

BUBBLE TRAJECTORY TRACKING BASED ON ORB ALGORITHM

WANG SHUJUAN¹, LU SHICHAO², LIU JIAQI¹, WANG MIAO¹, LUO HONGLIANG¹, SHEN JIHONG¹, QIAO SHOUXU³, DAI YUNTAO^{✉, 1}

¹College of Mathematical Sciences, Harbin Engineering University, Harbin, China; ²The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China; ³College of Nuclear Science and Technology, Harbin Engineering University, Harbin, China

e-mail: wangshujuan@hrbeu.edu.cn; lushichao@hrbeu.edu.cn; liujiaqi97@hrbeu.edu.cn;

wangmiao98@hrbeu.edu.cn; 1139312852@hrbeu.edu.cn; shenjihong@hrbeu.edu.cn;

qiaoshouxu@hrbeu.edu.cn; peach0040@126.com

(Received September 3, 2022; revised December 20, 2022; accepted January 29, 2023)

ABSTRACT

The system of gas-liquid two-phase bubbly flows is widely found in many industrial fields, such as nuclear energy, chemical, petroleum, and refrigeration. Bubbly two-phase flows measuring including detection and tracking affects the specific engineering problem solving to a great extent. The particle tracking velocity (PTV) algorithm is generally used for the tracking of the particles in the flow field. However, it does not take the shape change of particles into account in the process of flow. In this paper, a kind of bubble feature matching method based on ORB algorithm is proposed, and the edge detection method of findContours in OpenCV is used to extract the bubble contour in the image. The proposed algorithm implements the trajectory tracking of the bubbles with shape change when moving up in liquid. The feasibility of bubble trajectory tracking is shown by displaying of different bubble tracks in the plan, 3D plots and contour changing plots.

Keywords: edge detection, feature matching, gas-liquid two-phase flow, tracking.

INTRODUCTION

Two-phase flow refers to the simultaneous existence of two different phases of the flow of substances, which usually can be divided into gas-liquid two-phase flow, gas-solid particles two-phase flow, liquid-solid particles two-phase flow (Igor *et al.*, 2020). The phenomenon of gas-liquid two-phase is commonly involved in ship, weapon, power and nuclear energy, which has an important influence on the cognition, prediction, operation, control and optimization of related physical phenomena. For example, in a heat exchanger, water and air are usually used as heat exchange media, while air is bubbled in water. And during the movement of underwater vehicles, super-empty bubbles are produced due to the high speed. The characteristics and dynamic behaviors of these bubbles, such as size, speed, trajectory, distribution, collision, rupture, have a significant effect on energy transfer, motion resistance, etc. As a common problem in many fields, the accurate grasp of bubble behavior in liquid directly determines the cognition of related physical phenomena (Augustyniak *et al.*, 2020).

In this paper, the trajectory of bubble movement is studied. At present, the trajectory tracking of bubbles

mainly uses the Particle Tracking Velocity (PTV) algorithm. It establishes the correspondence between identical particles in a sequence of exposure images, and tracks the motion of particles to determine their trajectories by identifying discrete particles in the image (Liu *et al.*, 2007). For the tracking of the bubble, the geometric center of the bubble is taken as the particle in the flow field (Sokoray-Varga B, Józsa J, 2008). However, it does not take into account the shape changes of the bubbles. (Kolaas *et al.*, 2015).

In the image, the edge of the bubble is a collection of pixels. In order to consider the change of bubble shape, we can perform feature point detection on each image and extract the edge information of the bubble. Corner detection algorithm is a feature point detection method based on grayscale images (Harris *et al.*, 1988). By calculating the first derivative of each pixel in the moving window, the pixel corresponding to the local maximum is found as the feature point. However, the algorithm has no scale invariance. When the scale of the moving window changes, it will cause the feature points to become edge points. Mikolajczyk K and Schmid C combine Harris-Laplace and Hessian-Laplace to create a robust, constant-scale, high-repeat detection operator

that compensate for Harris's lack of scale variability (Mikolajczyk *et al.*, 2004). Lowe first proposes Scale Invariant Feature Transformation (SIFT) algorithm, to handle the matching problem in the case of translation, rotation and affine transformation between two images. The method has good scale invariance and strong matching ability in feature point detection (Lowe, 1999). Lowe further perfects the SIFT algorithm, using the convolution of the original image and Gaussian core to establish scale space, and extract the characteristic points of scale-invariance on the Gaussian differential space pyramid (Lowe, 2004). The SIFT algorithm has a higher accuracy for matching, but its algorithm is slower. The Speeded-Up Robust Features (SURF) algorithm is a Robust local feature point detection and description algorithm, first presented by Herbert Bay *et al.* (Bay *et al.*, 2006). It is proposed on the basis of SIFT algorithm to make up for the shortage of SIFT algorithm in operation speed, but its detection accuracy can't be guaranteed. In 2011, the Oriented FAST and Rotated BRIEF(ORB) algorithm was proposed (Rublee *et al.*, 2011). It combines the FAST feature point detection method with the BRIEF feature descriptor and optimizes them on their original basis (Qin *et al.*, 2014). ORB is an order of magnitude faster than SURF, and over two orders faster than SIFT. It has been widely used in many fields in recent years, such as remote sensing image matching, target positioning and tracking (Zhang *et al.*, 2014; Xie *et al.*, 2022).

In this paper, we use intelligent labeling method to identify the contour and geometric center of bubbles. The feature points on the contour are extracted by ORB algorithm, then the feature points are matched. Finally, the bubble trajectory is tracked according to the geometric center. The proposed bubble matching method based on ORB algorithm takes into account the position information and the change of bubble shape by matching feature points. It implements the track of bubbles whose shape changes in liquid.

MATERIALS AND METHODS

DATA

In this paper, the data used are obtained by gas-liquid two-phase bubble flow experiment. Vertical rising two-phase flow is generated in a rectangular water tank with a container height of 60 cm and the liquid level height of 40 cm. The bubble flow video is recorded by a high-speed camera. The original bubble image set is derived from the bubbly flow video with a length of one minute and two seconds. We set FPS=30 (Frames Per Second), so the bubble image set consists of 1871 images, and the size of every image is 576×1024 pixels, as shown in Fig.1. Corresponding to the height of the tank

and the image pixels, the actual distance of a pixel is about 0.59 mm by simple conversion.

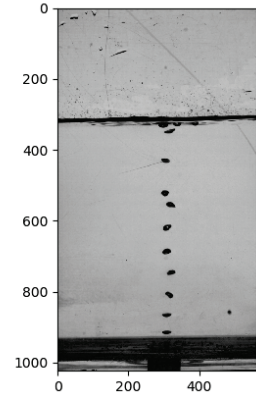


Fig. 1. *A original bubble image.*

EDGE DETECTION

Edge detection is a basic problem in computer vision, the purpose of edge detection is to identify the points with obvious brightness change in digital image. Significant changes in image attributes usually reflect important events and changes in attributes, including (I) discontinuities in depth. (II) discontinuities in surface direction. (III) changes in material properties, and (IV) changes in scene lighting. Edge detection is a research field in computer vision, especially in feature extraction (Gandhi *et al.*, 2020).

The most common method of edge detection is to detect the discontinuity of gray value by the first-order and second-order derivatives. It is to look for the location where the amplitude of the first derivative of gray scale is bigger than a specified threshold value or where the second derivative of gray scale has zero intersections. The edge detection algorithms based on the first-order derivative include Roberts operator, Sobel operator and so on, while the Canny operator (Ziou *et al.*, 1998) is based on the second-order derivative.

The Roberts operator is also called the cross-differentiation algorithm. Its principle is that the difference between adjacent pixels on the diagonal is regarded as the approximate solution of the gradient, and the edge lines are detected by local difference calculation. It is often used to process low-noise images with steepness, so it has good edge detection performance in both horizontal and vertical directions. The disadvantage is that the positioning of the edge is not very accurate, and the extracted edge lines are thicker.

Sobel operator is a commonly used edge detection algorithm. It is a discrete difference operator that uses difference approximation instead of gradient. The Sobel

operator detects the edge according to the gray-scale weighted difference between the upper and lower, left and right adjacent points of the pixel, and it is believed that the difference reaches the extreme value at the edge. It has a smoothing effect on noise and provides more accurate edge direction information. But the edge positioning accuracy is not high enough. When the accuracy requirements are not very high, it is a more commonly used edge detection method.

The Canny edge detection operator (Li *et al.*, 2022) is a multi-stage optimization operator with filtering, enhancement and detection. First, the Canny operator uses a Gaussian filter to smooth the image to remove noise. Second, the Canny operator computes the magnitude and direction of the gradient using finite differences of first-order partial derivatives. Third, non-maximum suppression of the gradient magnitude is performed according to the gradient direction angle. Fourth, the Canny operator uses a double-threshold algorithm to connect edges to remove isolated noise points. It is suitable for high-noise images, and has high localization accuracy, low false positives and can suppress false edges.

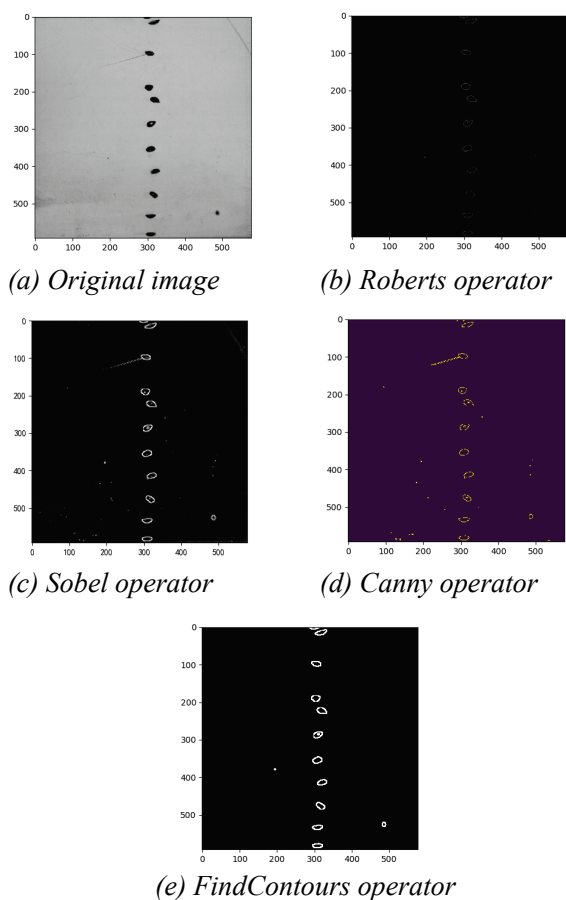


Fig. 2. The four operator extraction results are compared to the original image.

FindContours (Suzuki, 1985) is a contour detection function in OpenCV. It binarizes the image firstly, and

then searches the image from top to bottom, from left to right, to find the first pixel with non-zero value. The algorithm treats this point as the center, searching the next point which pixel value is not zero in counter clock-wise rotation, and constantly update the pixel value, to achieve the contour extraction of bubbles in the image.

Figure 2 shows the edge detection effect of the above algorithms on the bubble part. It can be seen that Roberts operator, Sobel operator and Canny operator indeed have effects on the edge extraction of bubbles, but there are still some problems. For example, the edges extracted by the Canny operator are the most complete, and the continuity of the edges is better. This is mainly due to the result of non-maximum suppression and morphological connection operations performed by the Canny operator. But compared to the findContours operator, the Canny operator easily identifies and classifies some small flaws as bubbles in the image (Fig. 2(d)). So, the findContours operator will be used to detect the bubble edge in this paper.

CONTOUR FEATURE MATCHING

The ORB algorithm is an algorithm for rapid feature point extraction and description, and consists of Oriented FAST and Rotated BRIEF algorithms. ORB algorithm is divided into two parts, namely feature point extraction and feature point description. Feature point extraction is developed from FAST algorithm, and feature point description is improved based on BRIEF algorithm.

The most outstanding feature of ORB algorithm is the high calculation speed of FAST. The calculation time is only about 1% of SIFT algorithm and 10% of SURF algorithm, mainly because FAST is used to accelerate the extraction of feature points (Rublee *et al.*, 2011). BRIEF algorithm is used to calculate the descriptor, which not only saves the storage space, but also greatly reduces the matching time.

In this paper, the algorithm is built up based on ORB algorithm, and the steps are as follows.

(1) Image graying.

(2) Extracting image feature points. The Oriented FAST algorithm is taken for the feature points extraction. Draw a circle with the pixel as the center and R as the radius. If there are consecutive N points on the circle and the difference between its pixel value and the pixel value of the circle center is greater than the threshold T , then the circle center is a feature point.

(3) Obtaining feature descriptor of feature points. The Rotated BRIEF algorithm is taken to obtain feature

descriptors. In the neighborhood of a feature point, select n pairs of pixel points (p_i, q_i) ($i=1, 2, \dots, n$). Then compare the gray values of these two points in each pair to generate a binary string of length n . $I(p_i) > I(q_i)$, the value of 1 is generated in binary string, otherwise 0 is generated. This binary string is a set of feature descriptors.

(4) Creating feature matcher to match the feature points between two adjacent images. For each feature point in the previous image, the best matching points in the next image is obtained according to the matching algorithm, then the successfully matched feature point pairs are recorded.

(5) Identifying the matched bubble in the next image for a given bubble in current image. For each feature point of a given bubble in current image, the matched feature point in the next image will be got by ORB algorithm. Then the matching rates of the given bubble and each bubble in the next image are calculated. That is the ratio of the number of successfully matched feature points of each bubble in the next image to all the feature points of the given bubble in current image. When the matching rate of two bubbles is greater than the set threshold, the bubble with the highest matching rate in the next image is selected as the matched bubble of that given bubble.

In step (5), the main factor affecting the success of bubble matching is the selection of bubble contour feature matching threshold. In order to improve the bubble matching effect, the remedial measure is implemented in this paper. If all the matching rates are smaller than the threshold, the bubble whose geometric center is closest to that of the given bubble is taken as the matched bubble in the next image. In the case that no bubble meeting the matching requirement, the bubble position information is applied to identify the matching bubble to ensure that each bubble in the current image can be

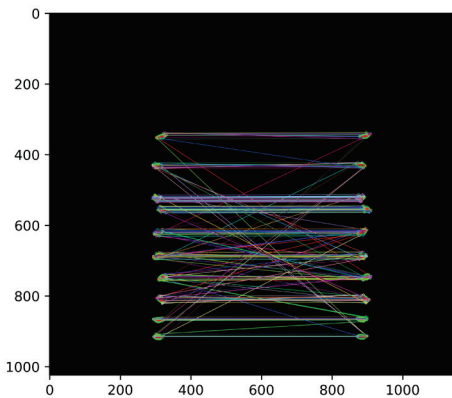


Fig. 3. *Bubble profile features matching results.*

matched to a bubble in the next image except for the bubbles that are about to disappear on the liquid surface.

The ORB algorithm performs the feature extraction and feature matching of bubbles in the image containing only bubble contour. The bubble matching effect between two adjacent images is shown in Fig.3, where the size of image is 1152×1024 pixels concatenated from two adjacent pictures. The left listed bubbles are the bubbles in the current image, and the right listed bubbles are in the next image, and each line connects the feature point pairs matched based on the ORB algorithm.

BUBBLE TRACK TRACKING

Based on the bubble matching rules stated above, a mechanism to identify and store the center point of a bubble during the process from generation to extinction is designed in this paper.

Assume that we start with the image i and image $i+1$, then place the geometric center coordinates of the same bubble in the list in the order of the image. The list format is as follows:

$$[[(x_{i,1}, y_{i,1}), (x_{i+1,1}, y_{i+1,1})], [(x_{i,2}, y_{i,2}), (x_{i+1,2}, y_{i+1,2})], \dots, [(x_{i,m}, y_{i,m}), (x_{i+1,m}, y_{i+1,m})]] \quad (1)$$

where, $(x_{i,k}, y_{i,k})$ and $(x_{i+1,k}, y_{i+1,k})$ represent the geometric center coordinates of bubble k in the image i and $i+1$ respectively, $k=1, 2, \dots, m$ and m represents the number of present bubbles currently in image i . So, each sub-list in the list (1) represents changing of the center point coordinates of a bubble. Similarly, for the image $i+1$ and image $i+2$, the bubble geometric center coordinates can be store in the following list:

$$[[(x_{i+1,1}, y_{i+1,1}), (x_{i+2,1}, y_{i+2,1})], [(x_{i+1,2}, y_{i+1,2}), (x_{i+2,2}, y_{i+2,2})], \dots, [(x_{i+1,n}, y_{i+1,n}), (x_{i+2,n}, y_{i+2,n})]] \quad (2)$$

where, n represents the number of the bubbles in image $i+1$.

Next, we traverse the sub-lists in (2) and (1), there will be two cases happen.

(a) If there is a sub-list in (1) with the second element of $(x_{i+1,q}, y_{i+1,q})$ and a sub-list in (2) with the first element of $(x_{i+1,p}, y_{i+1,p})$, and $(x_{i+1,q}, y_{i+1,q}) = (x_{i+1,p}, y_{i+1,p})$. Then, we update the sub-list in (1) with the following form

$$[[(x_{i,p}, y_{i,p}), (x_{i+1,p}, y_{i+1,p}), (x_{i+2,q}, y_{i+2,q})]].$$

(b) If there is no element in (1) that makes $(x_{i+1,q}, y_{i+1,q}) = (x_{i+1,p}, y_{i+1,p})$ valid, update list (1) into (3)

$$\begin{aligned}
 & [[(x_{i,1}, y_{i,1}), (x_{i+1,1}, y_{i+1,1})], \\
 & [(x_{i,2}, y_{i,2}), (x_{i+1,2}, y_{i+1,2})], \dots, \\
 & [(x_{i,m}, y_{i,m}), (x_{i+1,m}, y_{i+1,m})], \\
 & [(x_{i+1,q}, y_{i+1,q}), (x_{i+2,q}, y_{i+2,q})]]
 \end{aligned} \tag{3}$$

The list in (3) means that the number of bubbles recorded in the list is expanded by 1 than (1). We traverse the list from the image $i=1$ to the last image in sequence, so that we get the sequence of the geometric center coordinates of each bubble. Finally, the number of sub-lists in the list is the number of recorded bubble trajectory, and we could extract each sub-list to get a bubble center trajectory.

RESULTS

We define the horizontal and vertical geometric coordinates of the bubble trajectory in terms of the arrangement of pixel positions, which the positive direction of the y-axis is downward, and the positive direction of the x-axis is rightward, then draw the trajectory of the bubbles.

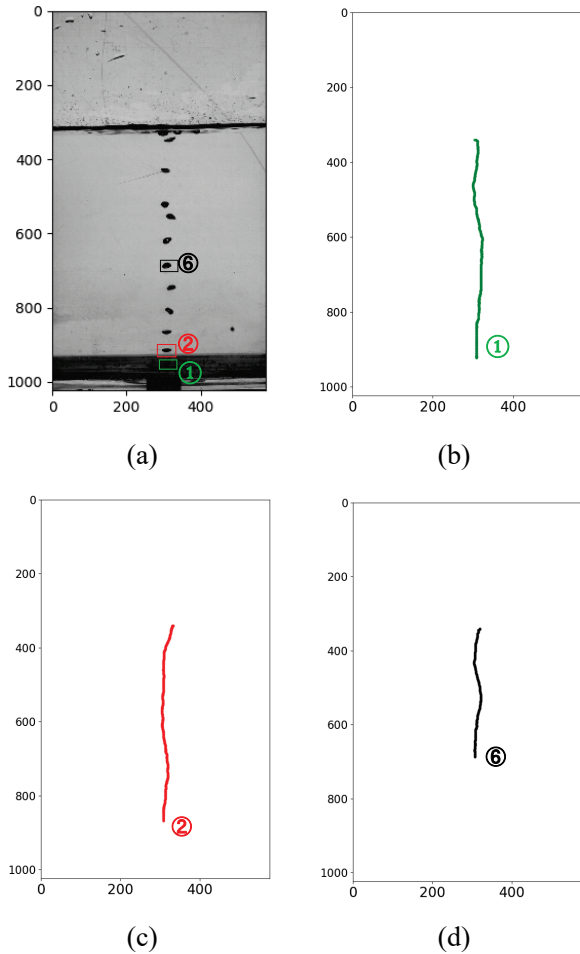


Fig. 4. The original image and parts of the bubble trajectory map.

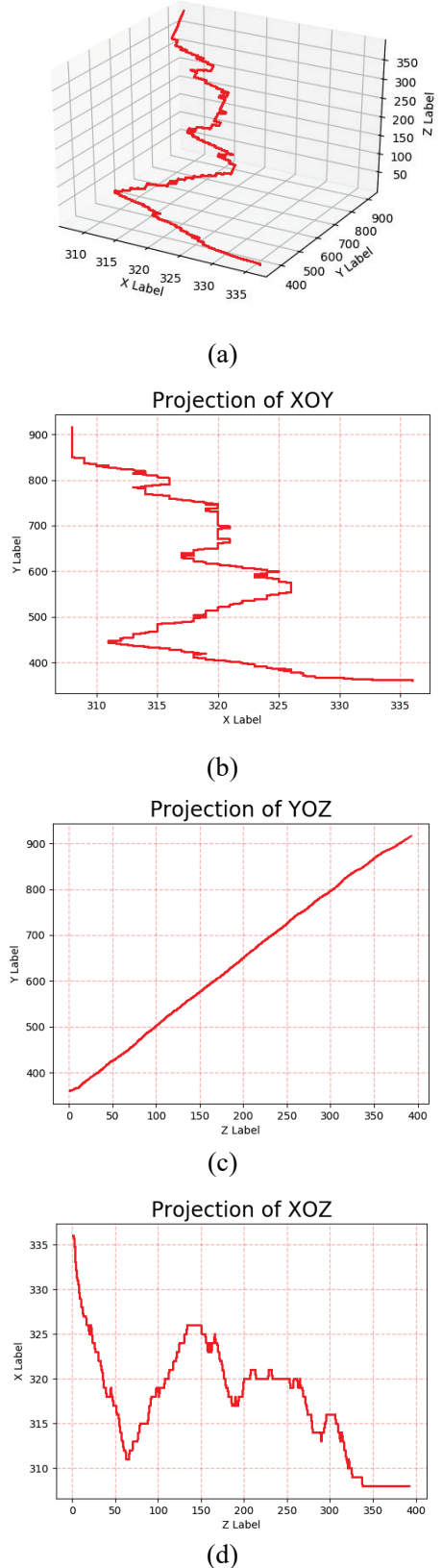


Fig. 5. The 3D trajectory of a bubble and its three views.

Fig. 4(a) shows the first frame bubble image in our data, in which the bubbles framed by different colors will be taken as the targets and their trajectory maps will be dis-

play. We use the same color as the frame to show the trajectory of the bubble. Fig. 4(b) shows the trajectory of blurred bubble framed by green rectangle in the bottom of Fig. 4(a), and the bubble is obscured by the experimental device. Fig.4(c) shows the trajectory of the second bubble framed by red rectangle in Fig. 4(a). Fig. 4(d) shows the trajectory of the sixth bubble framed by black rectangle in Fig. 4(a). These three trajectories start from different position (height), while end at almost the same position (height). Because the bubbles first appear at different height in Fig. 4(a) and finally reach the same liquid layer.

In order to better observe and understand the trajectory changes of bubbles during the ascent, we draw the 3D trajectory map of the bubble framed by red rectangle in Fig. 4(a) with time as z-axis in Fig. 5(a), and the three views of the 3D map are drawn from the top view (Fig. 5(b)), the left view (Fig.5(c)) and the positive view (Fig. 5(d)).

Furthermore, in order to observe the shape changing of the bubble during the ascent, a bubble contour trajectory is shown in Fig.6. The bubble tracked in Fig.6 is the bubble framed by red rectangle in Fig. 4(a). This bubble appears from the first frame image to the 378-th frame image in the original bubble image set, which means that this bubble takes 12.6 seconds from generation to disappearance. Corresponding to the height of the liquid layer and the original position of the bubble, it travelled 34cm, so the average speed of it is about 27.0mm per second. All the matched bubbles are superimposed on one image and other unnecessary bubble information is eliminated. To fully demonstrate the bubble deformation process, we display a bubble profile every 13 frames (0.43 seconds).

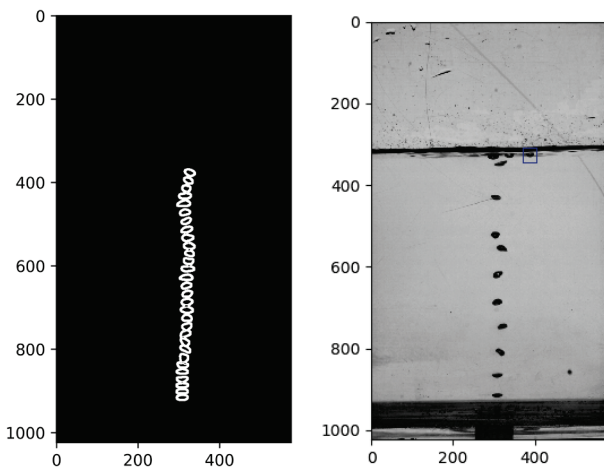


Fig. 6. *Bubble shape change image*

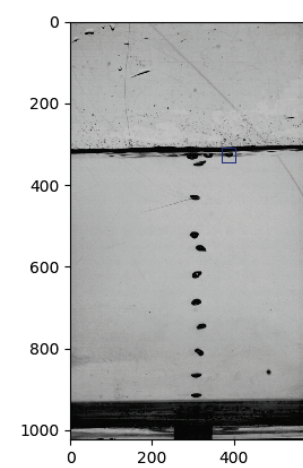


Fig. 7. *Reappearing bubble close to the liquid layer*

In the experiments, it is found that a new short bubble trajectory may occur after the bubbles have reached the liquid layer. That is because the bubbles did not immediately break when they reached the liquid layer, then re-entering the identification area by moving randomly (see the bubbles framed by blue rectangle in Fig.7.) This will affect the feature point matching of bubbles in the subsequent trajectory recognition process, and cause the phenomenon of mismatch occurs. To solve this problem, we can filter out unreasonable trajectories based on the sequence length of subsequent bubble matching results to obtain a reasonable bubble motion trajectory.

CONCLUSION

In this paper, an ORB feature matching algorithm is applied to track bubbles in liquid. We get a clear sequence of bubble motion images through high-speed photography. According to the features of the image, we preprocess the bubble images firstly, and we perform threshold segmentation on the image, then detect the bubble size and bubble geometric center in the two-phase flow image. Finally, we use the ORB feature matching algorithm to track the movement of multiple bubbles, which provides a basis for studying the characteristics and dynamic behavior of bubbles in liquid. Practice has proved that the correct measurement results of this method is a feasible multi-phase flow detection method.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central universities with number 3072022CFJ2403, 3072022JC2401 and 3072020CFT0104.

REFERENCES

- Augustyniak J, Perkowski D, Mosdorf R (2020). Measurement of multifractal character of bubble paths using image analysis. *Int Commun Heat Mass Transfer*: 117: 104701.
- Bay H, Tuytelaars T, Gool L V(2006). SURF: Speeded up robust features. *Lect Notes Comput Sci* 51: 404-17.
- Gandhi M, Kamdar J, Shah M (2020). Preprocessing of non-symmetrical images for edge detection. *Augment Hum Res* 5: 1-10.
- Harris C, Stephens M (1988). A combined corner and edge detector. In C. J. Taylor, editors, *Proceedings of the Alvey Vision Conference*, Alvey Vision Club: 23.1-23.6.
- Igor P, Mikhail P, Konstantin S (2020). Bubble patterns recognition using neural networks: Application to the

- analysis of a two-phase bubbly jet. *Int J Multiphase Flow* 126: 103194.
- Kolaas J, Drazen D, Jensen A (2015). Lagrangian measurements of two-phase pipe flow using combined PIV/PTV. *J Disper Sci Technol* 36: 1473-82.
- Lowe DG (1999). Object recognition from local scale-invariant features. In: *Proceedings of the seventh IEEE international conference on computer vision*. 2: 1150-1157.
- Lowe DG (2004). Distinctive image features from scale-invariant keypoints. *Int J Comput Vision* 60: 91-110.
- Liu C, Tao L(2007). Two-dimensional digital particle tracking velocimetry algorithm based on the image of particle trace. *J Coast Res*: 415-19.
- Li Y, Liu B (2022). Improved edge detection algorithm for canny operator. *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2022*: 1-5.
- Mikolajczyk K, Schmid C(2004) . Scale & Affine Invariant Interest Point Detectors. *Int J Comput Vision* 2004:63-86.
- Qin Y, Xu H, Chen H (2014). Image feature points matching via improved ORB. *2014 IEEE International Conference on Progress in Informatics and Computing (PIC)*. IEEE, 2014: 204-8.
- Rublee E, Rabaud V, Konolige K, et al (2011). ORB: An efficient alternative to SIFT algorithm or SURF algorithm. *2011 International conference on computer vision, ICCV 2011*: 2564-2571.
- Suzuki S (1985). Topological structural analysis of digitized binary images by border following. *Comput Vision Graph* 30: 32-46.
- Sokoray-Varga B, Józsa J(2008). Particle tracking velocimetry (PTV) and its application to analyse free surface flows in laboratory scale models. *Per. Pol. Civil Eng.* 52: 63-71.
- Ziou D, Tabbone S (1998). Edge detection techniques-an overview. *Int J Pattern Recognit Artif Intell* 8: 537-559.
- Zhang Y, Miao Z (2014). Object recognition based on ORB and self-adaptive kernel clustering algorithm. *2014 12th International Conference on Signal Processing (ICSP)*. IEEE, 2014:1397-1402.
- Xie YG, Wang Q, Chang, YX, Zhang, XY (2022). Fast Target Recognition Based on Improved ORB Feature. *Appl. Sci* 2022, 12: 786.