COVID-19 Diagnostic System

William Rattan, Chase Alexander,
Kevin Tran, William Zhong

Overall goal/motivation

- Team was fairly new to data analysis of image processing
    - Resulting image processed by using Segmentation & Registration were interesting and their algorithms were complex but also constructive
- Main motivation came from the idea of learning something completely new
- Overall goal was to a create system for Segmentation & Registration that will successfully process Chest CT scans of COVID-19 patients, specifically located in the lungs.
- Compare the results of our processed DICOM image with another Chest CT scan that doesn't have COVID-19 present

Technologies used and overall framework

- SimpleITK
- Windows
- Python with Anaconda
- ImageJ
- VS Code
- GitHub

System Overview

- Segmentation
  - Bilateral edge-preserving smoothing followed by a histogram based binary threshold segmentation technique
- Registration
  - Intensity-based rigid registration using mutual information as our metric, a one-plus-one evolutionary optimizer; an Euler 3D transform with rotation around the center of the image
- Quantification
  - Experimented with our system’s parameters to find the values that produce the best visualization of COVID-19 in the lungs
  - Quantify the severity of the COVID-19 case in question
    - Comparing three CT scans of known COVID-19 patients of different severity levels (severe, intermediate, mild) with the results our system.

System Overview: Driver Program

1. Apply a bilateral edge-preserving smoothing to the CT scan in question
2. Followed by histogram-based binary thresholding
   a. Turns all pixels in the CT scan black except for pixels in the intensity range of interest
   b. Pixels within the range of interest are turned a dark gray
3. Repeat for a CT chest scan of an individual who has tested negative for COVID-19
   a. This time the pixels of interest are turned a brighter gray color
   b. This is done to make the combined image easier to analyze once these scans have been registered.
4. The resulting NIfTI files are displayed using ImageJ
5. Call registration on the 2 segmented images
   a. Healthy patient's scan is used as the fixed image
   b. Scan in question is used as the moving image
   c. In regions of overlap the intensities of the grays in both images produce an intensity closer to white
   d. In regions where there are differences we can determine which scan the highlighted portion comes from.

Registration Brief Overview

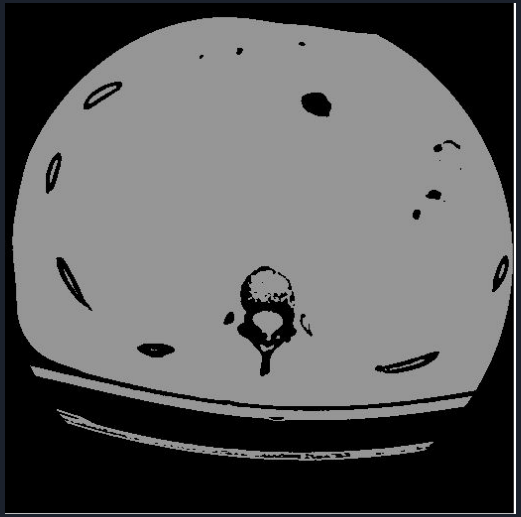
- Registration is a technique used in image processing to spatially align and combine two images into one
- 5 major components:
  - fixed image
  - moving image
  - metric
  - optimizer
  - transform
- The image that is being transformed is known as the moving image
- The image being referenced is known as the fixed image

Our Chosen Algorithm

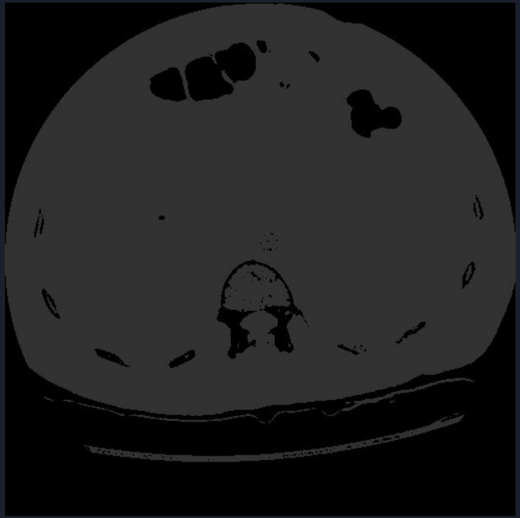
Intensity-based Rigid Registration with Mutual Information

- Intensity-based
  - Rely on intensity values to make our calculations/comparisons
- Rigid registration
  - Transform moving image using translations and rotations
- Metric: Mutual Information
  - Mutual Info = E(img1) + E(T(img2)) - E(img1, T(img2))
  - E(X) = entropy of X | T(X) = transform applied to X | E(X, Y) = joint entropy
- Optimizer: one-plus-one evolutionary
  - Based on a probabilistic, evolutionary approach
  - From the parent transformation, a child transformation is generated by slight modification
  - If the child transformation is better, it is selected as the new parent transformation
- Transform: Euler 3D
  - Applies a rotation and translation to the moving image given 3 Euler angles and a 3D translation
  - Specified rotation to be about the center of the image

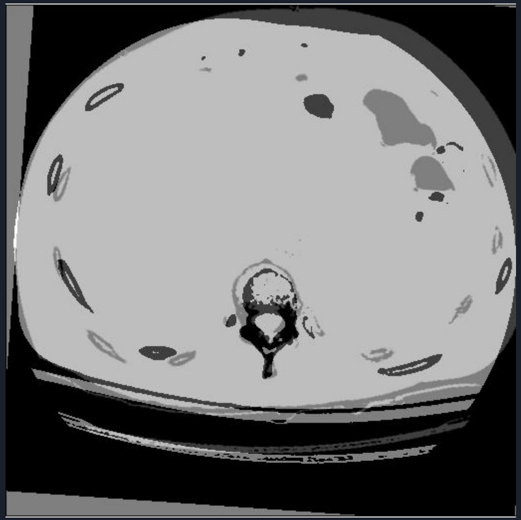
Registration Example Images



Fixed Image



Moving Image



Registered Image

Segmentation Intro

- Image segmentation is the division of an image into regions/categories
- Every pixel in a image is allocated to one of a number of these categories
- Pixels in same category have similar grayscale values and form a connected region
- Neighboring pixels have different values and are in different categories
- The goal of segmentation is to simplify and/or change the representation of an image into something more meaningful and easier to analyze.
- For our system we choose an Edge-preserving smoothing filter followed by a thresholding algorithm to segment our image.

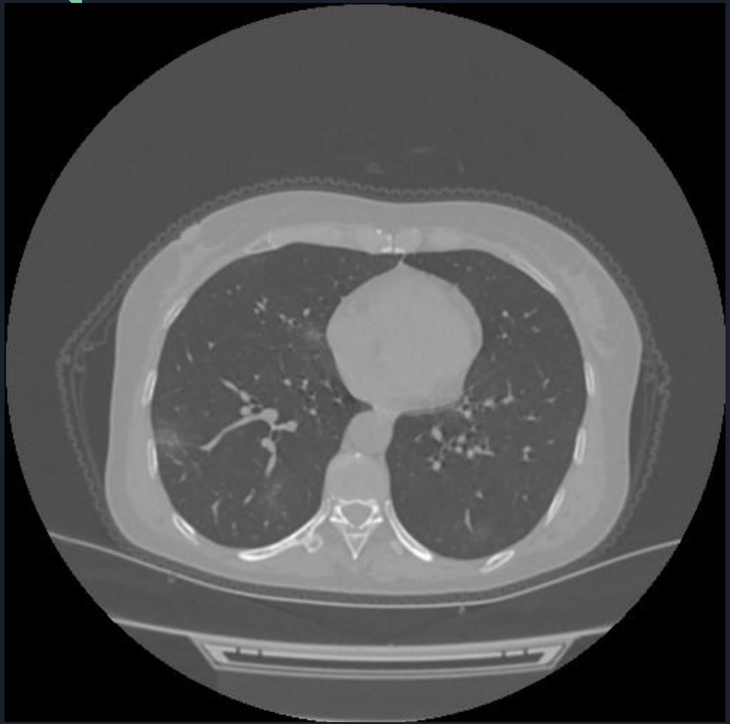
Edge-preserving smoothing

- Smoothing is a imaging process where the filter smooths away textures and noise from an image/data.
- However more often than not, important details are also smoothed away during the process
- The solution is to have a Edge-Preserving Smoothing filter, that smooths an image while retaining the sharp edges and details of the image.
- There are many different filters to achieve this, for our project we choose the Bilateral filter.

Bilateral

- The Bilateral filter is a non linear edge preserving filter.
- It is very similar, almost identical, to a Gaussian smoothing filter
- There is one key difference between Gaussian Smoothing and Bilateral.
- In addition to smoothing an Image we also have to rescale the image to a grayscale[0,255] so the thresholding algorithm can work.
- The bilateral filter in SimpleITK takes in two values for the sigma for both range and domain. As Well as a value for the range of Gaussian Samples.

Before Smoothing

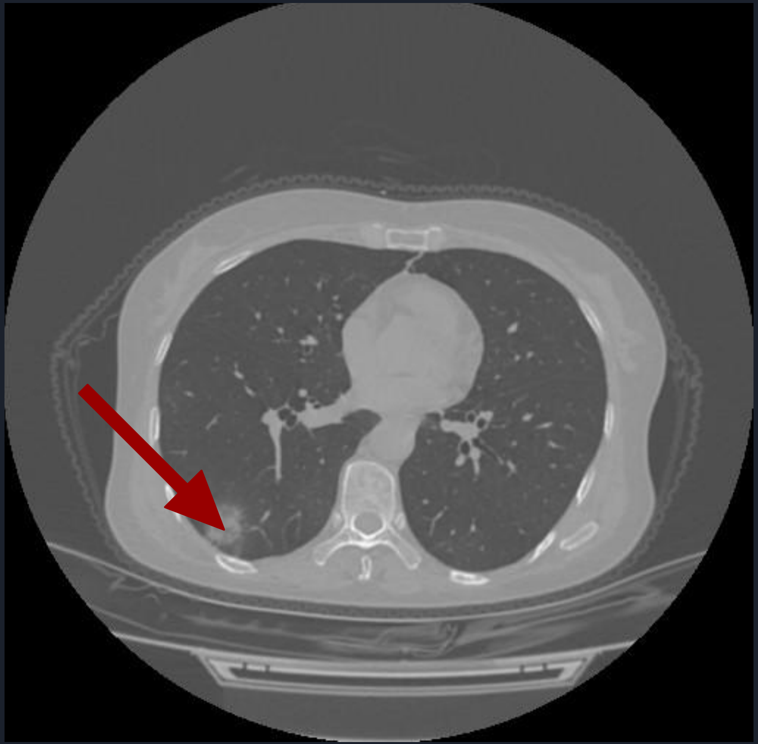


After Smoothing

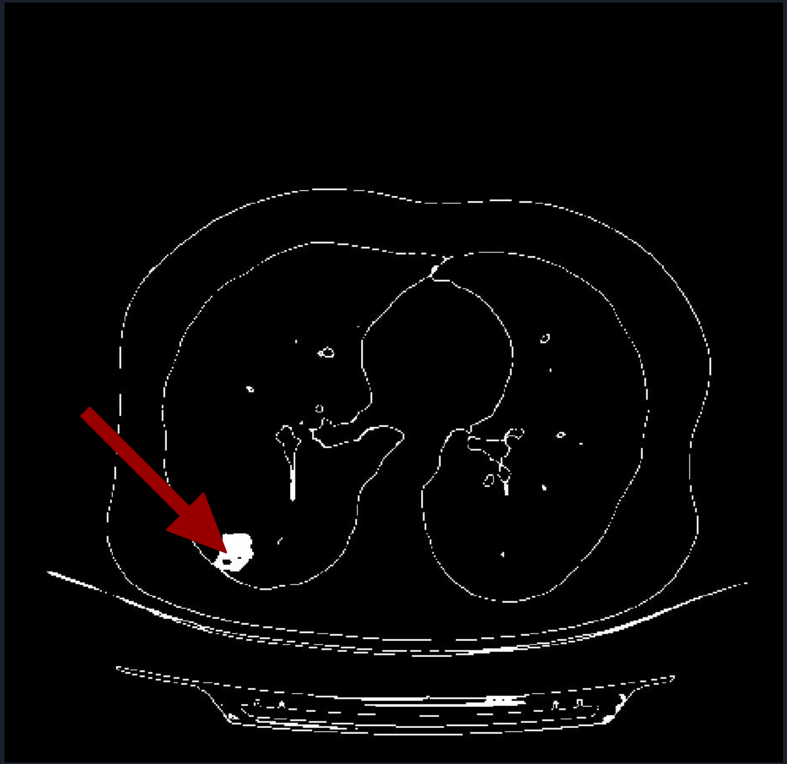
Thresholding

- The binary threshold technique to identify which pixel is black or white, and it is histogram based.
- Produces an output image whose pixels are either one of two values(OutsideValue and InsideValue), depending on whether the corresponding input image pixels lie between the two thresholds (LowerThreshold and UpperThreshold).
- A threshold values is used to classify pixel values as either 0 or 1, and further, converting gray-scale images to binary

Before Thresholding



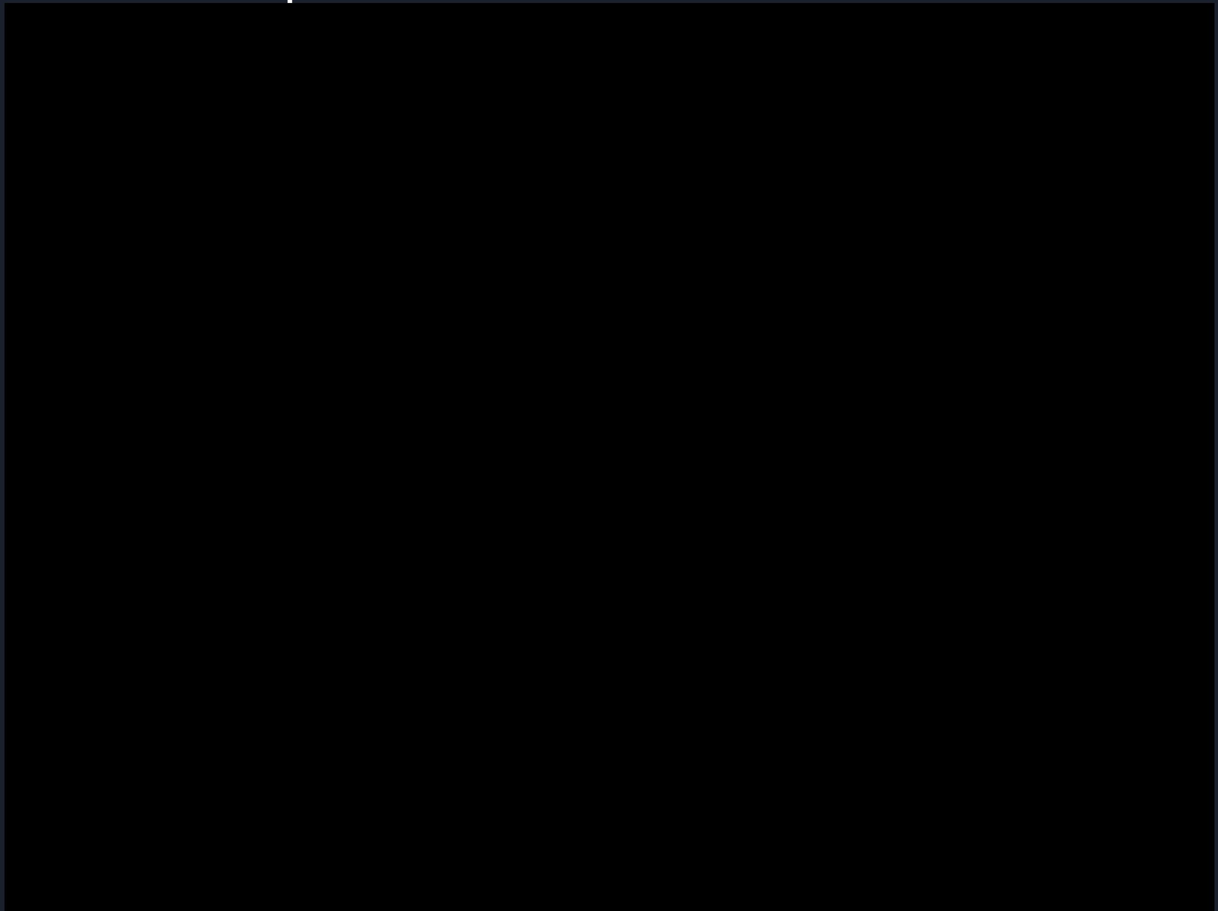
After Thresholding



Quantification

1. Experiment with different Values of our system components
    a. Bilateral Smoothing default values are 4.0 for domain,50.0 for range(Takes too long to compile)
    b. 0.8 for domain, 10.0 for range was sufficient enough with much faster compile time
    c. Thresholding values differ per picture, for our patients lower threshold of 110 and upper threshold of 135 gave us the best results for detecting covid in the lower left lung
2. Determine the severity of the COVID-19 case in the CT scan in question by examining the resulting segmented and registered image
    a. Can identify areas highlighted in the questionable scan that are not highlighted in the healthy scan as potential COVID spots
    b. Examine their location, shape, and size to make a final determination
    c. Can use the number of potential COVID spots and their size to determine the severity of the case by comparing the scan with images depicting mild, intermediate, and severe cases.

Final Output



Challenges

- Having our system work with a 3D image format
- Experimenting with certain values made the time to compile very long(Sometimes hours)
- Lost a Team member halfway through the project
- Simple ITK library had a steep learning curve
- Had to change smoothing algorithm halfway through project

What we learned

- Different types of 3D imaging and visualization
- Different types of 3D image formats and the physics behind creating a 3D image
- Applications of 3D imaging in medical use
- How to use Simple ITK
- How to use ImageJ
- How to implement a registration system and the various methods/algorithms to create the system
- How to implement a segmentation system and the various methods/algorithms to create the system

Future Work

- Make a GUI for our system using VTK
- Change one of the segmented images to a different color before registering them
- Improve the effectiveness of algorithms to reduce the number of false positives of Covid-19
- Find better/more data
- Make a subsystem that creates a normalized lung scan