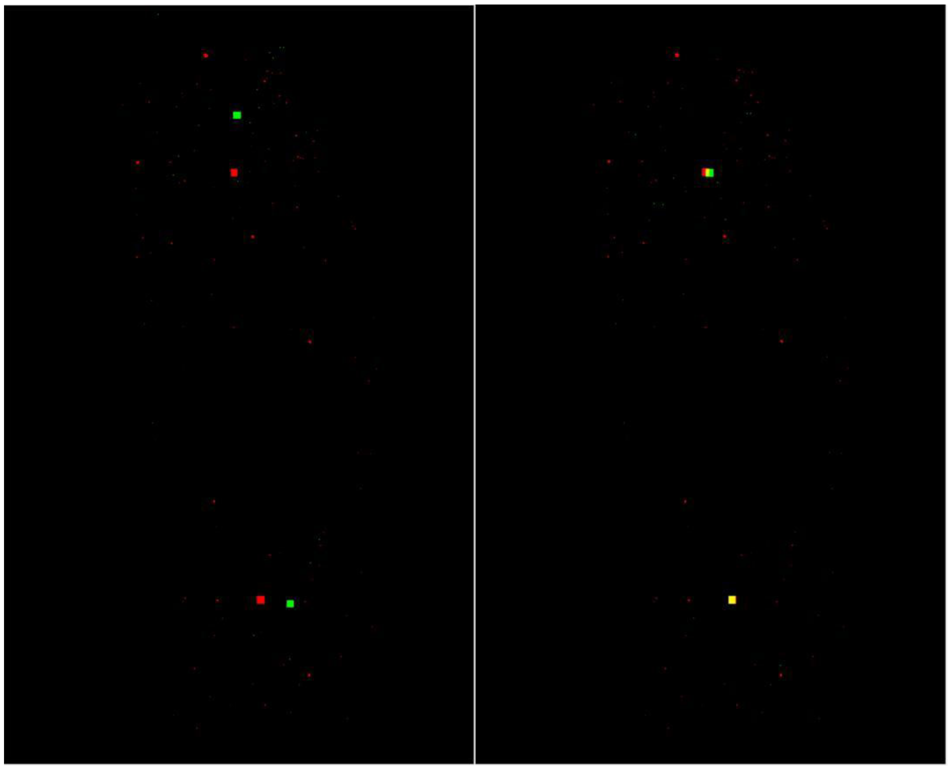
c. In order to register my image, I performed three stages of operations: rotation, scaling, and shifting. That is, after I retrieved the beads from part b., I implemented p1_image_alignment that took in the input of the reference beads, the test beads, and the test image itself. Transformations are done on the test beads and the test image simultaneously. In order to conduct the rotation, I cited a function by Jan Motl, a contributor from MathWorks, who implemented a rotation that utilizes any given image point as an axis.

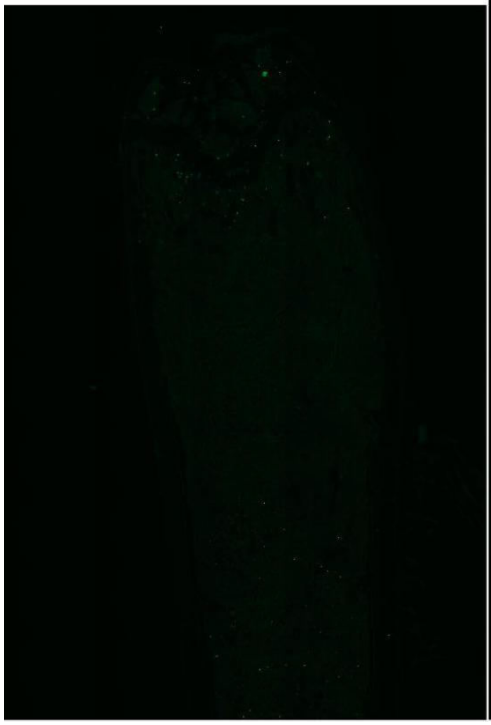
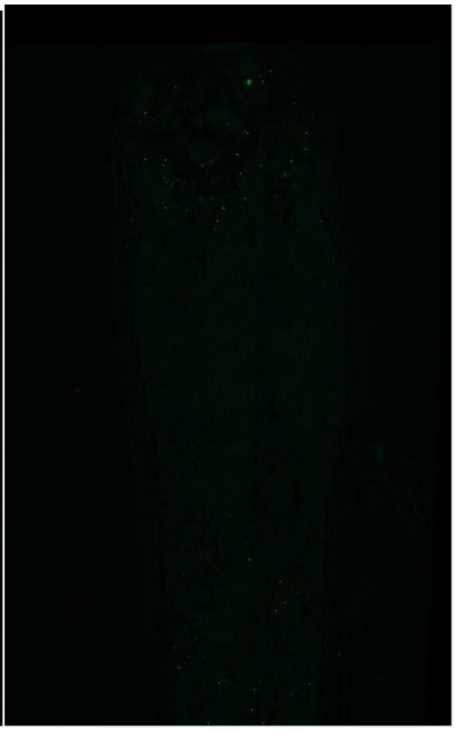
d. Since the difference in rotation angle between the test image and the reference image is not significantly high, I defined clusters above and below the vertical midpoint of each image. That is, I defined the upper and lower clusters for the test and reference images as the upper and lower halves of the image. Since the reference images all had a height of 10,400 pixels, I utilized a boundary at Y = 5,200.

e. For this project, I implemented k-th law filtering in rough_match.m, utilizing the Fourier transform of both images. I zero-padded the images along the bottom and the right. After utilizing the formula and finding the inverse Fourier transform of it, I found the maximum and subtracted the size of the test image from its coordinates. This value is the offset of the test image. I found that my function returned that the offset between the first pair of images was [172, -26], and that the offset between the second pair of images was much more significant at [-26, 1188].

Part F, G, H, I, J, K, L, M, N, O, P

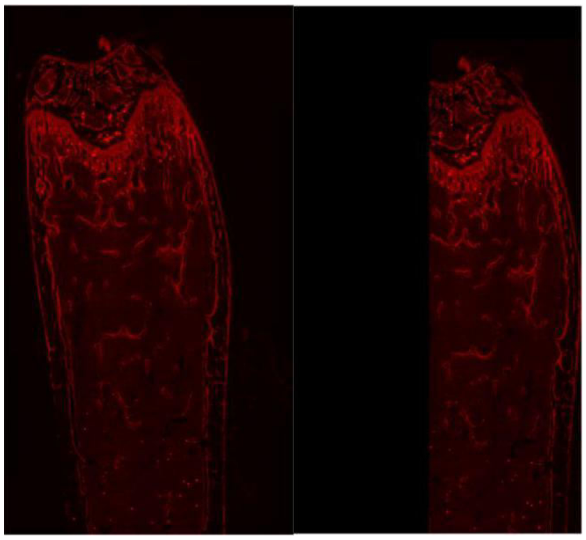
When running the p1_image_alignment function, I tried to create automatic procedure. I save copies of the image at each stage to highlight the clusters. I first store the bead data for both the reference and test images in an overlay matrix which stores it as a 3-D image. The reference beads are represented as red and the test beads are represented as green. Using the debugger, I am able to pause the function midway and show the locations of the beads before and after the transformation. Below is an example shown for the bead alignment of the first pair of images, and then below that the original and realigned test image. The transformation of Mineral 2 and Osteoclast 1 was successful.



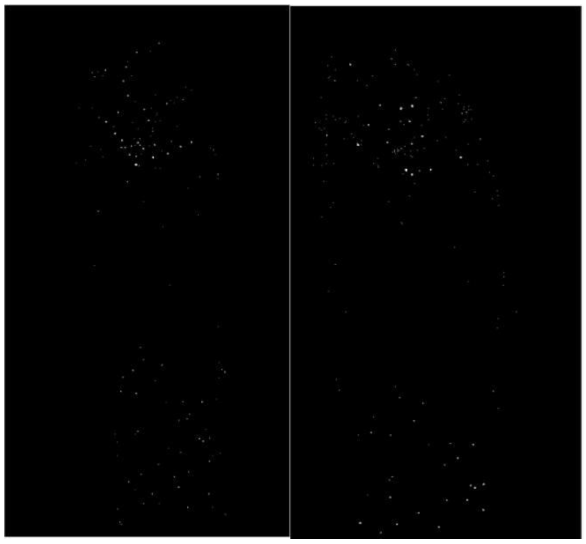



An interesting observation I made was that my program seems to return errors when the shifting distance between the centroids of the test and reference images is significant. This caused a lot of information to be lost after the initial transformation using the rough transformation information in part e. of [-26, 1188]. Because this distance is significant, much of the information about the beads was lost after the first transformation and rotation, creating the large discrepancy for the osteoblast data. Observed are the transformation using my program of the osteoblast data. Below that are the dots of the osteoblast data and the first mineral picture, where the shift between the centroids of the two are great.

Osteoblast 1 before and after:



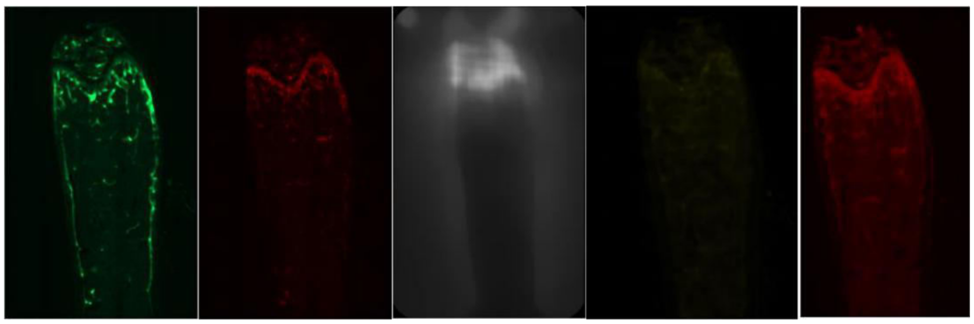
Distribution of beads of Mineral 1 (Left) and Osteoblast 1 (Right)



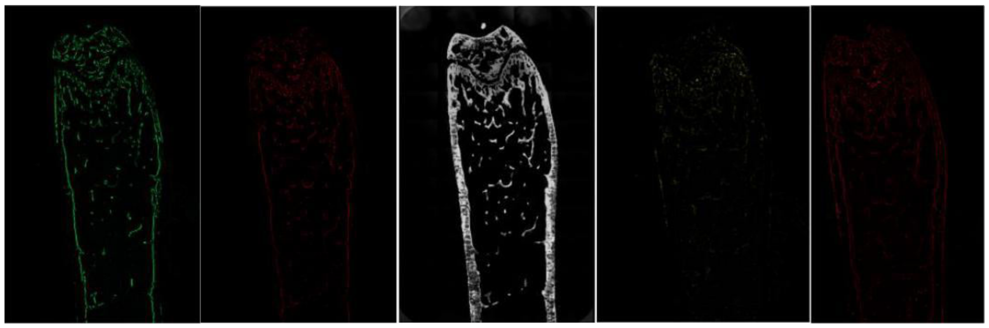
Part 2 and 3

Part 2

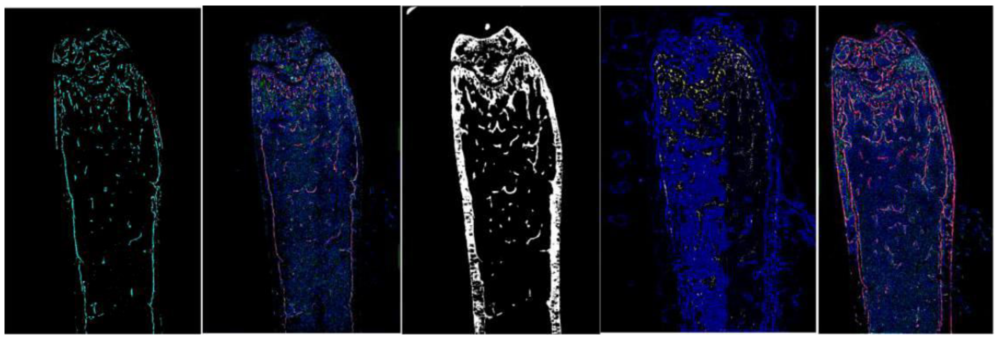
A. Using median filtering, we find the background of each image Calcein, AC, DIC, TRAP, and AP in that order.



B. We perform the subtraction.

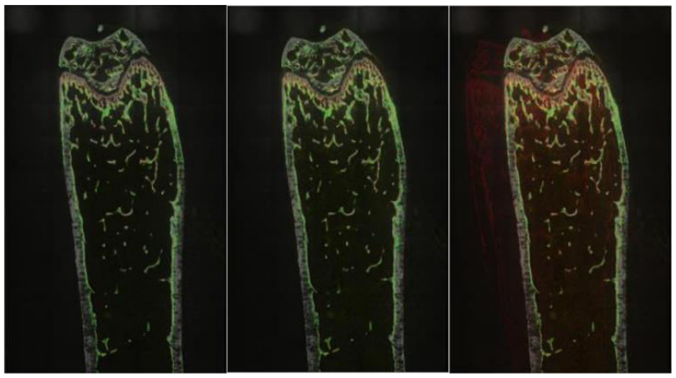


C.



Part 3

A, B, C



D.