Particle distribution calibration in two-phase flow using image processing based on sedimentation experiments

Research objectives

The ultimate objective of image processing is to accurately detect the edge of the sediment layer and understand how the layer grows over time by plotting the graph of thickness with respect to time. The evolution of a sediment layer will be used to calibrate the mean and standard deviation of the particle radii in the two-phase liquid-liquid flow at a given concentration.

The particle distribution analysis plays an important role in the fundamental understanding of two-phase channel flow and will be used in related research to model the flow in a two-phase single flow battery.

Image processing

Illustrated below is an original image extracted from a full video (1129^{th} frame from 'First Attempt Video'), it has already been cropped to a smaller one so that corresponding possessing is only performed on necessary sediment.

Figure.1 original image

First, to facilitate further analysis, the image should be converted to grayscale intensity, as is shown below.

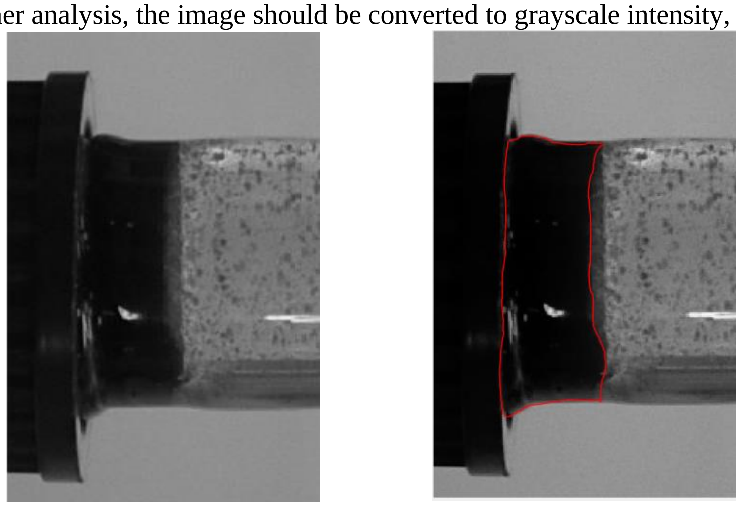

Fig.2 Grayscale Image(a&b)

The sediment enclosed by the light-red box is the edge we hope to identify. The fundamental strategy is to increase contrast between the two layers so that the boundary can be distinguished by built-in edge detecting function in MATLAB, and in the end measure the layer thickness. We notice that the boundary of the sediment isn't completely flat, so we'll measure its thickness in different positions, and treat the mean value as the calibrated thickness.

For most frames which are free of external disturbance such as the reflecting light, it's unnecessary to apply a filter before contrast enhancement because it only brings about detail losses in the boundary areas. However, for the frames which are susceptible to noises, it's essential to apply an appropriate filter so that the noises won't cause inaccuracy during the thickness measurement stage.

The image illustrated below (fig.3) is the detected edge at 1229th frame in the end if we perform identical analysis in the remaining steps except that we don't apply any filter from the beginning. It can be observed that except the edge we expect to be detected, there are many noises in the middle of the sediment, and this will cause imprecision during the thickness measurement stage. Thus, before increasing the contrast, an important step for this type of frame is to apply an appropriate filter to remove possible noises. It will be shown later how such noises are detected.

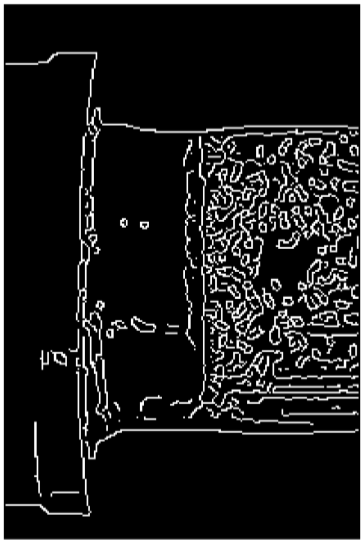

Fig.3 Detected edge at 1229th frame without any filter in first stage

The effect of a typical linear filter is subject to its order and domain(size). We'll first choose a proper order under the premise that a 6X6 filter is selected. Shown below is the edge detection comparison between different orders. 50% percent means each element in the 6X6 sorted set of neighbors is replaced by the pixel with the median pixel value, which is exactly the effect of a median filter. Similarly, 20% gives a effect close to the minimum filter, which tends to make the element in the current domain more dark, and 80% gives the opposite effect.

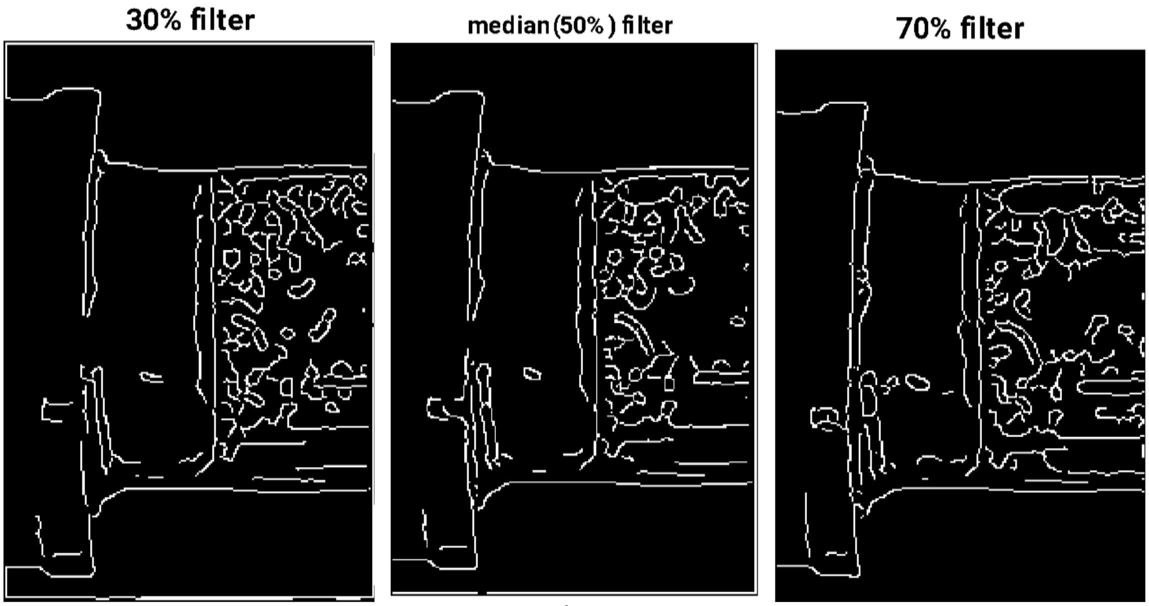

Fig.4 Detected edge at 1229th frame with different filer orders

Obviously, 70% filter is not appropriate since it even enhances part of the white noise in the sediment, giving counterproductive effects. Both median and 30% filters succeed to attenuate the noise, and 30% filter presents a better performance, but more edge is smoothed out as an offset. Since the 30% filter doesn't give a significant advantage in noise removal, and it tends to erase more boundary details, a median filter should be a proper choice of filter order.

Regarding the filter size, we've compared results from 3 different sizes. The 3X3 filter preserves most of the edge details, but it fails to remove the noises since the covering domain is overly limited, while the 9X9 filter appears to remove even more noises, but it also causes more deformation in the edge. Thus, we believe an expected edge-preserving filter is better implemented by a 6X6 median filter.

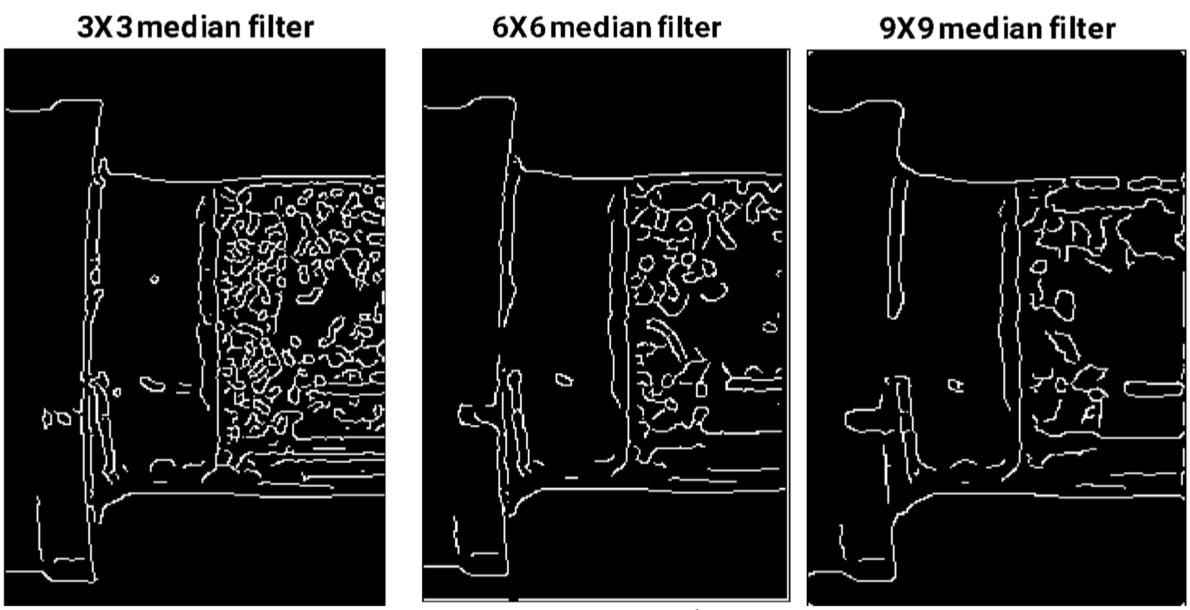

Fig.5 Detected edge at 1229th frame with different filter sizes

Then, we can use MATLAB function imadjust() to modify the contrast of the image to a desired state so that the boundary can be easily distinguished. Very similar result is derived from different saturation thresholds, thus it's safe to choose a 20% threshold which doesn't go to extremes.

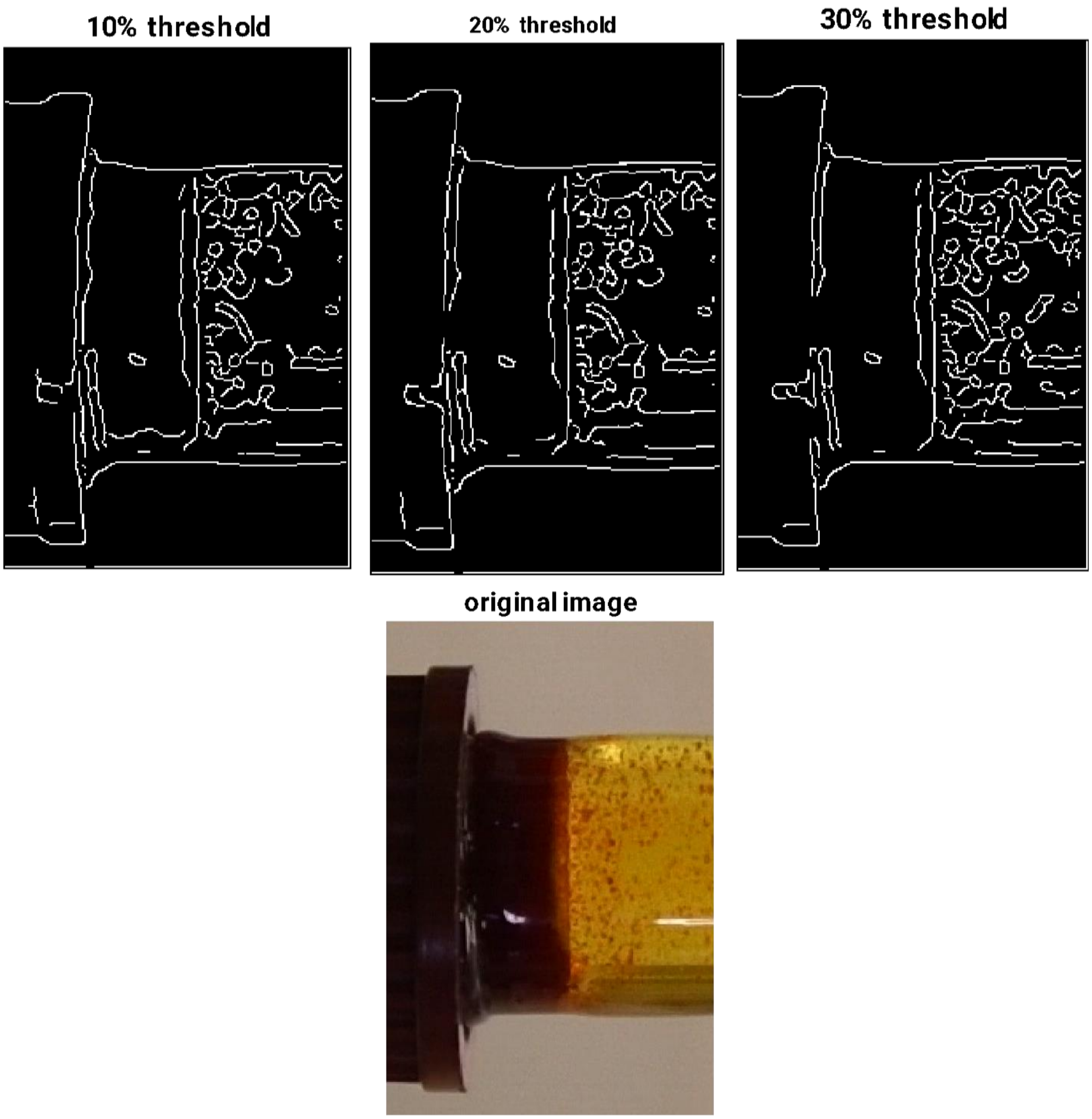

Figure.6 Comparison between different thresholds

Finally, the layer boundary can be detected with different methods provided by MATLAB edge() function, illustrated below is the final effect of each of them.

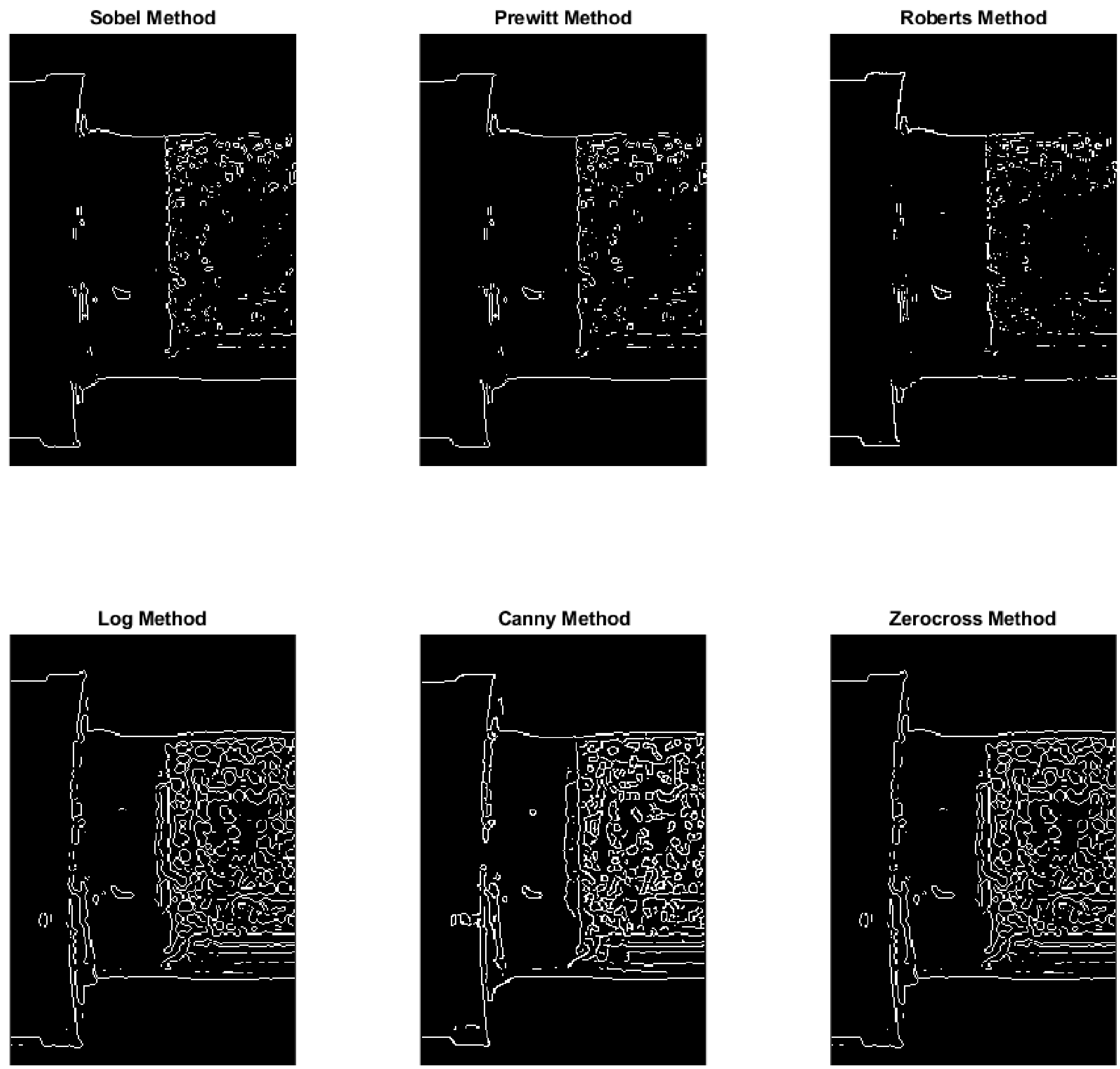

Figure.7 Edge detected by different method at 1229th frame

For Sobel, Prewitt and Robert method, wrong edges are detected, meaning they are not sensitive to pixel gradient change. For the remaining three methods, ‘Canny’ presents a more complete and continuous edge. Such a result is expected because ‘Canny’ uses a multi-stage algorithm to identify a wide range of edges and, compared to other methods, it is more effective and complex. Therefore, although ‘Canny’ takes more computation time, it is most recommended considering its performance.

llustrated below is the whole process we go through to find the edge at 1229th th frame. It should be emphasized that unless noises are detected in the image, no filter will be applied before we stretch the image so that minimal details are lost during the process. It will be shown how to decide the presence of noises in next section.

Also, it can be noticed that not all the noises are removed, because the filter size we apply is relatively small so that excessive edge deformation is averted. But since we will remove the outliers before computing the mean value, as long as the area occupied by the noise is small, it won't affect the final result.

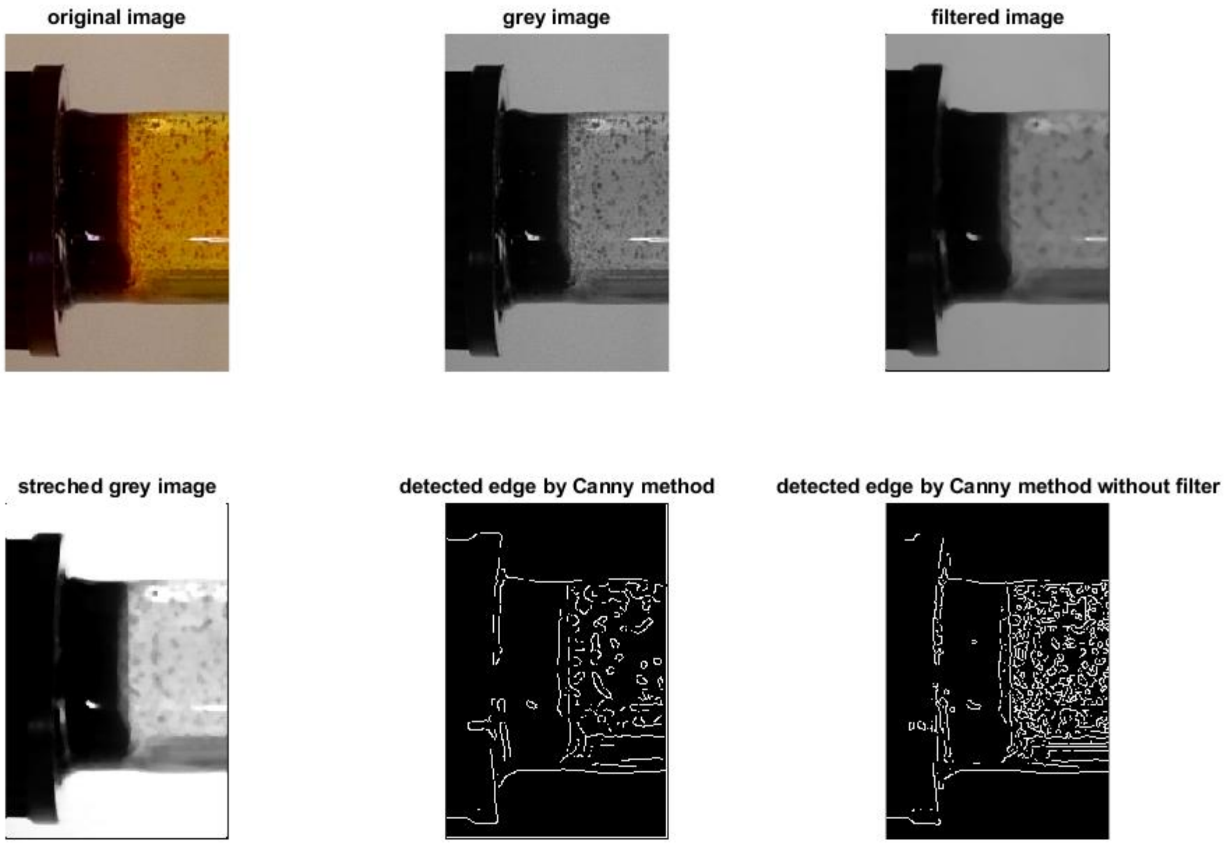

Figure.8 Whole process of edge detection of one single frame

Thickness measurement

To compute the thickness of the layer, we will first find the point which is around the mid-line of the layer. This step can either be finished by MATLAB or manually since we only need to choose the point for one time. Then, we'll compute the point's distance to boundary and bottom of the sediment.(The distance is considered as the length between the point and the first white pixel it meets in vertical direction) The lower part(blue) is only calculated for once, since Its value is theoretically fixed, and if we only care about the relative increase of boundary layer, there's no need to measure the lower point, because the first thickness we find can be treated as a reference value directly. Regarding the upper part(red), 10 sample distances covering a pixel length of 50 is calculated, and after removing the outlier, the mean value of the rest is considered as the layer thickness at the current frame.

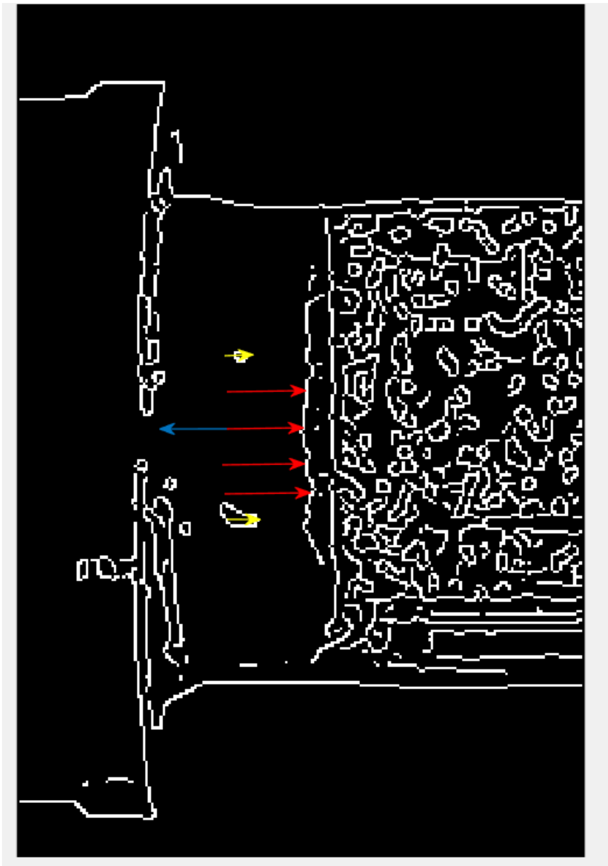

Figure.9 Illustration of thickness measurement

Noise Detection

The presence of noise is detected once the thickness of a sample point is found to be much smaller than the mean thickness of the previous frame. (The sample point marked by yellow arrow in figure.9, for example). In such a case, a 6X6 median filter will be applied, and another measurement will be performed. The threshold is set to be 8 pixel (1 mm), which is sensitive enough to most potential noises.

In addition, if the measured thickness of one single frame is abnormally higher or lower than the previous one ($\pm 0.5mm$), this data will be considered as noise, and the thickness of the current frame will be replaced by the thickness of the previous one.

Results

Repeating the whole process on each frame through the whole video, we get the following graph.
The data can be smoothed by Gaussian, mean and quadratic method, and the result appears quite similar, the illustrated graph is smoothed by Gaussian method. Other method such as 'median method' is not recommended, because they tend to give a linear regression or simply selecting the median, and this will yield a stepped increasement, thus deviate from the original data.

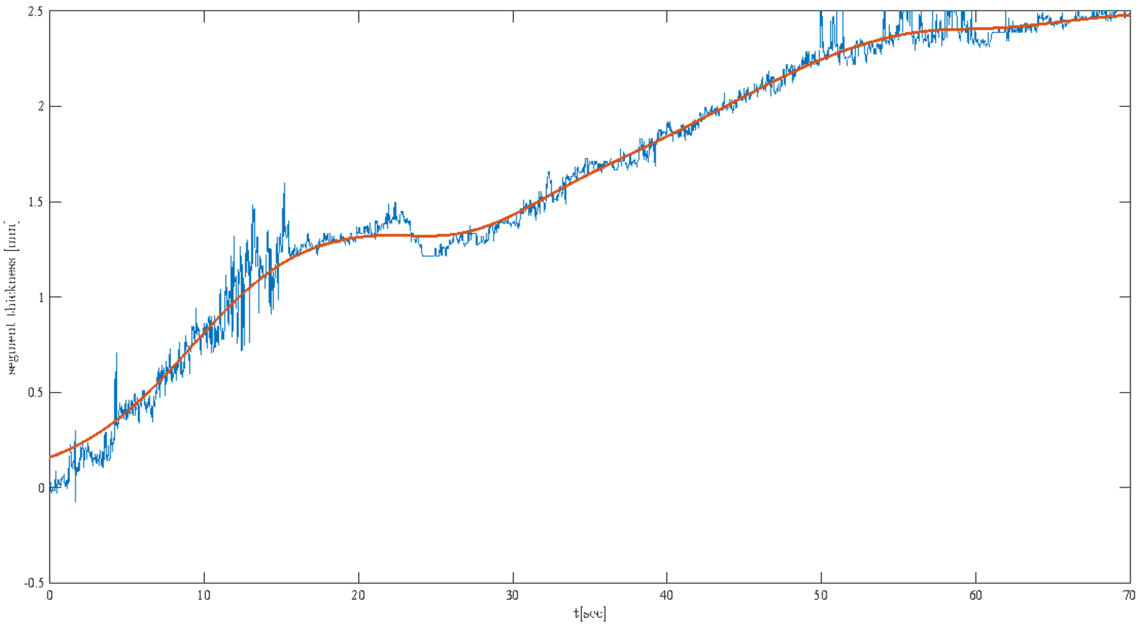

Figure 10. Sedimentation layer thickness vs time before and after adjustment