Implementation on Analyzing the Bubble Flows in Microchannels using Image Processing

Anjushree Nair, Neha Pandey, Kalpana R. Bodke, Mrs. Mohini Laghate

Abstract— **The paper aims at implementation of bubble flows in microchannel by an image analysis technique to measure the bubbles automatically by showing the radii of the bubbles and also to create the size distribution of the bubbles in horizontal and vertical columns of microchannel. The work which is involved here shows the measurement of diameter for drops found in bubble flows. Input image is taken from the user and then image enhancement technique is applied. By using Circular Hough Transform we can find the radii of different bubbles in different range. The drop size distribution graph is obtained as the output and also histogram is displayed in the spreadsheet. A GUI is made in MATLAB which is user friendly which automatically calculates the different radii of the bubbly flows using image processing.**

Index Terms: bubbly flows, Circular Hough Transforms, drop size distribution, GUI, image enhancement, MATLAB.

I. Introduction

This

This paper is application of image processing which is used in chemical industries. In many chemical industries the solvent extraction process is carried out so as to get the desired solute to transfer it from one liquid phase to other liquid phase. This process is done by mixing two immiscible liquids in mixer settler hence to create dispersion. The quality of dispersion is dependent on the speed of the bubble flows. This type of study is carried out in chemical industries of BARC. So it is required to measure the size and frequency of bubbles in the study of chemical flows. Size of the bubble is an important design parameter because it dictates the available interfacial area for Gas-liquid mass transfer. Bubble columns are widely used in many industrial applications, e.g. Fischer-Tropsch process for hydrocarbon synthesis, hydrogenation of unsaturated oil, coal liquefaction, fermentation and waste water treatment. For having a successful operation it is essential to quantify the bubble size distribution. This software is really great helpful to its user because it provide updating, adding and deleting of any module in software which is not available in the professional market software like ImageJ.

As the professional software does not allow any changes to be done on it hence it is not user friendly. The client wants something new that is suitable accordingly to their requirement so as to reduce the work load. As earlier client did not have any suitable software then calculating the bubbles one by one manually was a huge work load. Now this software which we are implementing solve almost all the basic problems which is faced by the users such as automated calculation of bubbles falling in which radii range, also manual counting option is also available such that the missed out bubbles can be encircled manually.

This software is made by knowing the client specification. So this makes current software more acceptable by its user's. In the current software the manual work which is needed to be done is reduce to its great extend and hence making it more popular among the user's. The user need to specify the radius range of the upper bound and lower bound and the software would automatically circles all the bubbles falling at this range. If manual counting is to be done of some bubbles which are left out then that can also be performed here. Also if some errors are occurred then that error correction facility is available here such as by using Reject module. All the required formulae's for the implementation of this project as per the user we are provided with the d3 formula as given to us by the user. If some more efficient algorithms come in the market then we may use that so as to make our software more efficient to user.

II. Review of Literature

In BARC's chemical department, earlier they used to use software called as ImageJ for analyzing the bubble flows in microchannels. ImageJ is a public domain, Java-based image processing program developed at the National Institutes of Health [1] [2]. ImageJ can display, edit, analyze, process,

Manuscript received March, 2015.

Anjushree Nair, *Computer Engineering Department, Mumbai University/ College AIKTC/. Mumbai, India, Mobile No +919757090731*

Neha Pandey, *Computer Engineering Department, Mumbai University/ College AIKTC/.Mumbai, India, Mobile No +919869835630*

Prof. Kalpana R. Bodke, *Computer Engineering Department, Mumbai University/ College AIKTC/., Mumbai, India.*

Mrs. Mohini Laghate, *Computer Engineering Department, Scientific Officer, BARC/, Mumbai, India.*

save, and print 8-bit color and grayscale, 16-bit integer and 32-bit floating point images.

In ImageJ, users had to manually count the number of bubbles, so as to measure the radii of the drops and the drop size distribution in horizontal and vertical columns. So, this used to take a lot of time and in order to reduce the time as well as to increase efficiency, they needed a software which needed less manual and more automated work to be done. So, they started using the software which we are going to implement in our project. The currently used software requires less manual work and is fully automated.

ImageJ is professional software. So, if any changes needed to be done the users could not do that as ImageJ was not customized according to the user's requirements. So, they changed their software in order to make it more customized. As the software which the scientists are using currently, was developed in BARC, so they were able to specify their requirements and the software was developed as per their need.

III. PROPOSED SYSTEM ARCHITECTURE

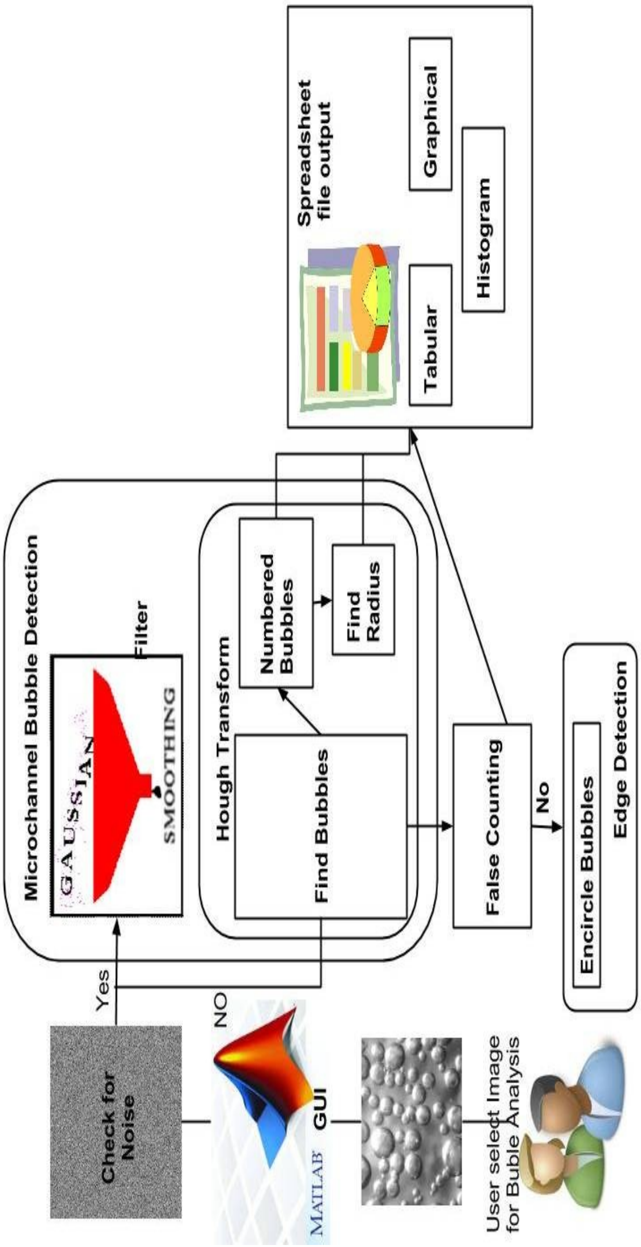


Fig.1. System architecture

First the user selects the image through the GUI designed in MATLAB and then checks whether noise is present in the image or not. If present, then Gaussian filter is applied so as to remove it from the image. After this, hough transform is applied on the filtered image to detect the bubbles in it. When the bubbles are detected, they are numbered starting from 1. After the numbering, the radius of the numbered bubbles is displayed in the list box. If any false counting of the bubble is encountered, then that can be rectified by using the manual counting option provided in our program. Finally, the output is displayed in excel sheets and also in the form of graphs and charts.

IV. DESCRIPTION OF THE METHOD

Given one of the photographic images of the dispersion, our approach, which is going to be, implemented in MATLAB/SCILAB [3], automatically identifies the contour of existing drops and classifies them according to their diameter. Below figures show the input images of the bubble flows and these are fed to the software to perform analysis.

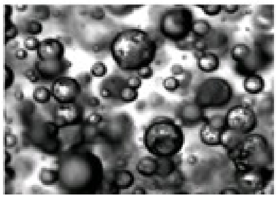


Fig. 2. Examples of images [6]

(a) is a very good image

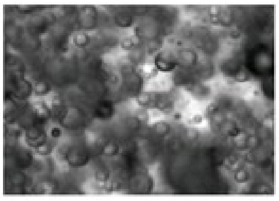


Fig. 2(b) is an image with low photographic quality

In our proposed approach, the process for the detection of the drops in an image has two distinct steps. In the first step, we detect the edges of the drops in the original image by monitoring the values of the gradient and the descending thickness and by creating an output image with those contours. In the second phase, we detect the drops in this contour image, using the Hough Transform. This transform is widely used in image processing to detect lines and also to detect circles.

First, the edges of the bubbles are detected. All the steps carried out in the detection process of the drops is shown in the below Figure 2. In the Figure 2 (a) the original image is displayed, the edges are detected as shown in Figure 2 (b) and in the next step; we detect the contours of the drops using the Hough transform as shown in Figure 2 (c).

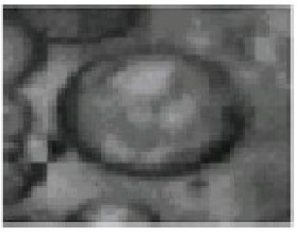


Fig.3. Steps in detection process [6]
(a) original image

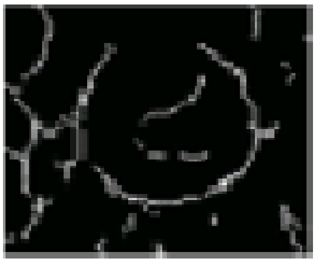


Fig. 3(b) the edge detection

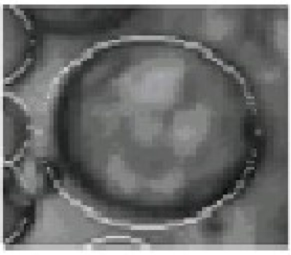


Fig. 3(c) The detection of drop using Hough transforms.

For the images which have very poor quality like the Figure 6 (a) image, we need to reduce the noise and consequently reduce the probability of false edges detection, the original image is smoothed using a Gaussian filter. This eliminates some false contours and reduces the detection of false drops. This process is shown in the below Figure:

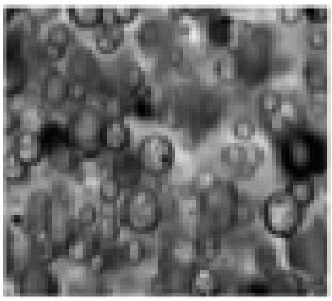


Fig. 4. Representation of drops [6]

Fig. 4 (a) the poor quality image

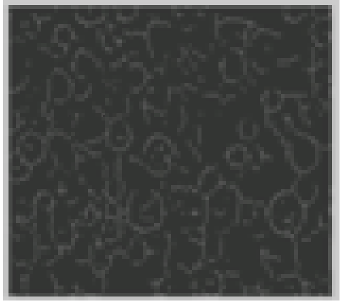


Fig.4. (b) the resulting contour image

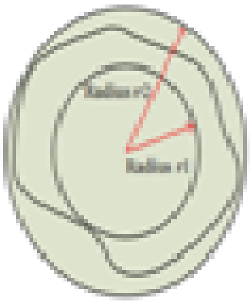


Fig. 4 (c) Representation of the drop contour as an irregular object

In the second step after edge detection, we apply the Hough transform to the contour image to detect the contour of the drops for different values of the radius. The detection process starts with the set of drops with radii between 17 and 48 pixels (0.17 and 0.48 millimeter, resp.) because these are the most frequent. Then, we treat the drops with at least 48 pixels of radius. Finally, we process the drops with radius between 8 and 16 pixels (0.08 and 0.16 millimeter, resp.)Finally we get the following result after performing Hough transform.

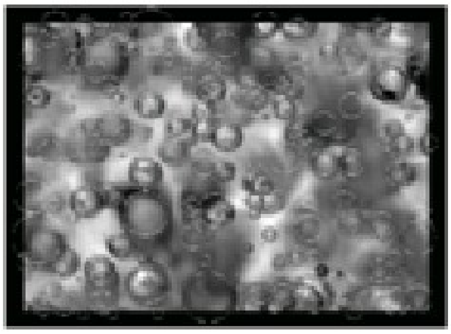


Fig.5. the drops detected by our program. Each detected drop is marked with a white circle on top of the original image [6]

After the detection process, if the user's find any error in the encircling of the bubbles, then they manually write the number which is displayed inside the bubble to a option provided in the GUI and reject it by clicking first on the "Highlight" option and then on the "Reject" option. If any bubble is left to be circled then by manual intervention, the bubble is encircled. Each bubble is assigned a number after Hough transform is performed. For the users of this software, it would be easier to understand the process, if the result is displayed in a graphical format. For this, an option "Write to excel file" is given which writes all the data regarding the radius of the bubbles, number of bubbles in that range of radius and the frequency of the bubbles. Finally, the process is completed and this process is repeated for the subsequent images.

V. IMPLEMENTATION

The project has been developed in MATLAB. So, the implementation part for the project is explained below:

- GUI implementation

The GUI that we have designed is as shown below:

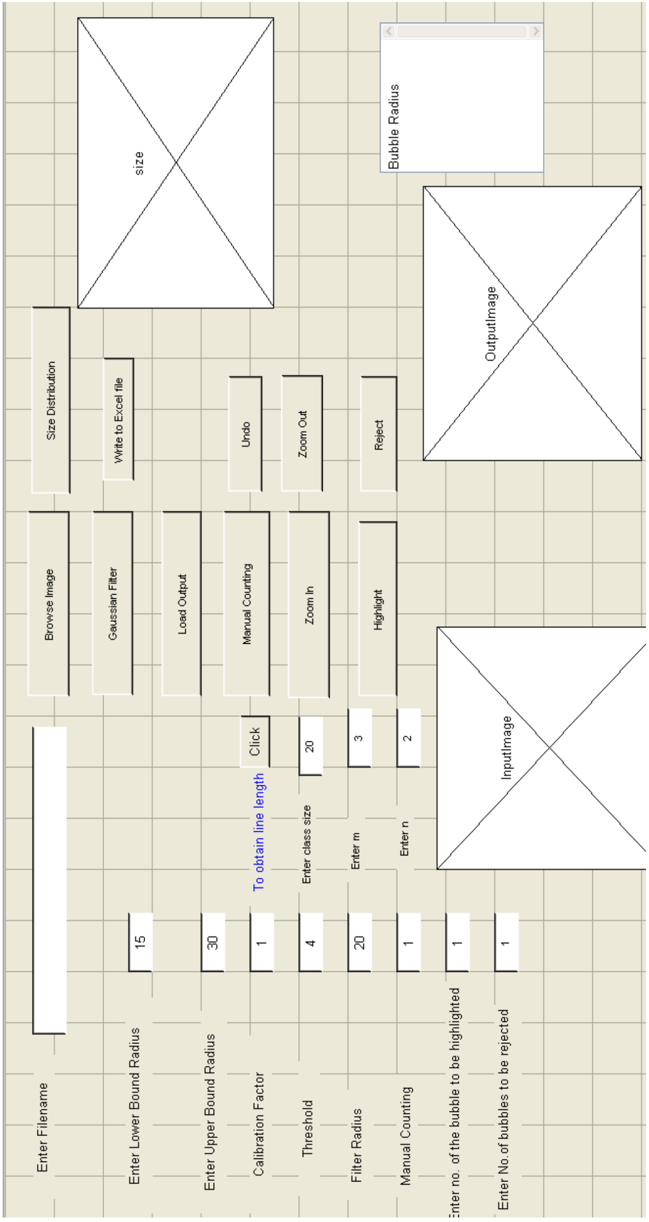


Fig. 6. Design of GUI interface

The various components used for making GUI are Pushbutton, list box, axes, edit text, static text.

- Browsing the image

In this, we have created a function which browses through the files in the computer and lets the user choose the required file for performing further operations.

So the code line so as to select the files form the computer is:

[Fname,Pname]=uigetfile('*.*','Enter Image File'); where, Fname and Pname stand for filename and pathname respectively.

- Applying Gaussian filtering

Gaussian filtering is applied to remove the noise and make the image smooth for better representation of the image. The Gaussian filtering methods effect that we have applied in our project is shown below:

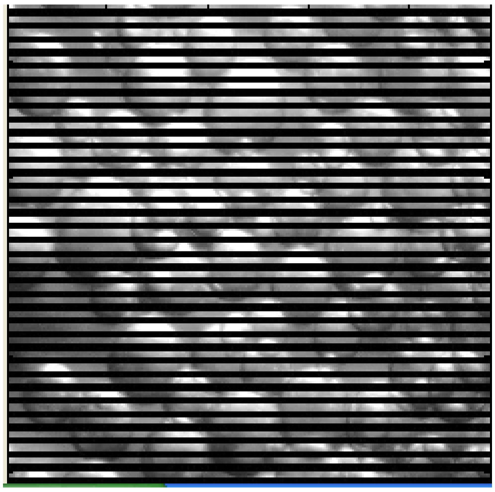


Fig.7. Poor quality image

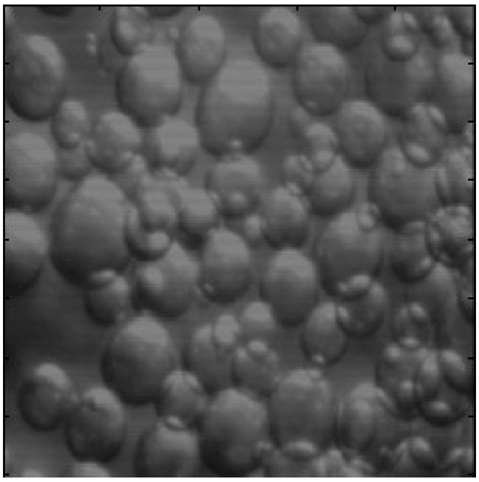


Fig.7 after applying the Gaussian filter

- Hough Transform

Hough transform is applied so as to mark the bubbles in the image. For this, we need to specify the upper and lower bound of radius. Other parameters that can be altered are class size, calibration factors, threshold and filter radius.

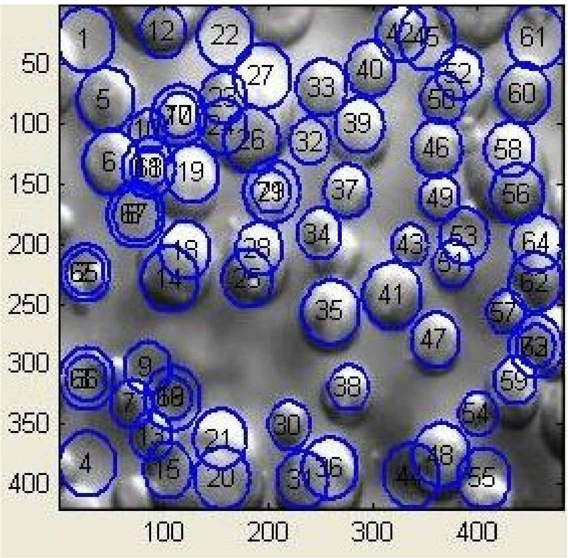


Fig.9. Hough transform output for the given image.

- Size distribution graph

The size distribution graph is as shown below:

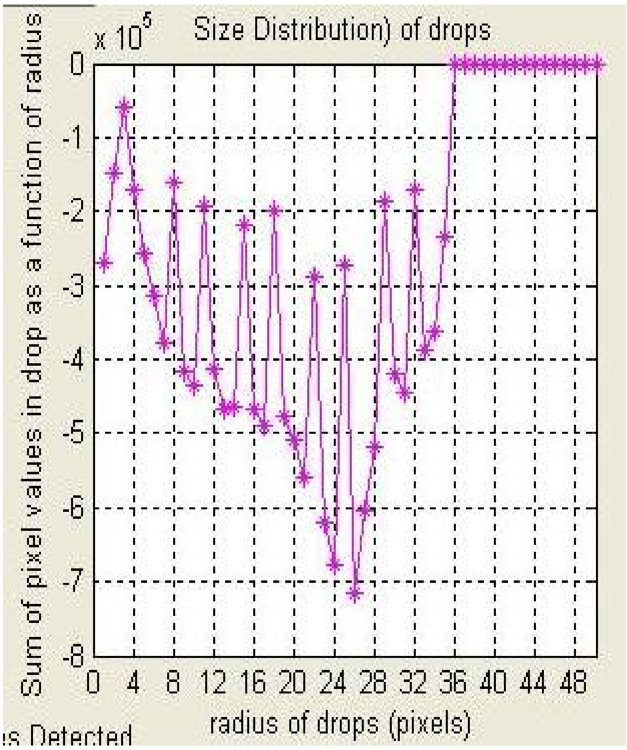
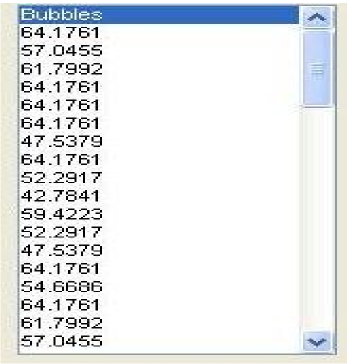


Fig. 10. Size distribution graph

- List box for displaying the radius

Here we have made a list box which gives the radius of the detected bubbles and it is as shown below:

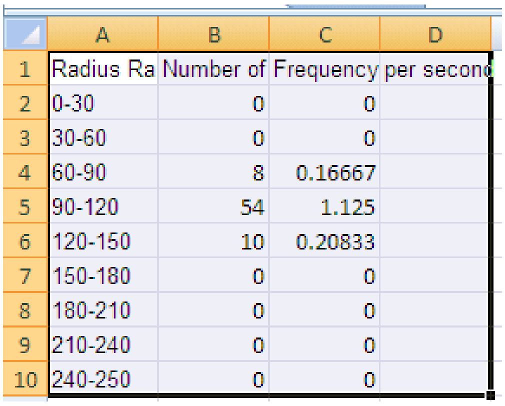


```
Bubbles
64.1761
57.0455
61.7992
64.1761
64.1761
64.1761
47.5379
64.1761
52.2917
42.7841
59.4223
52.2917
47.5379
64.1761
54.6686
64.1761
61.7992
57.0455
```

Fig.11. List box showing the bubble radius

VI. RESULTS

The result of our project, described in this paper is the radius of the detected bubbles. So these will be displayed in excel sheet with the radius range, number of bubbles in that range and the frequency of those bubbles as the constraint for differentiating the bubbles. The result will be also displayed in graphical and histogram format so that it will be easy for the users of this program to analyze them. The results have been shown below:



	A	B	C	D
1	Radius Ra	Number of	Frequency	per second
2	0-30	0	0	
3	30-60	0	0	
4	60-90	8	0.16667	
5	90-120	54	1.125	
6	120-150	10	0.20833	
7	150-180	0	0	
8	180-210	0	0	
9	210-240	0	0	
10	240-250	0	0	

Fig. 12. Expected outcomes in Excel file

(a) Table in excel sheet showing the range of bubbles along with their frequency etc.

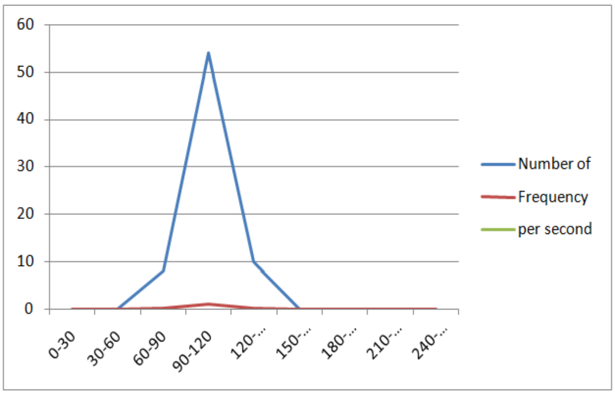


Fig. 12(b) Line chart for the above table

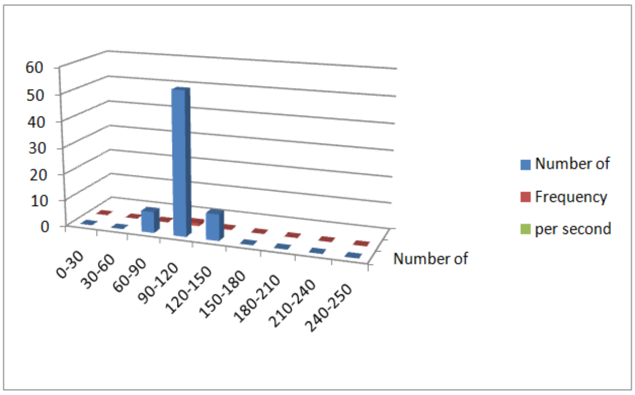


Fig. 12(c) Table chart for the table shown in Fig. 8 (a)

VII. APPLICATIONS

- Our project will be used in the chemical department of BARC for measuring diameter/length and frequency of bubbly/slug flows in lab scale bubble columns.

- Bubble columns are widely used in many industrial applications, e.g. Fischer- Tropsch process for hydrocarbon synthesis, hydrogenation of unsaturated oil, coal liquefaction, fermentation and waste water treatment. For its successful operation, it is essential to quantify the bubble size distribution.

- Bubble size is an important design parameter, since it dictates the available interfacial area for Gas-liquid mass transfer. For example, in blood oxygenators large bubbles favor CO2 removal, whereas small bubbles favor O2 transfer.

VIII. ADVANTAGES

The following are the advantages of our implementation:

- To evaluate our approach more efficiently we have chosen two images having different levels of photographic quality and we also compared the sets of drops obtained automatically with the drops which were identified manually in our project.
- A method is described for the automatic identification of drops in images taken from liquid-liquid dispersion.
- The results hence obtained gives 100 drops with diverse conditions such as radii, overlapping, border, quality etc gives the feedback that our program is able to detect a good percentage of bubble drops.
- This work which is started by us can be a promising start point for the possibility of performing an automatic drop classification with good results.
- In the case of a better quality images, as well as the noisy images the program recognizes more than 70% of the drops.

IX. DISADVANTAGES

- We have observed that the approach is less efficient for smaller values of the radius, since very small drops can be easily mistaken by noise.

X. CONCLUSION

In this paper, we have presented a method that would automatically identify the bubbles in the images that were taken during agitated liquid-liquid dispersion. The results that we have obtained after executing our program has lead to the conclusion that our program can recognize a good percentage of drops in better quality images as well as the noisy images.

We also provided the Gaussian filter method to clear the noisy images. We have implemented the manual counting method such that bubbles that we missed out could be encircled. By the implementation of highlight and rejection technique we are also able to modify our given image. Thus, overcoming all the disadvantages of the previous method, we have implemented new software for ease of working of the users.

XI. FUTURE WORK

In the future, we will be implementing this program using motion images as our input. Further, additional functionalities as per requirements of the user will be added which may include any other efficient algorithm if available than the one currently used.

ACKNOWLEDGMENT

We take this opportunity to express our deepest gratitude and appreciation to all those who have helped us directly or indirectly towards the successful completion of this Paper.

Foremost, we sincerely express our deep sense of gratitude to our guide **Prof. Kalpana R. Bodke** and **Mrs. Mohini Laghate** for their advice, constant support, encouragement and valuable suggestions throughout the Paper completion helped us successfully complete the Paper. Without their continuous support and interest, this Paper would not have been the same as presented here.

Besides our guide, we would like to thank entire teaching and non-teaching staff in the Department of Computer Engineering for all their help during our tenure at AIKTC.

We also take this opportunity to thank whole-heartedly Honorable Director **Dr. Abdul Razak Honnutagi** and our HOD **Prof. Tabrez Khan** who has imparted valuable teaching and guidance that has inspired us to attain new goals.

REFERENCES

Book:

[1] Schneider CA, Rasband WS, Eliceiri KW, NIH *Image to ImageJ*: 25 years of image analysis, Nature Methods 9 (7): 671–675, 2012

[2] Collins TJ, *ImageJ for microscopy, Bio Techniques* 43 (1 Suppl): 25–30, 2007

[3] Matlab R2006a, The Math Works, 2006.

[4] C. R. Gonzalez and R. E. Woods, *Digital Image Processing Using MATLAB*, Prentice Hall, Upper Saddle River, NJ, USA, 2004.

[5] J. C. Russ, *Image Processing Handbook*, CRC Press, Boca Raton, Fla, USA, 2006.

Journal Paper:

[6] Luis M. R. Bras, Elsa F. Gomes, Margarida M. M. Ribeiro, M. M. L. Guimaraes, *Drop Distribution Determination in Liquid-Liquid Dispersion by Image Processing,* Hindawi Publishing Corporation, 2009

AUTHORS PROFILE

Anjushree Nair

Pursuing her B.E degree from Mumbai University, in duration of four year from 2011 to 2015. Currently working on the project on Analysis of bubble flows in microchannels using image processing from Bhabha Atomic Research Centre [BARC] Mumbai.

Neha Pandey

Pursuing her B.E degree from Mumbai University, in duration of four year from 2011 to 2015. Currently working on the project on Analysis of bubble flows in microchannels using image processing from Bhabha Atomic Research Center [BARC] Mumbai.

Prof. Kalpana R. Bodke

Working as Assistant Professor, Department of Computer Engineering, at Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai, India. Having 6 years of Teaching Experience, Published 3 papers in International

Mrs. Mohini Laghate

Working as Scientific Officer-D,

Computer Department, BARC, Mumbai.