Presentation

Drop on Thin Film

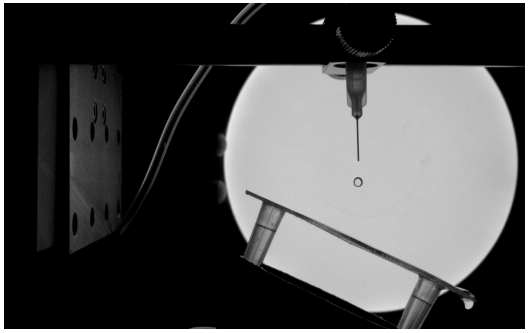
Kulwinder Kaur, Harikesh Kushwaha

Indian Institute of Technology, Delhi

June 15, 2022

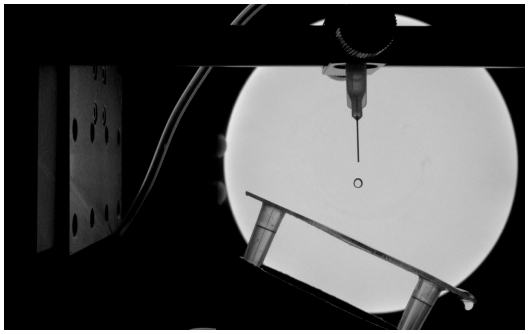
Introduction

1. **The Problem**
   We are given a video of a drop leaving a needle and falling through a thin film. We are required to find the position of the drop in the film.



Introduction

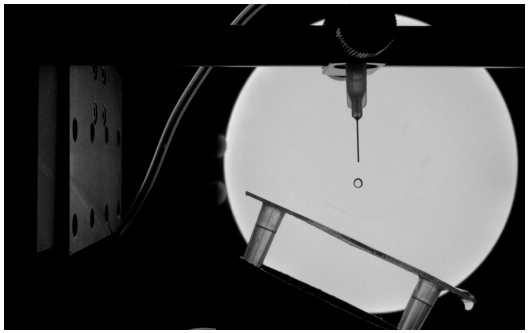
1. **The Problem**
   We are given a video of a drop leaving a needle and falling through a thin film. We are required to find the position of the drop in the film.



2. **Background Information**

Introduction

1. **The Problem**
   We are given a video of a drop leaving a needle and falling through a thin film. We are required to find the position of the drop in the film.



2. **Background Information**
   - **How Computer Stores Image?**
     Every image is made up of pixels. These pixels are stored in the form of an array with the shape (Height, Width, Colour Channel).Each of the pixels that represents an image stored inside a computer has a pixel value which describes how bright that pixel is, and/or what color it should be.

Background Information

- **Various Type of Images**
  Images can be classied into various types on the basis of how many colour channels they have. Some are:

Background Information

- **Various Type of Images**
  Images can be classied into various types on the basis of how many colour channels they have. Some are:
  - Colour Image
    Having 3 or 4 colour channels.

Background Information

- **Various Type of Images**
  Images can be classied into various types on the basis of how many colour channels they have. Some are:
  - Colour Image
    Having 3 or 4 colour channels.
  - Grey Image
    Having a single colour channel, hence the name grayscale.

Background Information

- **Various Type of Images**
  Images can be classied into various types on the basis of how many colour channels they have. Some are:
  - Colour Image
    Having 3 or 4 colour channels.
  - Grey Image
    Having a single colour channel, hence the name grayscale.
  - Binary Image
    Having a single colour channel. Also, the value of pixels are either 0 or 1.

The Solution

1. **Preprocessing**
   Before we can try extracting the center of the drop, we need to do some preprocessing. Here are the steps:

The Solution

1. **Preprocessing**
   Before we can try extracting the center of the drop, we need to do some preprocessing. Here are the steps:
   - **Loading The Original Image**
     The image provided are grayscale in nature. We can use matplotlib or Pillow libraries to read them. This will give us a numpy array on which we can work.

The Solution

1. **Preprocessing**
   Before we can try extracting the center of the drop, we need to do some preprocessing. Here are the steps:
   - **Loading The Original Image**
     The image provided are grayscale in nature. We can use matplotlib or Pillow libraries to read them. This will give us a numpy array on which we can work.
   - **Cropping The Image**
     Only a small portion of the image is useful to us as the drop is remaining in a region which spans for just about 500×500 px. That's why we cropped the image to contain just this region.

Preprocessing

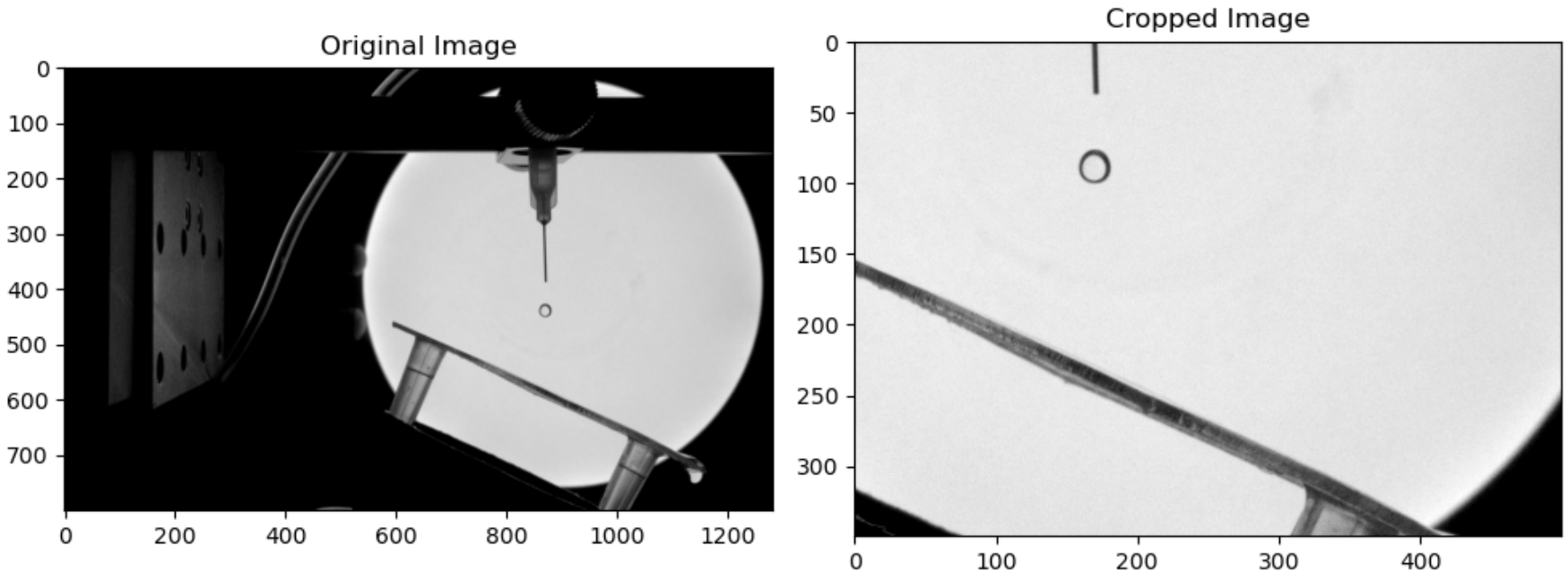


Figure: Image Before and After Cropping

Preprocessing

- **Thresholding Image**
  Thresholding is the process of converting an image to a binary image. The thresholding is done by setting a certain value as the threshold. If the pixel value is greater than this threshold, it is set to 1, else it is set to 0.

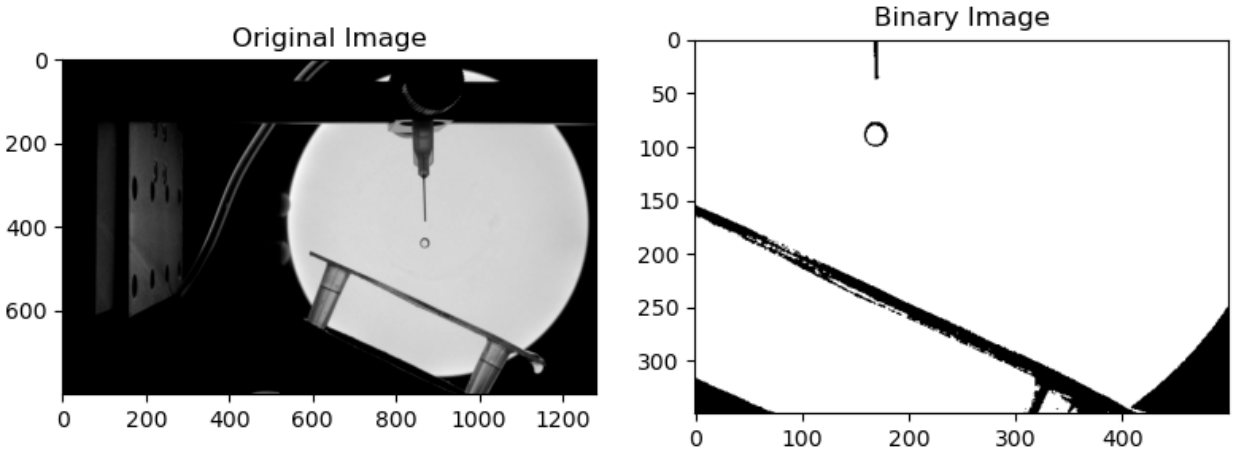
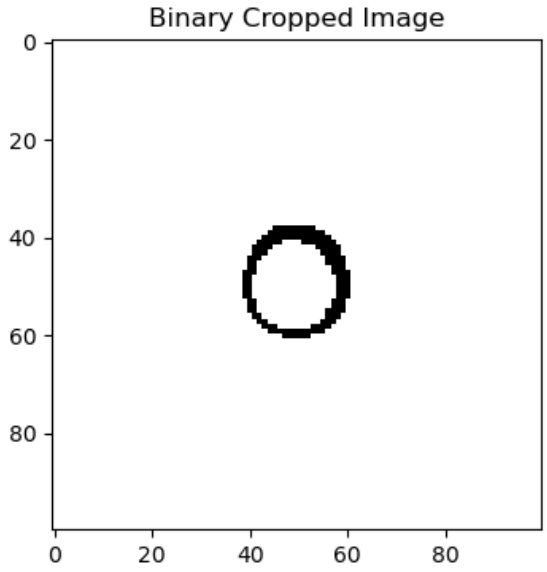


Figure: Image Before and After Cropping and Thresholding

The Main Method



The image above shows the drop with its surrounding restricted to just a 100 pixels. The image is a binary one, meaning that the pixel value is either 0 or 1. Zero for the black points and One for the white points. The idea behind the method we implemented to find the center of the drop is the following:

The Main Method

The image above shows the drop with its surrounding restricted to just a 100 pixels. The image is a binary one, meaning that the pixel value is either 0 or 1. Zero for the black points and One for the white points. The idea behind the method we implemented to find the center of the drop is the following:

- Loop through the rows and find all the black pixels for each row.

The Main Method

- The the position of the first such pixel will give the top-most point of the drop and the last such pixel will give the bottom-most point of the drop.

The Main Method

- The the position of the first such pixel will give the top-most point of the drop and the last such pixel will give the bottom-most point of the drop.
- Loop through the columnss and find all the black pixels for each columns.

The Main Method

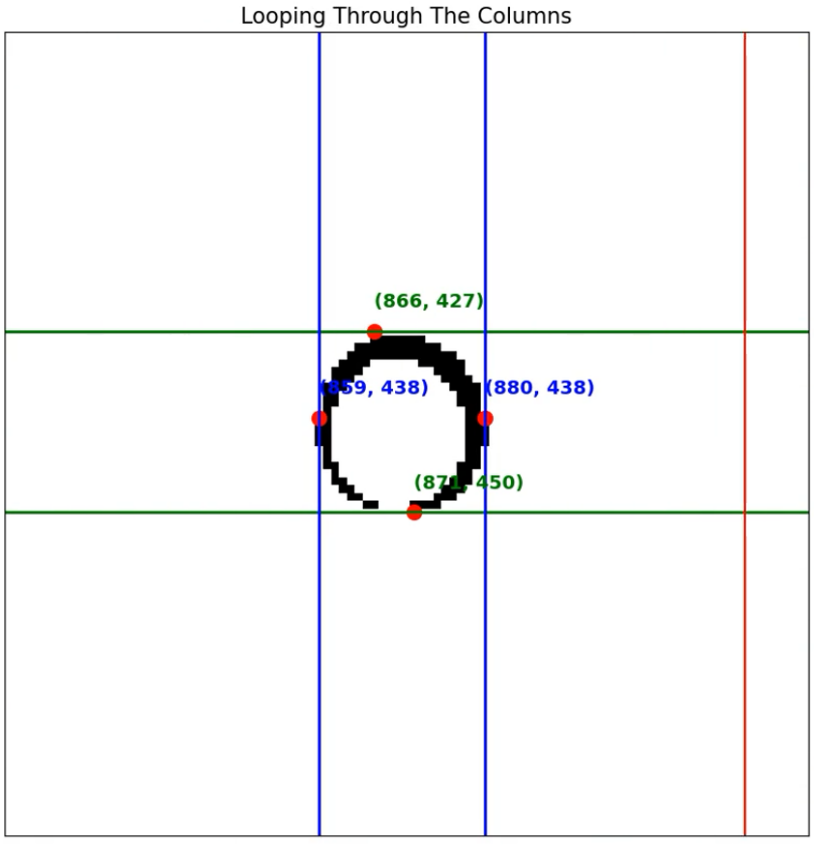
- The the position of the first such pixel will give the top-most point of the drop and the last such pixel will give the bottom-most point of the drop.
- Loop through the columnss and find all the black pixels for each columns.
- The the position of the first such pixel will give the left-most point of the drop and the last such pixel will give the right-most point of the drop.

The Main Method

- The the position of the first such pixel will give the top-most point of the drop and the last such pixel will give the bottom-most point of the drop.
- Loop through the columnss and find all the black pixels for each columns.
- The the position of the first such pixel will give the left-most point of the drop and the last such pixel will give the right-most point of the drop.
- Once we have the top-most, bottom-most, left-most and right-most points of the drop, we can find the center as well as the horizontal and vertical radii of the drop.

The Main Method

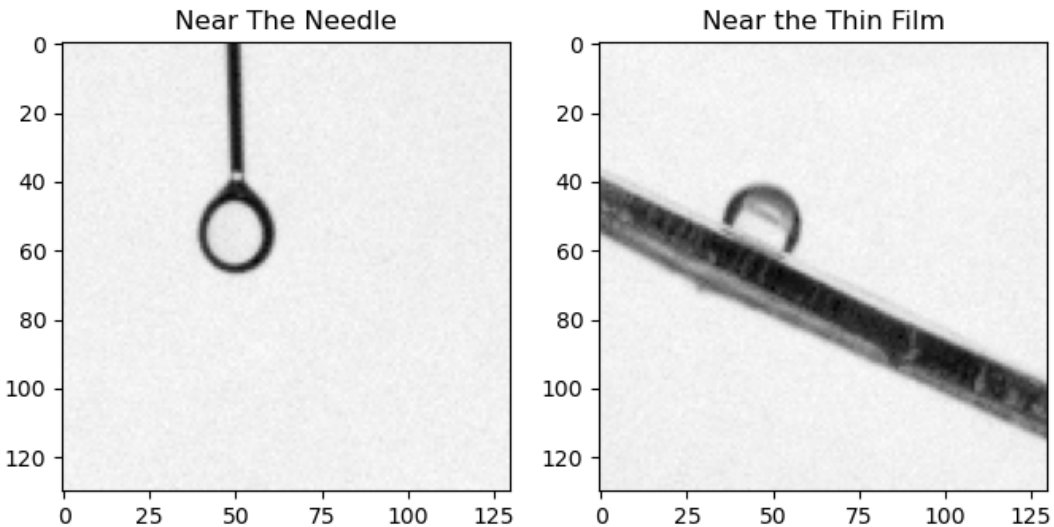
Here is an animation showing the process of finding the center of the drop.



Problems With The Main Method

The method implemented above works fine for most cases but it does have some problems:

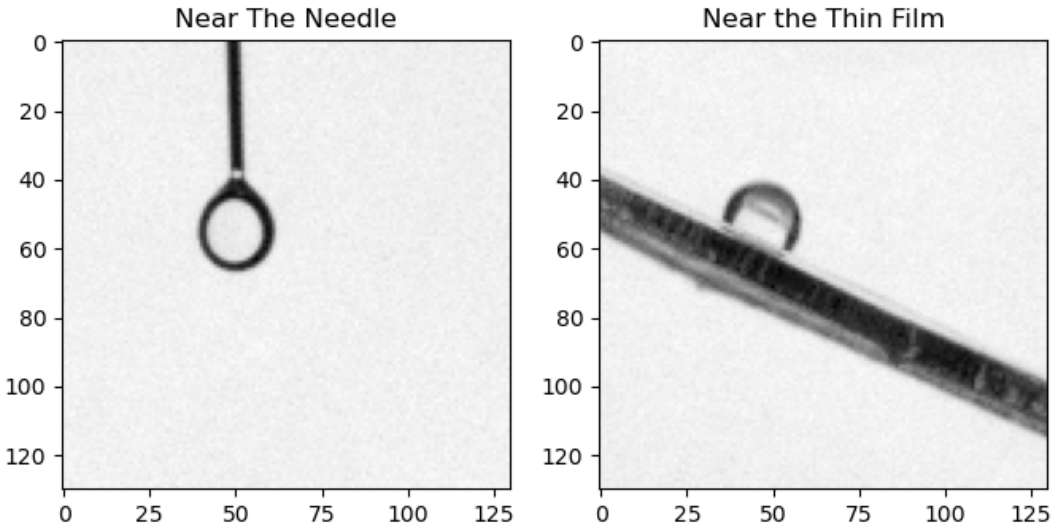
- When the drop is just leaving the needle and when the drop is leaving or near the edge of the film, the method fails.



Problems With The Main Method

The method implemented above works fine for most cases but it does have some problems:

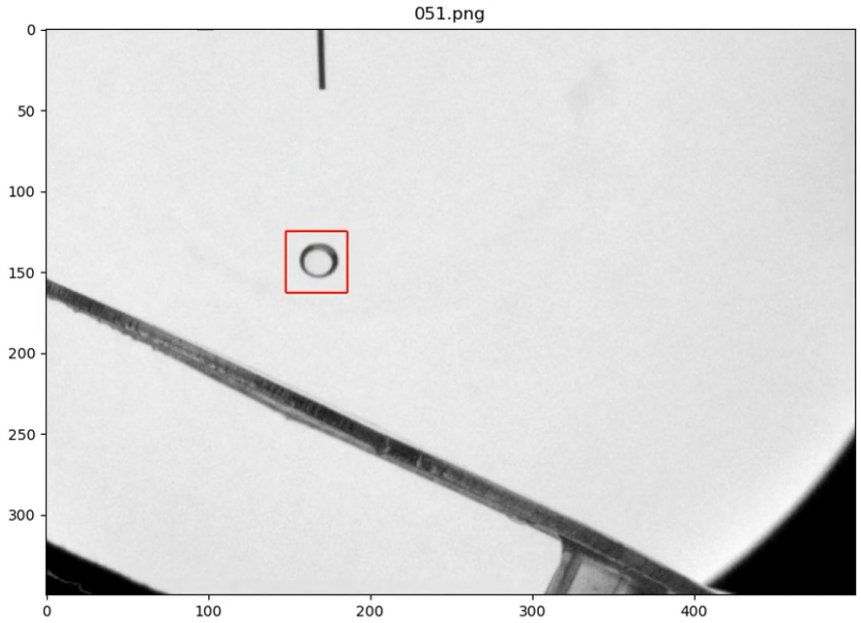
- When the drop is just leaving the needle and when the drop is leaving or near the edge of the film, the method fails.



- We need to specify the crop coordiantes for each image. This leaves the method from being generic.

Problems With The Main Method

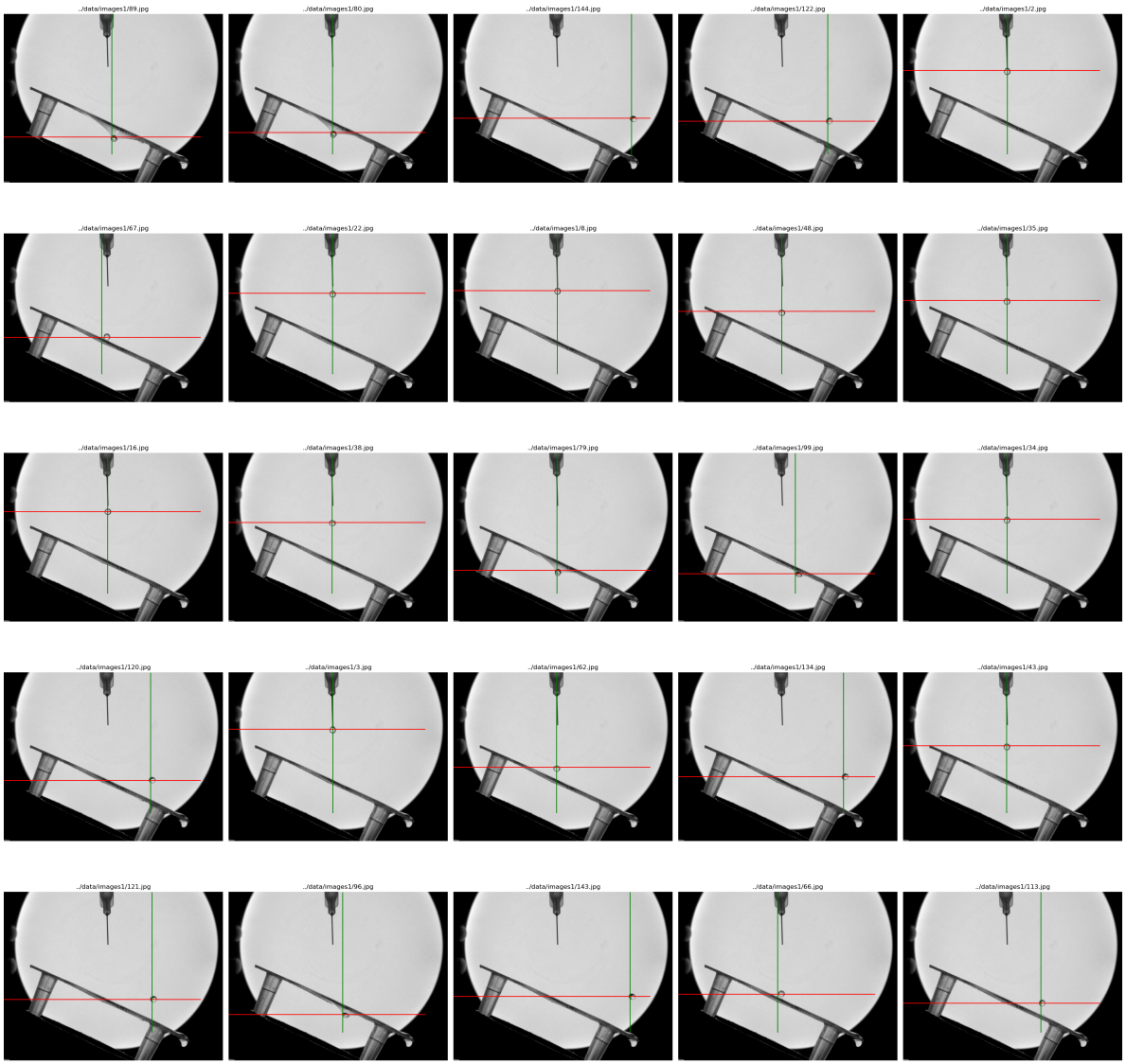
The second problem can be solved partially by using the dynamic cropping. In this method, we decide which portion of the image to crop on the basis of the center of the drop in previous image. We define a height and width and calculate the bouning box for the crop. Here is an animation showing the same.



Of course, we still need to provide the crop coordiantes for a couple of frames. Even after this, the method gives results for just about 140 images out of 171.

Problems With The Main Method

Here are some sample images and their center determined using the dynamic cropping method.



Subtracting The Background

We saw that the dynamic cropping method is not robust enough. The problem arised because the needle and the thin film in the image acts as noise which our method fails to detect. To overcome this problem we subtracted the image in consideration with a reference image.

Subtracting The Background

We saw that the dynamic cropping method is not robust enough. The problem arised because the needle and the thin film in the image acts as noise which our method fails to detect. To overcome this problem we subtracted the image in consideration with a reference image.

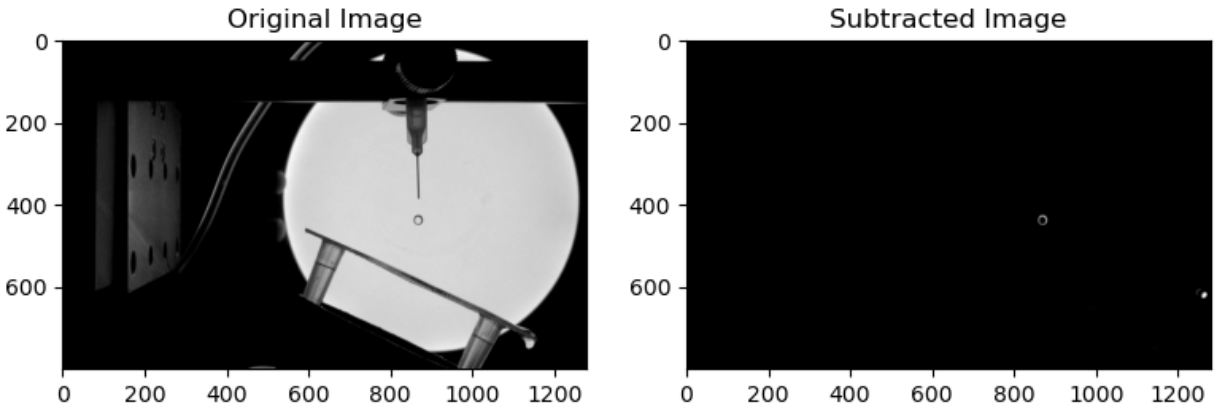


Figure: Original and Subtracted Image

Subtracting the Background

- As we can see, after subtracting from a reference image, the final image does not contain any background noise, just the drop. This means that the previous method should work very well.

Subtracting the Background

- As we can see, after subtracting from a reference image, the final image does not contain any background noise, just the drop. This means that the previous method should work very well.
- This was the case. This modification results in a better result. We were able to extract the center of more images (over 160 out of 172 images). Furthermore, the center extracted were more "accurate". The next slide shows some of the sample images with their center determined using the modified method.

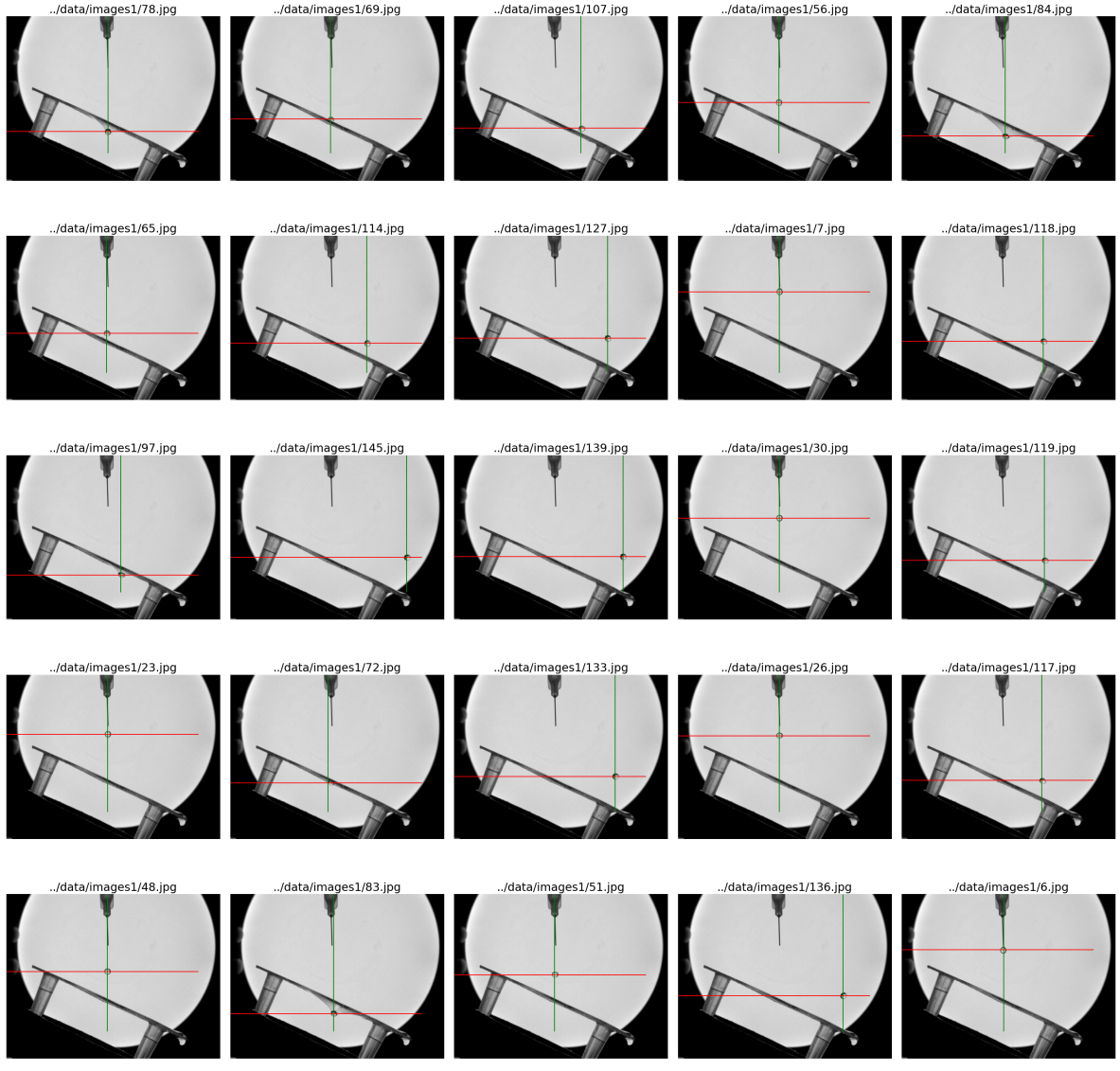


Figure: Sample Images with Their Center Determined Using the Modified Method

Fitting an Ellipse

1. **Motivation**
   If we see the animation of the drop falling, we'll notice that the drop is oscillating and rotating. We wanted to see whether we can catch this rotation using some method. This was the one reason we decided to use the ellipse fitting method. Another reason was to further increase the accuracy of the center extracted.

Fitting an Ellipse

1. **Motivation**
   If we see the animation of the drop falling, we'll notice that the drop is oscillating and rotating. We wanted to see whether we can catch this rotation using some method. This was the one reason we decided to use the ellipse fitting method. Another reason was to further increase the accuracy of the center extracted.
2. **Determining the Coordiantes**
   First step was to determine the coordinates of all the points which form the 'circumference' of the drop. This was determined by looping through the rows and columns and finding all the white pixels, which are the points on the circumference of the drop.

Fitting an Ellipse

Here is an image showing how the points on the circumference are determined.

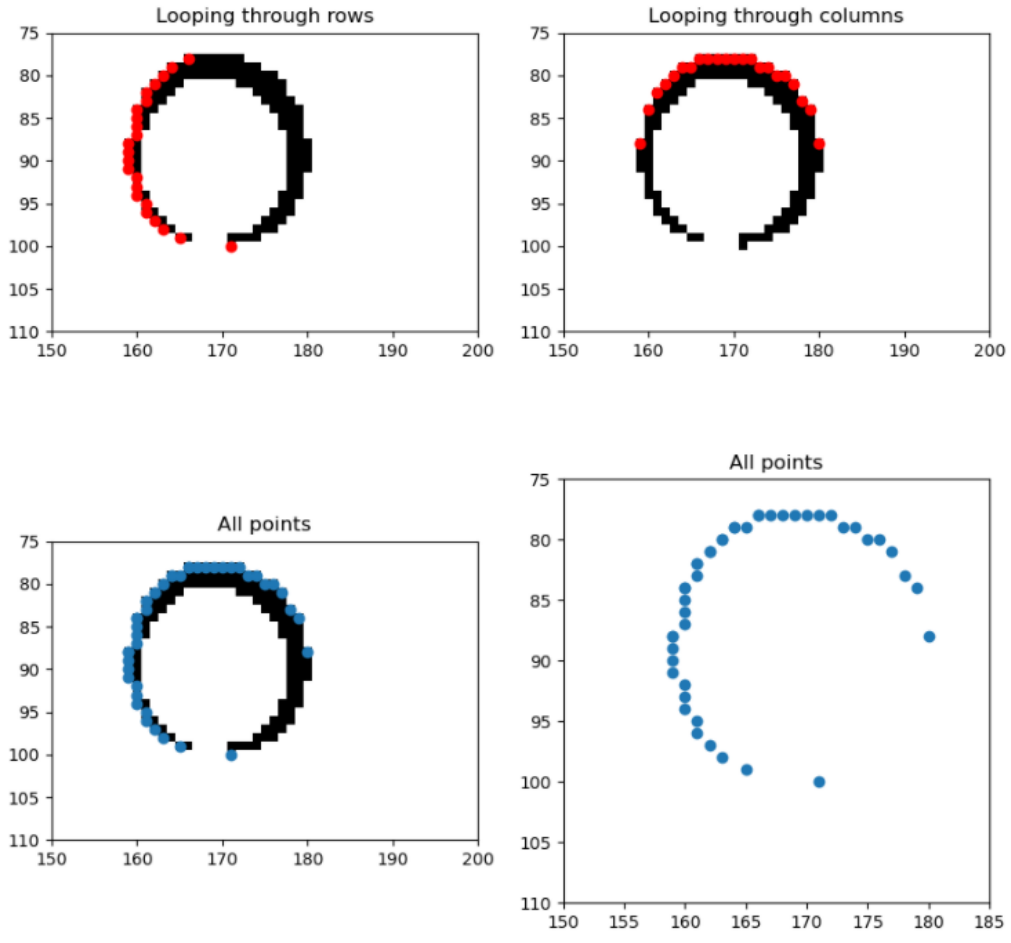


Figure: Points on the Circumference of the Drop

Fitting an Ellipse

Once we have the coordiantes of all the points, we used the EllipseModel class from the skimage library to fit an ellipse to the points. The class fits an ellipse and return the parameters of the ellipse:

- The coordiantes of the center
- the semi major and minor axes
- The angle the major axis of the ellipse makes with the x-axis

Fitting an Ellipse

Once we have the coordiantes of all the points, we used the EllipseModel class from the skimage library to fit an ellipse to the points. The class fits an ellipse and return the parameters of the ellipse:

- The coordiantes of the center
- the semi major and minor axes
- The angle the major axis of the ellipse makes with the x-axis

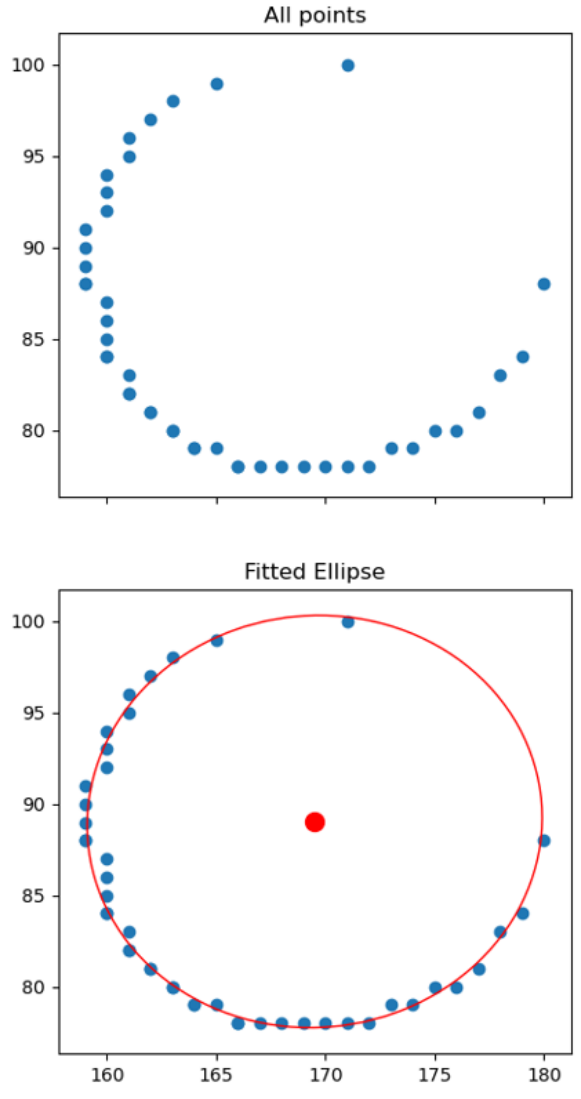


Figure: Fitting an Ellipse

Fitting an Ellipse

Though we were not able to infer the roation of the drop, this method works better than the previous methods; in a sense that the center determined using this are more accurate. Here are some samples.

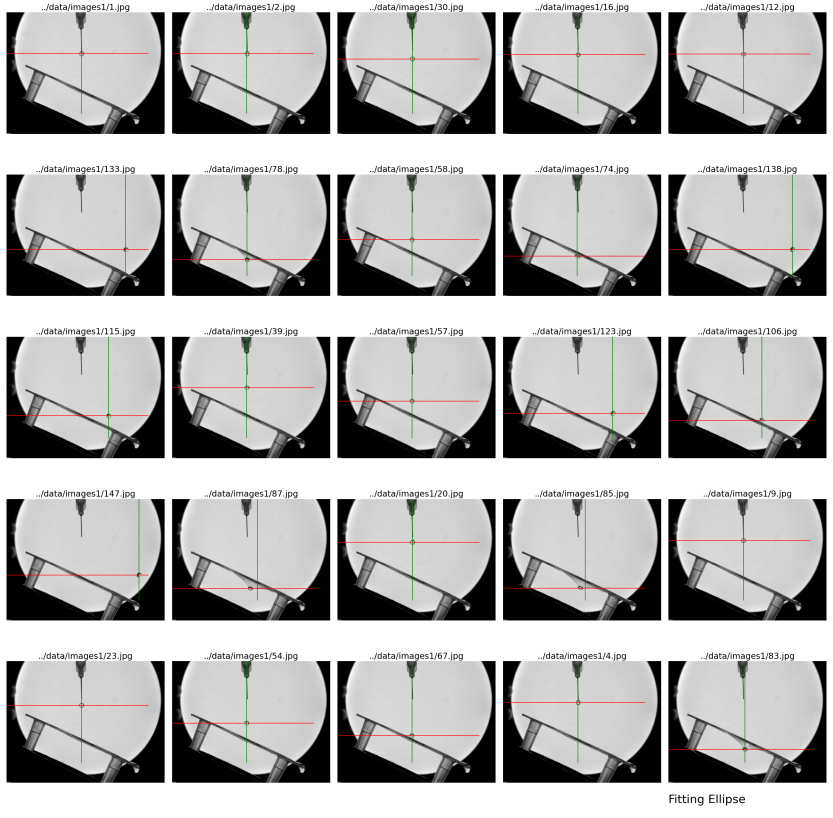


Figure: Sample Images by fitting an Ellipse

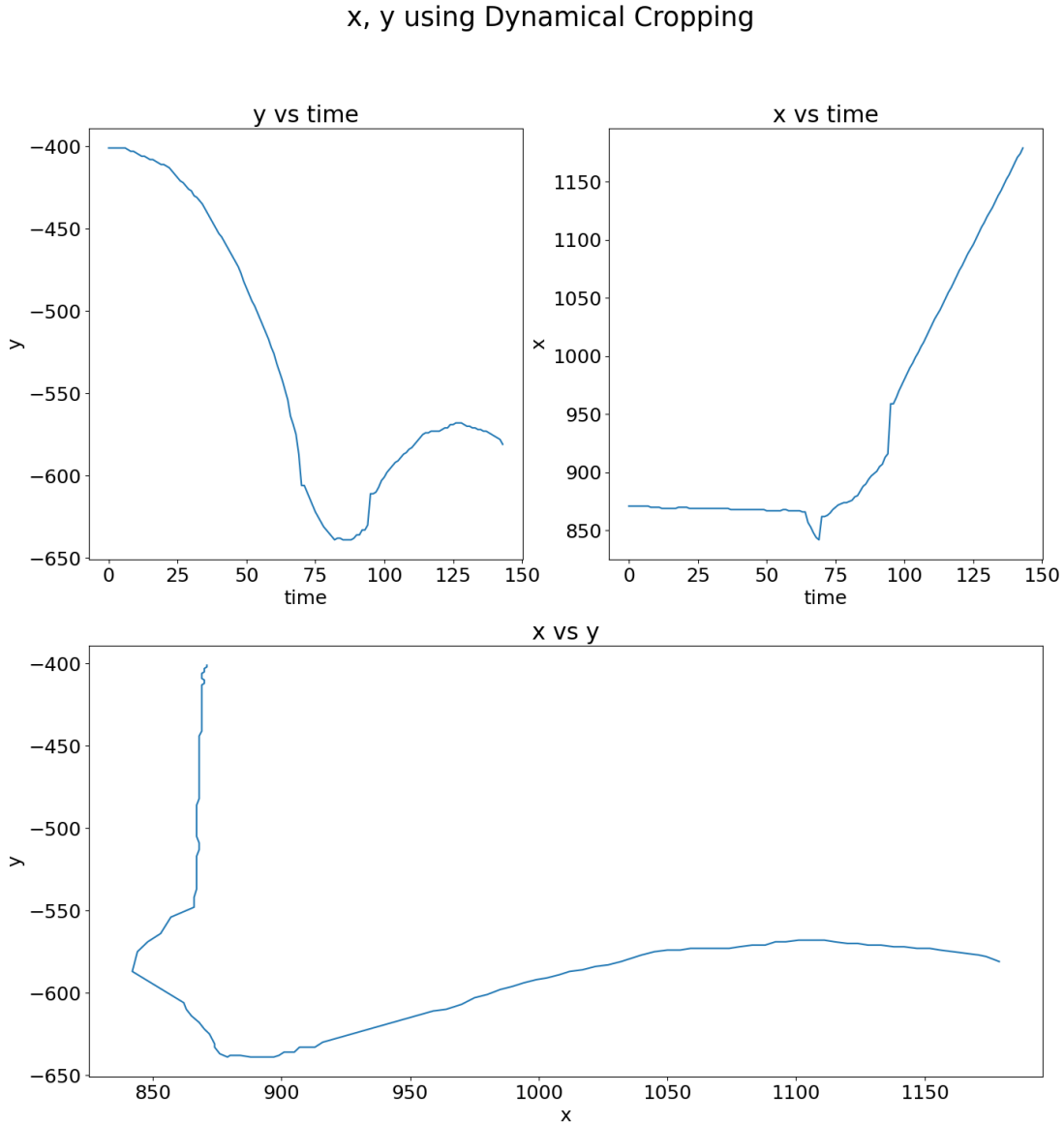
Preliminary Analysis

Though our main focus till now has been to extract the coordiantes of the center of the drop with more and more accuracy, we also did some preliminary analysis on the extracted data, mainly to determine whether our method is working or not and to compare the different approaches we took.

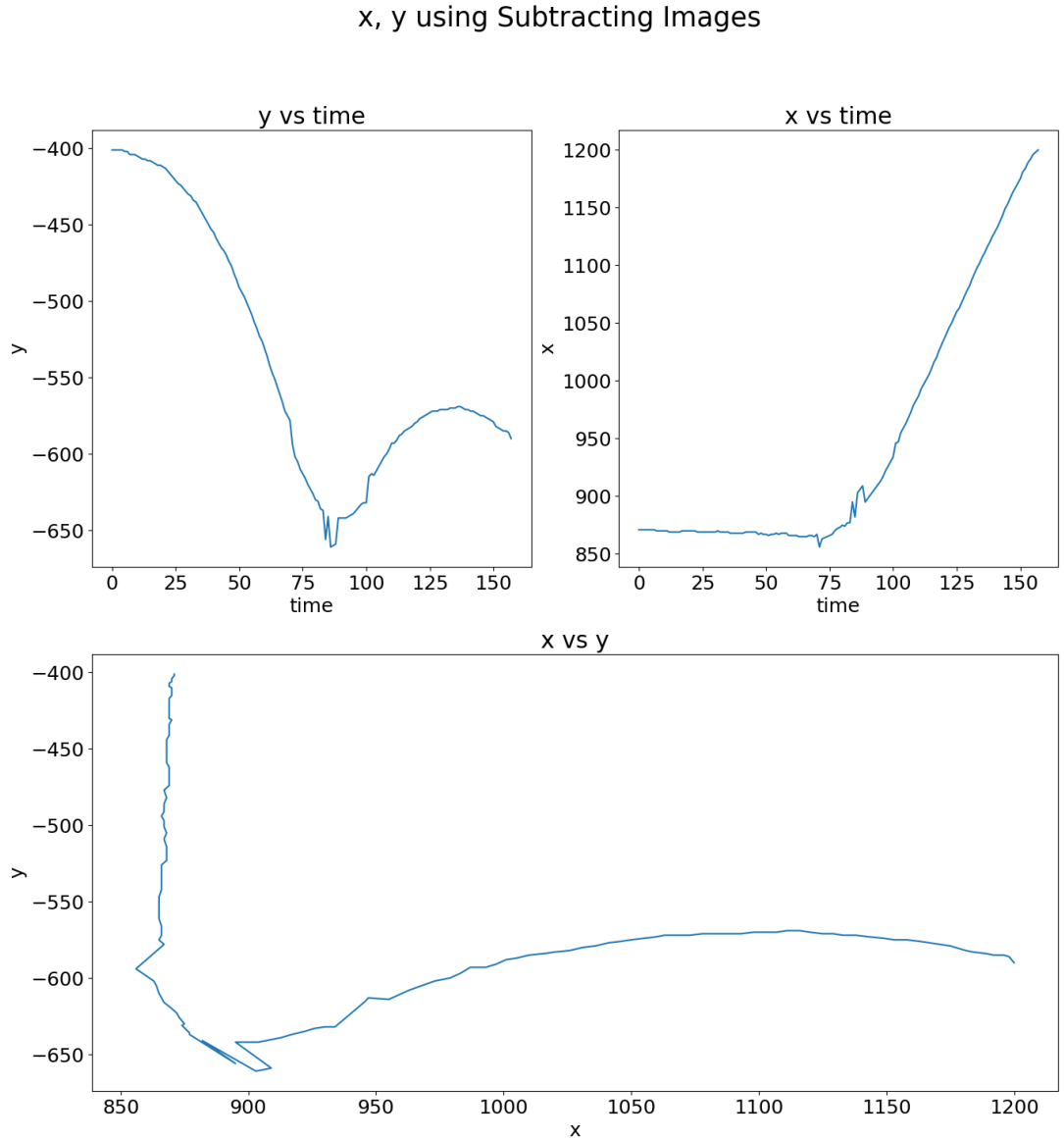
Preliminary Analysis

Though our main focus till now has been to extract the coordiantes of the center of the drop with more and more accuracy, we also did some preliminary analysis on the extracted data, mainly to determine whether our method is working or not and to compare the different approaches we took.
So, we made some plots. We plotted x, y, v_x, v_y, r_1, r_2 with time. We also plotted x vs y, v_x vs v_y, r_1 vs r_2. However, the most illuminating plots are those involving x and y.
Next few silde will show the plots of x and y with time as well as for x with y. The plots are made using "raw data", that is, we did not make any smoothing or any other processing.
We are working on some smoothing techniques. But it is work in progress.

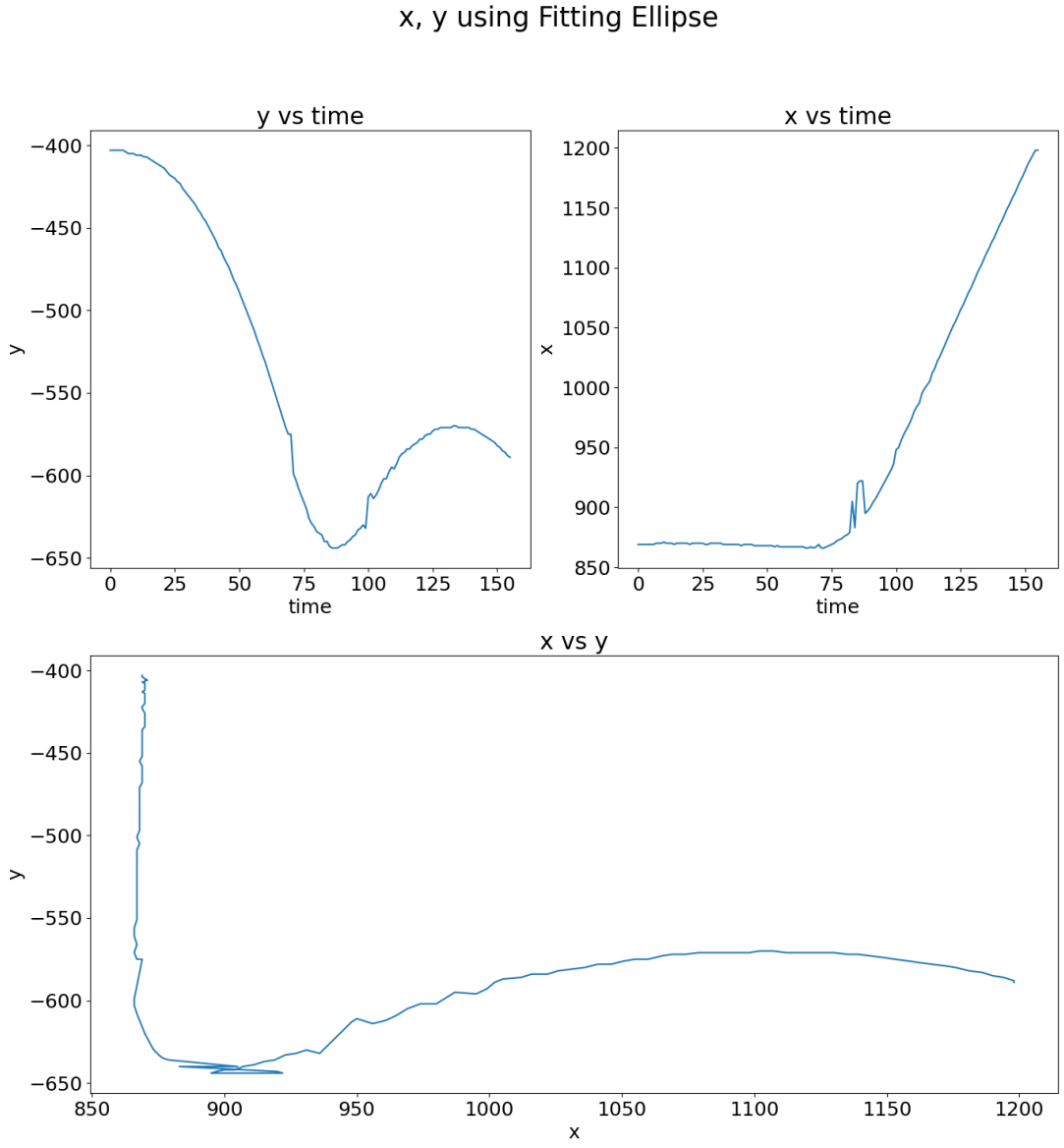
Preliminary Analysis



Preliminary Analysis



Preliminary Analysis



What Next?

1. **Smoothing**
   We are working on some smoothing techniques, like taking moving averages or using convolution.

What Next?

1. **Smoothing**
   We are working on some smoothing techniques, like taking moving averages or using convolution.
2. **Increasing Accuracy**
   Though fitting an ellipse is working very good, we are thinking of some ways to increase its accuracy even more. The main techniques, we are trying to implement are:
   - Using grayscale image instead of binary image
   - Plotting intensity vs time and using it to determine the coordinates