Available online at www.sciencedirect.com
SCIENCE DIRECT®
International Journal of Thermal Sciences 45 (2006) 321–330
International Journal of Thermal Sciences
www.elsevier.com/locate/ijts

Template-based bubble identification and tracking in image sequences

Da-chuan Cheng, Hans Burkhardt *

Institute of Pattern Recognition and Image Processing, Department of Computer Science, Albert-Ludwigs-University, 79110, Freiburg, Germany

Received 17 October 2003; received in revised form 5 August 2004; accepted 5 August 2004

Available online 6 October 2005

Abstract

We have developed a template-based system for identifying and tracking vapour bubbles. The bubble images are acquired from a high-speed digital camera. Each image sequence is recorded digitally for off-line image processing. The bubble detection process is organized in two phases: bubble identification and bubble tracking. In the first phase, a principal component analysis is applied for detecting bubbles roughly. After the detection, some bubbles' sub-images are used as templates in order to identify the bubble contours accurately. The prerequisite of using this system is that the bubbles are assumed to be of elliptic shape. In the second phase, the templates are used again for tracking bubbles. The modified Newton–Raphson algorithm is applied to reduce the search time. The advantage of this method is that it can recognize bubbles that are recorded in a wide-range of experimental environments such as varied illumination, different bubble appearances and so on. After the bubble identification, some parameters such as bubble size, position, moving speed, and the bubble nucleation sites are obtained. They are important for the research on heat transfer.

Keywords: Bubble identification; Tracking; Newton–Raphson method

1. Introduction

The goal of our cooperative research group [1,2] is to explore the heat transfer relationship between given liquids and heated tubes with different size, materials, surface structure and so on. The eventual goal is to discover the optimal heat transfer construction, which can save energy and prevent some possible critical situations. For this purpose, many experiments are designed and carried out. Some experimental processes are recorded with a high-speed digital video camera. These images are saved digitally for off-line processing. Each sequence contains five hundred images and each image has a resolution of 512 by 512 pixels. The sampling rate is 1000 frames per second. These experiments are made under different testing conditions such as liquid categories, tube diameters, surface material, surface structure, pressure, temperature, and heat flux. These factors have influences on image qualities. For example, an emery ground surface type has much more light reflection than the sandblasted surface type has. See [3, Fig. 1].

The goal of this research is to extract the bubbles' geometry such as bubble size, bubble position, and bubble moving speed. In order to reach this goal, every bubble in the image sequence should be recognized and identified. In one of our previous study [4], we used region-based snakes to identify the bubble contour. This method can be applied even if the bubble is in irregular shape if an appropriate initial contour is given near the detected target. This scheme maximizes the difference of the inner and outer region defined by the contour. If we want to obtain an accurate contour, the selected feature has to be able to identify the contour from a noisy environment (such as shadows or light reflections). However, in our images, some bubbles' appearances are very similar to background both in gray-level and in texture. It is non-trivial to find features which can distinguish the bubble contour precisely. This method works well when the bubble appearances are quite different to background and without shadows and light reflections. Consequently, its application is very limited in our research.

Another previous study [5] shows that we are able to identify bubbles in a circular shape in a limited experimental en-

* Corresponding author.
E-mail addresses: cheng@informatik.uni-freiburg.de (D. Cheng), burkhardt@informatik.uni-freiburg.de (H. Burkhardt).
URL: http://www.informatik.uni-freiburg.de/index.html.

1290-0729/$ – see front matter
doi:10.1016/j.ijthermalsci.2004.08.008

vironment. The limited experimental environment means the experimental conditions generating bubbles that have circular shapes. Generally, this condition should have an intermediate heat flux under high pressure. This is done with the help of a database. The database is a set of sub-images that are selected manually. In this database, some sub-images are bubble templates and some are background templates. Each sub-image has the same size dependent of the selected largest bubbles. Each bubble template contains a bubble in the center. The bubbles are at first roughly detected by a spectrum analysis method. Then, the database is applied to locate the bubble center and the bubble radius can be identified by the cross-correlation method compared to the bubble template in the database. After the identification, the bubbles are tracked and its nucleation sites can be then detected.

In our recent work [3], we are able to track bubbles by giving a bubble contour manually. This is made by using the local contour information and a circular model. A cost function is defined, in which not only the gray-level information is used, but also some gradient information and gradient direction. In addition, the occlusion problem is solved. Refs. [3,5] are based on an assumption that the bubble has a circular shape. However, many bubbles do not follow this assumption such as large bubbles or bubbles that are generated by lower pressure and higher heat flux. Some bubbles are of elliptic or irregular shape, especially when two or three bubbles are merging together. The underlying scheme is developed for observing bubbles generated from nucleation sites.

A similar study for tracking bubbles is reported in [6]. The author developed a PTV (*Particle Tracking Velocimetry*) system based on image processing techniques. The bubble flow is limited to a column and a back light source with a highly divergent light that is applied. With this construction, some bubbles can be focused by adjusting the depth. Only bubbles in focus appear in the images and the other bubbles are blurred. These blurred bubbles can be rejected by a subsequent image segmentation technique. The segmentation technique used in this study is a thresholding process combined with some edge detection method and morphology operations. It is based on the fact that:

(1) The bubble images have homogeneous background.
(2) The bubble appearance has less difference.

This kind of images is relatively easy to be processed in order to identify the bubbles. In the case of our images, the background is normally the surface of the heated tube. Therefore, it is not homogeneous because it depends on the surface structure. The simple thresholding technique cannot be applied to segment bubbles from the background.

Another study using PTV to track particles in water is reported in [7]. In the images, the background is dark and homogeneous, the particles are white so that a thresholding technique and region growing process can be used to segment the particles from the background.

The problems encountered in our research can be divided into two major groups, i.e. the geometric problem and the optical problem [3]. In the geometric problem, it contains three difficulties:

G-1 The bubble shape depends on the surface structure, pressure, and given heat flux.
G-2 Two or more bubbles can be merged together to form a larger bubble. In the coalescence process, the shape of bubbles is irregular.
G-3 One bubble can be visually occluded by another one.

In the optical problem, it contains five difficulties:

O-1 The reflection on the bubble surface depends on the position of the bubble.
O-2 The reflection on the background depends on the bubble's position. (See [3, Fig. 1a].)
O-3 Large or flowing bubbles have a strong shadow problem. This is because they have distinct distance to the tube (background). Moreover, the shadow position depends on the bubble position.
O-4 Some bubbles are in ill-illuminated areas.
O-5 Some bubbles are out of focus. This might be due to two possibilities:
  (1) It is a speedy flowing bubble.
  (2) It is not in the focus depth.

Case G-1, as we have mentioned before, the bubbles in intermediate heat flux and high pressure are mostly circular. The bubbles in high heat flux and low pressure are elliptic or even irregular. The difficulties of G-2 and G-3 are sometimes non-trivial to distinguish. The only information is time. If bubble A is visually occluded by bubble B and if their moving speeds are different, then after several frames, bubble A can be observed again. However, this is under the assumption that both bubbles are in focus. Normally, a flowing bubble is out of focus and this will increase the difficulty of recognition. In many cases, we observe that two flowing bubbles are out of focus and one is occluded by another one. In our previous study [3], we solved the occlusion problem but under the assumption that the target bubble is of circular shape.

The difficulties O-1, O-2, and O-3 hint that the single cross-correlation method cannot be applied. All of the above possibilities make bubbles quite different in geometry and in appearance and non-trivial to be identified. This is due to the fact that every image sequence has different combinations of difficulties.

Unlike our works in [3,5], we expend the model from circular to elliptic in order to better identify more bubbles under different experimental conditions. In addition, the bubble recognition and feature selection in [5] are modified in this paper which can recognize bubbles in a better performance.

In Section 2, the system architecture of the whole process is introduced. The algorithm is divided into three major parts:

(1) Bubble identification (Section 2.1);
(2) Bubble tracking (Section 2.2);
(3) Manual correction (Section 2.3).

Afterwards, the results are given and we end with a conclusion.

2. System architecture

The experimental facility is shown in [2, Fig. 4.1]. The investigated heating tube is fixed in the center of the container. Through a window one can observe and record the experimental process. Some recorded images with different experimental conditions are shown in [2, Fig. 5.4] or in [8, Fig. 1].

Normally, each tube is recorded at least by three image sequences under the same experimental condition. They are at different window positions shown in Fig. 1. Generally, the upper window contains not only the bubbles generating from the upper surface of the tube but also from the middle and lower part. Those bubbles are normally flowing and out of focus. Therefore, they might interfere the observed target bubbles. The bubbles in the middle window are usually in focus, this is due to the fact that the distance between the tube surface and the video camera do not change in a wide range and most of the target bubbles are in the same depth. On the contrary, the bubbles in the upper or lower windows are partially not in focus.

Each image sequence contains 500 images and they are recorded in 0.5 sec. A GUI (**G**raphic **U**ser **I**nterface) is developed in a Matlab [9] platform for user to operate the bubble recognition system. This system is generally divided into five parts:

- Database setup.
- Bubble recognition.
- Bubble contour identification.
- Bubble tracking.
- Manual correction.

There are two kinds of databases: one is for bubble recognition and another one is for bubble contour identification. In this paper, the phrase 'recognition' means that some target is acknowledged on an image. However, the boundary of the target might not be accurately acknowledged. Nevertheless, the phrase 'identification' means that the target is accurately acknowledged.

In the bubble recognition, bubbles in the images are roughly recognized. It gives the system information where the bubbles are. Thereafter, the system concentrates on these areas to identify the bubbles' accurate positions and contours.

In the bubble tracking, the system uses information not only from the results of recognition but also from the results of the bubble identification in the last image. Notably, we track bubbles in a backward direction. In the following sections, the details of the system will be outlined.

2.1. Bubble identification

For the intention to deal with the first four optical problems, it is intuitive that making a system to learn the target (bubbles) is necessary [10]. The reason is obvious: Since the bubble appearances vary according to the experimental conditions and parameters, it is reasonable to have template information of the bubble appearances from the processed images.

Feature extraction. A commonly used feature extraction method is the Karhunen–Loève transformation (or Principal Component Analysis) [11,12]. In this study, we use a database to collect some manually selected sub-images as patterns. These patterns have the same size of 40×40 pixels. They include bubble and background patterns. The user should choose as many patterns as possible to cover varied bubble or background appearances, and as less as possible to decrease computation time and inconvenience. How many patterns should be selected is a black art. Normally, we select patterns from the first 50 images. The number of patterns should be at least 20 for each group: bubble and background.

Let $\mathbf{B}_i$ denote the ith bubble pattern and $\mathbf{BG}_j$ denote the jth background pattern, $\mathbf{B}_i, \mathbf{BG}_j \in R^2$. The expectation of every pattern is zero, for both bubble or background patterns. The kth column in $\mathbf{B}_i$ is denoted by $\mathbf{B}_{i,k} \in R$, $\mathbf{B}_{i,k}$ is a vector and $1 \leqslant k \leqslant 40$. Assuming there are N bubble patterns and M background patterns, the composition of all kth column from every pattern is denoted by $X_{B,BG,k}$, where $X_{B,BG,k} = \{\mathbf{B}_{1,k}, \mathbf{B}_{2,k}, \ldots, \mathbf{B}_{N,k}, \mathbf{BG}_{1,k}, \mathbf{BG}_{2,k}, \ldots, \mathbf{BG}_{M,k}\} \in R^2$ is a $40 \times (N + M)$ matrix. Its correlation matrix is $R_{X,k} = X_{B,BG,k} X_{B,BG,k}^{\mathrm{T}}$. Thereafter, the eigenvectors of the correlation matrix can be obtained and denoted as $\boldsymbol{\nu}_{1,k}, \boldsymbol{\nu}_{2,k}, \boldsymbol{\nu}_{3,k}, \ldots$. Assuming the largest four eigenvalues are $\lambda_{1,k}$, $\lambda_{2,k}$, $\lambda_{3,k}$ and $\lambda_{4,k}$, and their corresponding eigenvectors are $\boldsymbol{\nu}_{1,k}$, $\boldsymbol{\nu}_{2,k}$, $\boldsymbol{\nu}_{3,k}$ and $\boldsymbol{\nu}_{4,k}$. Thus, the first eigenimage can be defined as a composition of the eigenvectors:

$$\mathbf{K}_1 = \{\boldsymbol{\nu}_{1,1}, \boldsymbol{\nu}_{1,2}, \boldsymbol{\nu}_{1,3}, \ldots, \boldsymbol{\nu}_{1,40}\} \in R^2 \tag{1}$$

where $\boldsymbol{\nu}_{1,k}$ is the first eigenvector of the correlation matrix composed by the kth column vectors in $\{\mathbf{B}_i, \mathbf{BG}_j \mid 1 \leqslant i \leqslant N,\ 1 \leqslant j \leqslant M\}$. Then the second, third and fourth eigenimages can be obtained in the same way. They are denoted by $\mathbf{K}_2$, $\mathbf{K}_3$ and $\mathbf{K}_4$, respectively. Afterwards, a scalar made by $\mathbf{K}_1$ can be defined:

$$D_{i,k,1} = \mathbf{K}_{1,k}^{\mathrm{T}} \mathbf{B}_{i,k}, \quad 1 \leqslant k \leqslant 40 \tag{2}$$

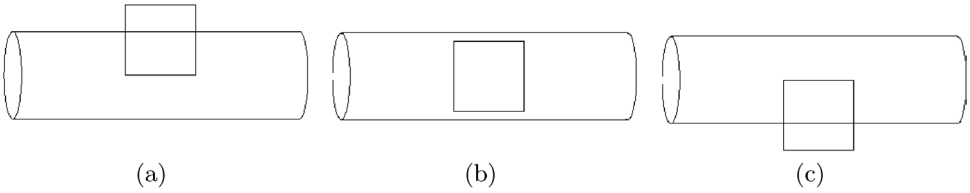

Fig. 1. The positions of observing windows (a) The upper window. (b) The middle window. (c) The lower window.

where $\mathbf{K}_{1,k} = \boldsymbol{\nu}_{1,k}$. Define a vector $\mathbf{D}_{i,1}$:

$$\mathbf{D}_{i,1} = \{D_{i,k,1} \mid k = 1, 2, \ldots, 40\}^{\mathrm{T}} \tag{3}$$

and the feature vector of ith pattern is then defined by:

$$\mathbf{D}_i = \left\{\mathbf{D}_{i,1}^{\mathrm{T}}\mathbf{D}_{i,2}^{\mathrm{T}}\mathbf{D}_{i,3}^{\mathrm{T}}\mathbf{D}_{i,4}^{\mathrm{T}}\right\}^{\mathrm{T}}, \quad 1 \leqslant i \leqslant N \tag{4}$$

where $\mathbf{D}_{i,2}$, $\mathbf{D}_{i,3}$ and $\mathbf{D}_{i,4}$ is made by $\mathbf{K}_2$, $\mathbf{K}_3$ and $\mathbf{K}_4$, respectively. Let us redefine the feature vector of the ith bubble pattern and the jth background pattern as $\mathbf{D}_{B,i}$ and $\mathbf{D}_{BG,j}$, respectively. Therefore, the feature space is composed by the feature vectors $\mathbf{D}_{B,i}$ and $\mathbf{D}_{BG,j}$ where $1 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant M$.

Bubble recognition. In this study, we use a linear classification to segment bubbles from the background. After extracting the bubble and background features, the center of each group is calculated. Let the centers of bubble and background features be denoted by $\mathbf{C}_B$ and $\mathbf{C}_{BG}$, respectively. Each image to be processed is divided into several overlapping sub-images of the same size. The overlapping percentage is 66.7%. If the ith sub-image is defined by S_i, then a scalar can be defined similarly to Eq. (2)

$$E_{i,k,1} = \mathbf{K}_{1,k}^{\mathrm{T}}\mathbf{S}_{i,k}, \quad 1 \leqslant k \leqslant 40 \tag{5}$$

The feature vector of the ith sub-image is obtained and denoted as $\mathbf{E}_i$. Then, the sub-image belongs to bubble group if it satisfies: $|\mathbf{E}_i - \mathbf{C}_B| < |\mathbf{E}_i - \mathbf{C}_{BG}|$. Via this way, most sub-images that contain bubble(s) are recognized. Since there is a 66.7% overlapping, there might be several sub-images indicating the same bubble. In order to decrease the number of coincidences, each sub-image is allowed to be compared with the local nearby sub-image to get the best estimation. Let $\mathbf{S}_i(c_x, c_y)$ denote the sub-image at center point (c_x, c_y), and $\mathbf{E}_i(c_x, c_y)$ its corresponding feature vector. Then, the best estimation is assumed to be:

$$(c'_x, c'_y) = \arg\left\{\min_{\substack{|c'_x - c_x| < \Delta x \\ |c'_y - c_y| < \Delta y}} \left(\left|E_i(c'_x, c'_y) - C_B\right|\right)\right\} \tag{6}$$

where Δx and Δy depict the shift in x- and y-direction. Here we use $\Delta x = 13$ and $\Delta y = 13$. Based on this scheme, most bubbles can be recognized. However, the smaller bubbles cannot be recognized. This is due to the fact that the small bubbles have less information than background in the sub-images. Another problem is that large bubbles can be recognized by some sub-images. These two problems can be partially solved in Section 2.2.

Database setup. After the process depicted in Section 2.1 bubble recognition, most of the bubbles' positions are roughly recognized. Nevertheless, the accurate positions and contours are not yet known. In order to accurately identify the position and contour, we need another database. In this database, only bubble templates are saved. The user selects the bubble by giving four points around the bubble: left, right, top and bottom of the bubble's contour. Since the bubble shape is assumed to be elliptic, there are four parameters in this model, i.e. the center position (c_x, c_y), and two axes lengths a and b. We do not take the angle into consideration since it increases the computation time and only few bubbles are rotated. Every bubble template is normalized to have the same size, i.e. 41×41 pixels, through a linear image resize. Some information are saved in the database such as the bubble's position, bubble's two axes lengths, bubble appearance (gray-level) and so on. The bubble's position and size are taken into consideration when the templates are chosen for matching in tracking and identification. The number of selecting templates depends on the image itself. Some image sequences have less bubbles, therefore, the bubble appearance has less variance. In this situation, we do not need to select too many bubbles as templates. However, in complicated situations, some bubbles are out of focus. There are too many possibilities of bubble appearance and therefore it is necessary to have more bubble templates to cover as many as possible appearances.

Bubble contour identification. In this study, the template matching is applied to identify the bubble accurate contour. Since there are some initializations of bubbles' positions, the system can focus on their local regions to search the bubbles. Firstly, five templates are selected from the database. The selection strategy is the five templates that have the nearest distances to the current bubble's initialization position. This is due to the fact that there is no additional information except the position. We choose the nearest templates because the bubble appearance is strongly dependent on its position. Generally, the bubble appearance's variance in y-direction is larger than in the x-direction.

Assuming the template image is denoted by $I_2(\mathbf{x})$, where $\mathbf{x} = [x \ y]^{\mathrm{T}}$. The bubble to be identified is denoted by a motion model $I_m(\mathbf{x}, \widehat{\mathbf{T}})$, where $\widehat{\mathbf{T}}$ is a parameter vector $\widehat{\mathbf{T}} = [\hat{d}_x, \hat{d}_y, \hat{a}, \hat{b}]^{\mathrm{T}}$. d_x and d_y are the translations in x- and y-direction. a and b are the axis lengths as defined before. Thus, the parameter vector changes until the error function $J(e(\widehat{\mathbf{T}}))$ defined by the image difference $e(\widehat{\mathbf{T}})$ between $I_2(\mathbf{x})$ and $I_m(\mathbf{x}, \widehat{\mathbf{T}})$ reaches a minimum. The error function is chosen to be the expectation value of the squared image difference

$$J\left\{e(\widehat{\mathbf{T}})\right\} = \frac{1}{2}E\left\{e^2\right\} = \frac{1}{2}E\left\{\left(I_m(\mathbf{x}, \widehat{\mathbf{T}}) - I_2(\mathbf{x})\right)^2\right\} \tag{7}$$

which is assumed to be twice differentiable. A direct solution for the minimization of the error function is given by a global search for the optimal parameter vector $T = \widehat{T}$ resulting in a four-dimensional cross-correlation calculation which causes tremendous numerical complexity [13].

Burkhardt et al. [13–16] propose a modified Newton–Raphson method in searching the minimum of an error value. The iterative structure is given by the equation

$$\widehat{\mathbf{T}}_{k+1} = \widehat{\mathbf{T}}_k - \mathbf{H}^{-1} \cdot g(\widehat{\mathbf{T}}_k) \tag{8}$$

where the gradient vector $g(\cdot)$ and Hessian matrix $\mathbf{H}$ are the first and second derivative of the error function $J(e(\widehat{\mathbf{T}}))$ with respect to the parameter vector $\mathbf{T} = \widehat{\mathbf{T}}$. In [16], the Hessian matrix is calculated only once since the template is very similar to the target candidate. This modification reduces the tremen-

dous computational complexity. The gradient vector is derived directly from the error function

$$\mathbf{g}(\widehat{\mathbf{T}}_k) = \frac{\partial J\{e(\widehat{\mathbf{T}})\}}{\partial \widehat{\mathbf{T}}}\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} = E\left\{\big(I_m(\mathbf{x},\widehat{\mathbf{T}}) - I_2(\mathbf{x})\big)\frac{\partial I_m(\mathbf{x},\widehat{\mathbf{T}})}{\partial \widehat{\mathbf{T}}}\right\}\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} \tag{9}$$

And the Hessian matrix is derived from the gradient vector

$$\mathbf{H}(\widehat{\mathbf{T}}_k) = \frac{\partial^2 J\{e(\widehat{\mathbf{T}})\}}{\partial \widehat{\mathbf{T}}\partial \widehat{\mathbf{T}}}\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} = E\left\{\frac{\partial I_m(\mathbf{x},\widehat{\mathbf{T}})}{\partial \widehat{\mathbf{T}}}\left(\frac{\partial I_m(\mathbf{x},\widehat{\mathbf{T}})}{\partial \widehat{\mathbf{T}}}\right)^{\mathrm{T}} + \big(I_m(\mathbf{x},\widehat{\mathbf{T}}) - I_2(\mathbf{x})\big)\frac{\partial^2 I_m(\mathbf{x},\widehat{\mathbf{T}})}{\partial \widehat{\mathbf{T}}\partial \widehat{\mathbf{T}}}\right\}\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} \tag{10}$$

For the intention to reduce the computational complexity, Burkhardt et al. [15,16] make a coordinate transform:

$$\mathbf{x} = h^{-1}(\mathbf{x}', \widehat{\mathbf{T}}) \tag{11}$$

Therefore, the error function can be rewritten as

$$J\{e(\widehat{\mathbf{T}})\} = \frac{1}{2}E\{\big(I_m(\mathbf{x}') - I_2(\mathbf{x}',\widehat{\mathbf{T}})\big)^2\} \tag{12}$$

And the gradient vector can be rewritten as

$$\mathbf{g}(\widehat{\mathbf{T}}_k) = E\left\{\big(I_m(\mathbf{x}') - I_2(\mathbf{x}',\widehat{\mathbf{T}})\big)\left(-\frac{\partial I_2(\mathbf{x}',\widehat{\mathbf{T}})}{\partial \widehat{\mathbf{T}}}\right)\right\}\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} = -E\left\{\big(I_m(\mathbf{x},\widehat{\mathbf{T}}) - I_2(\mathbf{x})\big)\frac{\partial I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}}\right\}\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} \tag{13}$$

The $\frac{\partial I_m(\mathbf{x},\widehat{\mathbf{T}})}{\partial \widehat{\mathbf{T}}}$ term in Eq. (9) can be replaced by $-\frac{\partial I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}}$ in Eq. (13). Since $I_2(\mathbf{x})$ is a template image, it can be computed once in the beginning. The derivative can be rewritten as

$$\left(\frac{\partial I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}}\right)\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} = \left(\frac{\partial I_2(\mathbf{x})}{\partial \mathbf{x}}\right)\left(\frac{\partial \mathbf{x}'}{\partial \mathbf{x}}\right)^{-1}\left(\frac{\partial \mathbf{x}'}{\partial \widehat{\mathbf{T}}}\right)\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} \tag{14}$$

with (x', y') the coordinate transform of an ellipse without rotation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ax \\ by \end{pmatrix} - \begin{pmatrix} \hat{d}_1 \\ \hat{d}_2 \end{pmatrix} \tag{15}$$

Therefore, the derivative can be derived as

$$\frac{\partial I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}} = \begin{bmatrix} \frac{\partial I_2(\mathbf{x})}{\partial \hat{d}_1} \\ \frac{\partial I_2(\mathbf{x})}{\partial \hat{d}_2} \\ \frac{\partial I_2(\mathbf{x})}{\partial a} \\ \frac{\partial I_2(\mathbf{x})}{\partial b} \end{bmatrix} = \begin{bmatrix} -\frac{\partial I_2(\mathbf{x})}{\partial x}\frac{1}{a} \\ -\frac{\partial I_2(\mathbf{x})}{\partial y}\frac{1}{b} \\ \frac{\partial I_2(\mathbf{x})}{\partial x}\frac{x}{a} \\ \frac{\partial I_2(\mathbf{x})}{\partial y}\frac{y}{b} \end{bmatrix} \tag{16}$$

Similarly, based on the coordinate transformation listed in Eq. (11), the Hessian matrix in Eq. (10) can be rewritten as

$$\mathbf{H}(\widehat{\mathbf{T}}_k) = E\left\{\frac{\partial I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}}\left(\frac{\partial I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}}\right)^{\mathrm{T}} - \big(I_m(\mathbf{x},\widehat{\mathbf{T}}) - I_2(\mathbf{x})\big)\frac{\partial^2 I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}\partial \widehat{\mathbf{T}}}\right\}\bigg|_{\widehat{\mathbf{T}}=\widehat{\mathbf{T}}_k} \tag{17}$$

By this, the computational complexity is tremendously reduced. In [16], based on the assumption that the difference between the template and the target candidate is very limited, the second term in the right-hand side in Eq. (17) can be omitted. However, owing to the fact that there might be large differences between the templates and target bubbles, we cannot omit this term in this study.

The second order derivative in Eq. (17) can then be derived

$$\frac{\partial^2 I_2(\mathbf{x})}{\partial \widehat{\mathbf{T}}_i \partial \widehat{\mathbf{T}}_j} = \begin{bmatrix} \frac{I_{2xx}}{a^2} & \frac{I_{2xy}}{ab} & \frac{I_{2x}-xI_{2xx}}{a^2} & \frac{-yI_{2xy}}{ab} \\ - & \frac{I_{2yy}}{b^2} & \frac{-xI_{2xy}}{ab} & \frac{I_{2y}-yI_{2yy}}{b^2} \\ - & - & \frac{x^2I_{2xx}-xI_{2x}}{a^2} & \frac{xyI_{2xy}}{ab} \\ - & - & - & \frac{y^2I_{2yy}-yI_{2y}}{b^2} \end{bmatrix} \tag{18}$$

where $I_{2x} = \frac{\partial I_2(\mathbf{x})}{\partial x}$, $I_{2xx} = \frac{\partial^2 I_2(\mathbf{x})}{\partial x^2}$ and so on. Notably, Eq. (18) is a symmetric matrix. Although the Hessian matrix has to be updated in every iteration since $I_m(\mathbf{x},\widehat{\mathbf{T}})$ is changed, however, a tremendous computation time is reduced via the coordinate transformation.

2.2. Bubble tracking

Bubble tracking process is a combination of template selection, bubble contour identification, and coincidental bubble removal. As we have mentioned before, the template selection in the bubble contour identification is important because bubbles' appearances depend on the positions of bubbles. In the tracking phase, the bubbles in the last image are identified. Thus, additional information are obtained such as bubble's two axis length a and b. A cost function is defined in order to get five templates from the database to be candidate templates for tracking a bubble.

$$f_j = c_1 d_{j,x} + c_2 d_{j,y} + c_3 d_{j,a} + c_4 d_{j,b} \tag{19}$$

where $c_1 \sim c_4$ are constants. $d_{j,x}$ and $d_{j,y}$ are the x- and y-coordinate distances between the bubble candidate and the jth template in the database, respectively. $d_{j,a}$ and $d_{j,b}$ are the absolute difference of axis length a and b between the bubble candidate and the jth template in the database, respectively. Five templates in the database that have the smallest cost values are selected. Each template candidate is tested by giving the same initialization vector $[\,d_{1,i} \quad d_{2,i} \quad a_i \quad b_i\,]^{\mathrm{T}}$. $d_{1,i} = d_{1,i+1}$ is x-translation inherited from the identified bubble in the last image, i.e. $(i+1)$th image. $d_{2,i} = d_{2,i+1} - c\sqrt{a_{i+1}}$ is the y-translation modified from the last image, where $c = 8$ and a_{i+1} is the axis length in the last image. Since the backwards tracking is applied, the bubble goes downwards and thus we use negative sign in the position estimation. The cost value in Eq. (7) is calculated after convergence while it reaches a local minimum. The least one is chosen and its corresponding template is used to track the bubble.

Once the template is determined, a random process based on a Gaussian distribution in estimating the translation and axis length is applied. The probability in estimating axis length a_i is given by

$$p(a_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(c_a a_i - a_{i+1})^2}{2\sigma^2}} \tag{20}$$

with $c_a = 1$. The estimation of axis length b_i is the same as shown in Eq. (20) except that c_a is replaced by c_b with $c_b = 1.25$. The axis length a_i and b_i can be decreased or increased since the bubble might change its shape. The estimation of x-translation is given by

$$d_{1,i} = d_{1,i+1} + c_x r_x t a_{i+1} \tag{21}$$

where t is the index number, $t = 1, 2, \ldots, 40$. r_x is a random number that satisfies the Gaussian distribution with zero mean and $\sigma = 1$, $c_x = 0.0156$. The estimation of y-translation is given by

$$d_{2,i} = d_{2,i+1} - \frac{c_y a_{i+1} b_{i+1} t}{N} \tag{22}$$

where $c_y = 0.0313$ and $N = 40$. N is the number of total estimations. These estimations are given as initializations in tracking bubbles. This mechanism avoids being trapped into a local minimum and tries to find the global minimum. The final convergence results are saved and their corresponding cost values are compared. The least one is chosen and its parameter set $[\,d_1 \quad d_2 \quad a \quad b\,]^{\mathrm{T}}$ indicates the geometric parameters of the bubble to be tracked. If the minimal cost value satisfies $\frac{J(e(\widehat{\mathbf{T}}))}{100} < 2$, then it is assumed to be correctly tracked. Otherwise, it might fail due to the following reasons:

- The bubble is occluded by some other bubble.
- The bubble might be near the image boundary resulting in a higher cost value.
- The template might be not suitable for this bubble or there are no matched template in the database. This might be due to that this bubble is out of focus.

The above paragraph describes how to track a bubble from $(i + 1)$th image to current ith image. However, the bubble recognition in Section 2.1 has to be applied to every image. Afterwards, The bubble contour identification is applied on every recognized bubble before tracking. Therefore, some bubbles are identified that might be tracked in the current ith image. If there are coincidental bubbles, only one bubble remains. Our strategy is choosing the largest bubble to be remained. This is based on the observation that some inner part of a large bubble is similar to a bubble in vision.

2.3. Manual correction

In order to make sure that every bubble is detected and identified correctly and accurately in five-hundred images of a sequence, the manual correction is necessary. This is based on the fact that this system cannot assure every image sequence of different image quality in a constant quality. The identification ratio strongly depends on the image quality. If the illumination condition is bad or if many bubbles are out of focus, then the ratio will decrease apparently. A friendly GUI is designed for the user to delete the false-positive bubble and to add/identify the false-negative bubble manually.

3. Results

Fig. 2 shows examples of bubble and background patterns from the database. There are totally twenty bubble patterns and twenty background patterns. Each pattern is of the same size of 40×40. The eigenimages are then calculated and shown in Fig. 3. In Fig. 4(a), the feature space of the bubble patterns is illustrated. Each feature vector is of the size 1×160 and there are totally twenty features of each category. In Fig. 4(b), the feature space of the background is shown and it is easily to be noticed that there are apparent differences between these two categories.

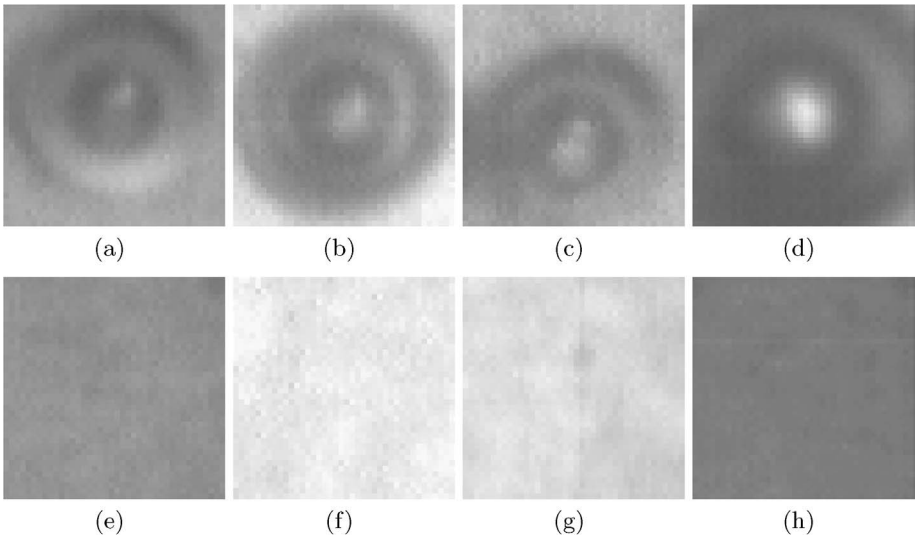

Fig. 2. Bubble and background patterns: From (a) to (d) are bubble patterns. The rest are background patterns.

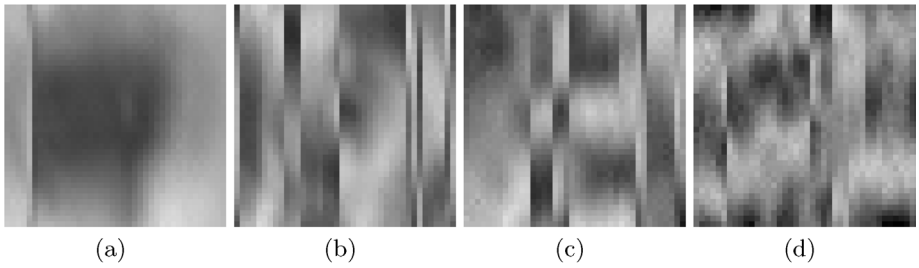

Fig. 3. Eigen images. From (a) to (d) is the first, second, third, and fourth eigenimage.

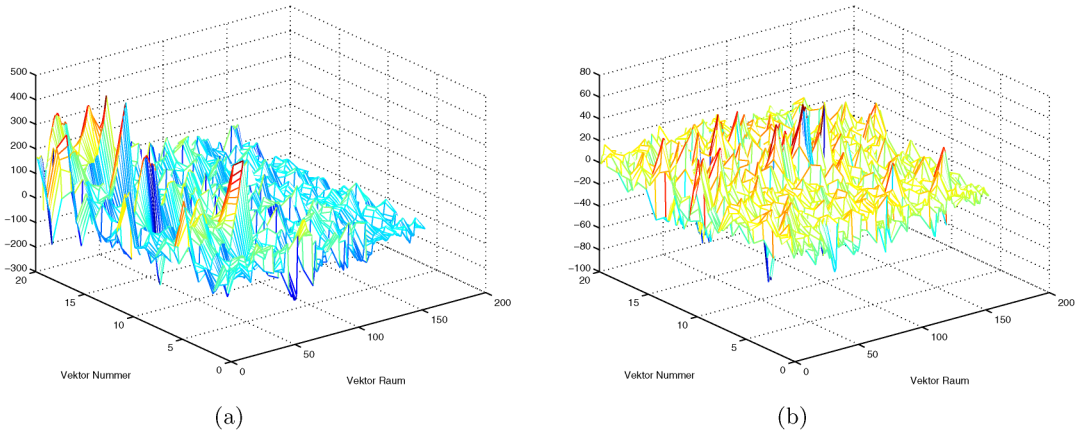

Fig. 4. The feature space plot. (a) is the bubble's and (b) is the background's feature space plot.

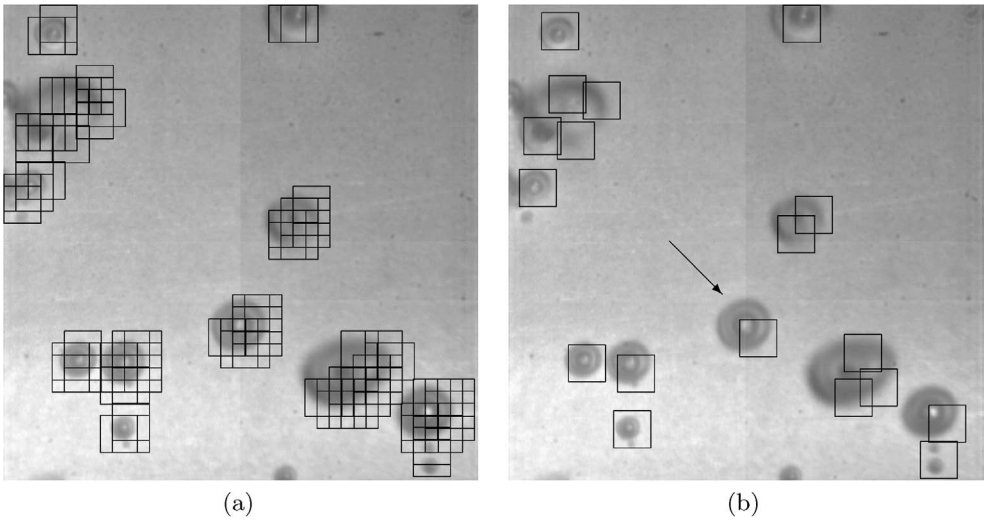

Fig. 5. (a) Bubble recognition pre-processing result. (b) Bubble recognition final result.

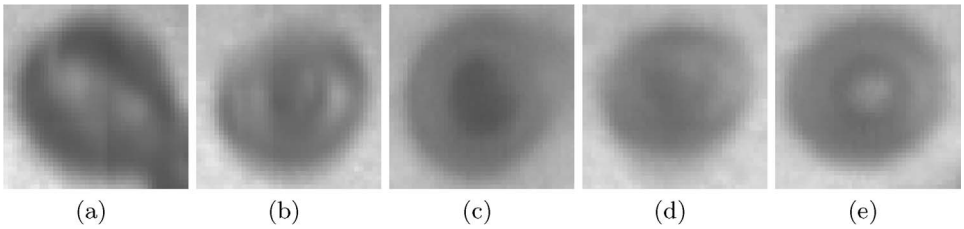

Fig. 6. Five template candidates.

Each image in an image sequence has to be processed by the bubble recognition. The result of the first image is shown in Fig. 5. Fig. 5(a) is the result of bubble recognition and (b) is the final result after coincidence detection. Most bubbles are recognized and marked by squares. Each square is of the same size of the eigenimage. Large bubbles are recognized by more than one square. This is reasonable because the square is not large enough to contain the whole bubble. By the way, some squares do not recognize the accurate bubble positions. This problem can be solved in the bubble contour identification procedure.

In the bubble contour identification, every square has to be processed via the template matching based on the modified Newton–Raphson method. An example is demonstrated how the bubble marked by an arrow in Fig. 5(b) is identified. In this example, we have forty-eight templates in the database. Five of them, shown in Fig. 6, are automatically selected to test and the best one is chosen to be the template. The test method is giving the five candidates the same initial parameter sets $\widehat{\mathbf{T}}$ and use the modified Newton–Raphson method described in Section 2.1 (Bubble contour identification) to obtain the minima of the error function. Only the minimum error value of each template is chosen and they are shown in Fig. 7. The criterion to select the best template is according to the error values. A smaller error value indicates a better match between the template and the bubble to be identified. In this example, the fifth template candidate is therefore chosen. From the view point of vision, it is the most likely template to the target bubble. After the template is determined, the modified Newton–Raphson method is applied again to identify the bubble by giving more initial parameter sets. In order to demonstration, an initial parameter set is given $Zi = [\, d_1 \quad d_2 \quad a \quad b\,]^{\mathrm{T}}$ where (d_1, d_2) is the center coordinate of the square and $a = 1, b = 1$. Fig. 8 shows how the bubble is identified in iterations. From (a) to (j) we see the resulting image in the square. Once the bubble locates in the center of the square and its boundary locate the same position to the template, then the matching is done. Fig. 9 illustrates the error value of each iteration. From the plot, one can easily notice that the convergence takes place at iteration number six. The following iterations assure that the template matching is already in a stable state. This example demonstrates how efficient the modified Newton–Raphson method is.

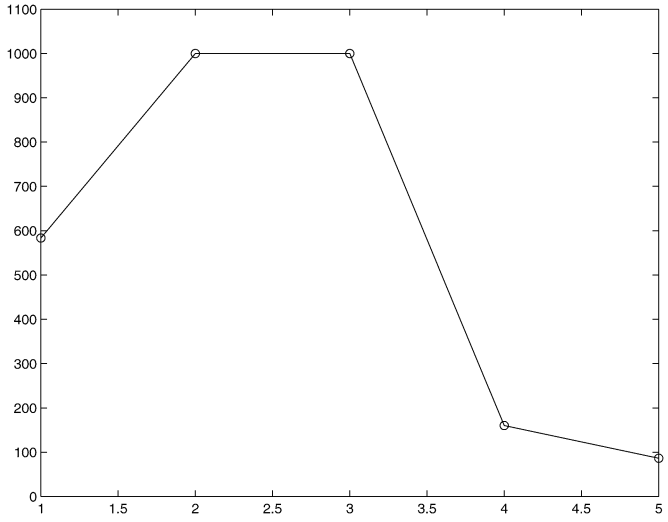

Fig. 7. Error criterion plot. The abscissa indicates the template number and the ordinate indicates the error criterion value.

Fig. 10 is the tracking result of the first image. The boundary of each bubble is represented by four points. This representation is better than a fully ellipse contour because the later one will interfere the observation of the boundary. After five-hundred images are processed, manual correction follows. Therefore, the correct identification ratios can be calculated for each image and they are illustrated in Fig. 11. The mean correct identification ratio of this sequence is 55%.

4. Discussion

This study can be separated into two parts:

(1) the object recognition and
(2) multi non-rigid object tracking via static camera.

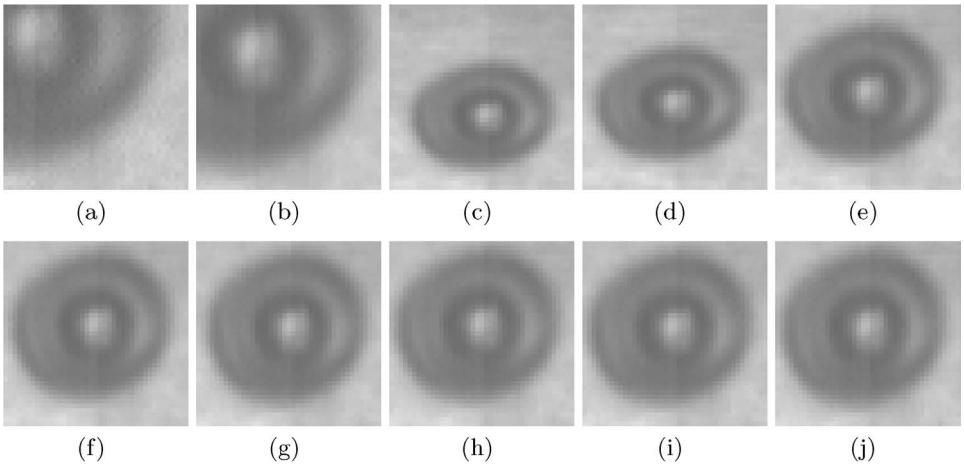

Fig. 8. Modified Newton–Raphson method in identifying a bubble.

Some similar works such as kernel-based object tracking [17] and trust-region method [18,19] have still background. The initial contour is given manually in [17] and automatically in [19]. In [19], the object can be detected via a background subtraction process. The background is obtained through a training process in which there are few or no objects in the image scene. However, in our applications, some sequences have many flowing bubbles in the image scene so that it is non-trivial to obtain a reliable background image. That is the reason why we need additional database to recognize the bubbles. The number of the patterns in the database are not less than twenty for each group. In the future, a non-linear classification method should be applied to separate the bubbles from background.

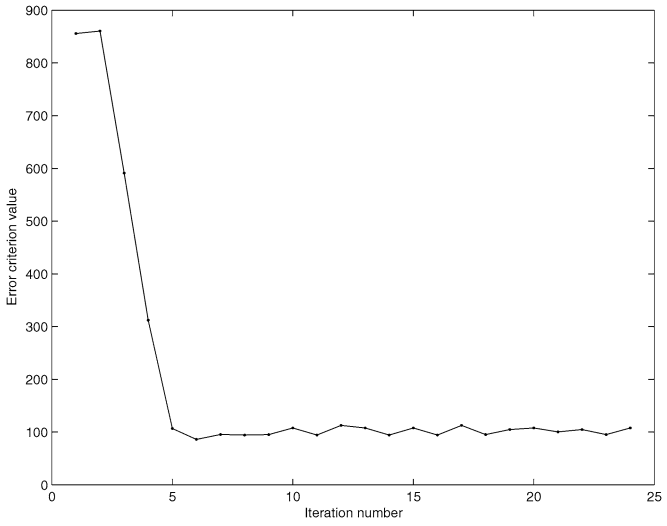

Fig. 9. The error value during the iterations.

In the manual correction, we have a very strict requirement. Some bubbles' contours identified automatically are only few pixels difference to the manual corrected contours. Those contours are removed and corrected manually. This is due to the effect of the shadows and reflections since we have only forty-eight templates that cannot convey all of the situations especially for those bubbles out of focus. However, another reason of low ratio of identification is the feature and cost function. In this study, we use the original gray-level as feature and cross-correlation as similarity measure. It is known that the correlation is sensitive to illumination change. That is why we force the expectation of $I_m(\mathbf{x}, \widehat{\mathbf{T}})$ and $I_2(\mathbf{x})$ to be zero before the similarity measure.

5. Conclusion

This paper presents a work based on templates to identify and track non-rigid objects in images sequences. In the first phase, the bubbles are roughly recognized via classification in the feature space after a Karhunen–Loève transform of the original images. In the second phase, the modified Newton–Raphson method is applied to identify the accurate contours via a correlation method. This study provides a tool to extract the bubbles' positions, bubbles' size, moving speed, nucleation

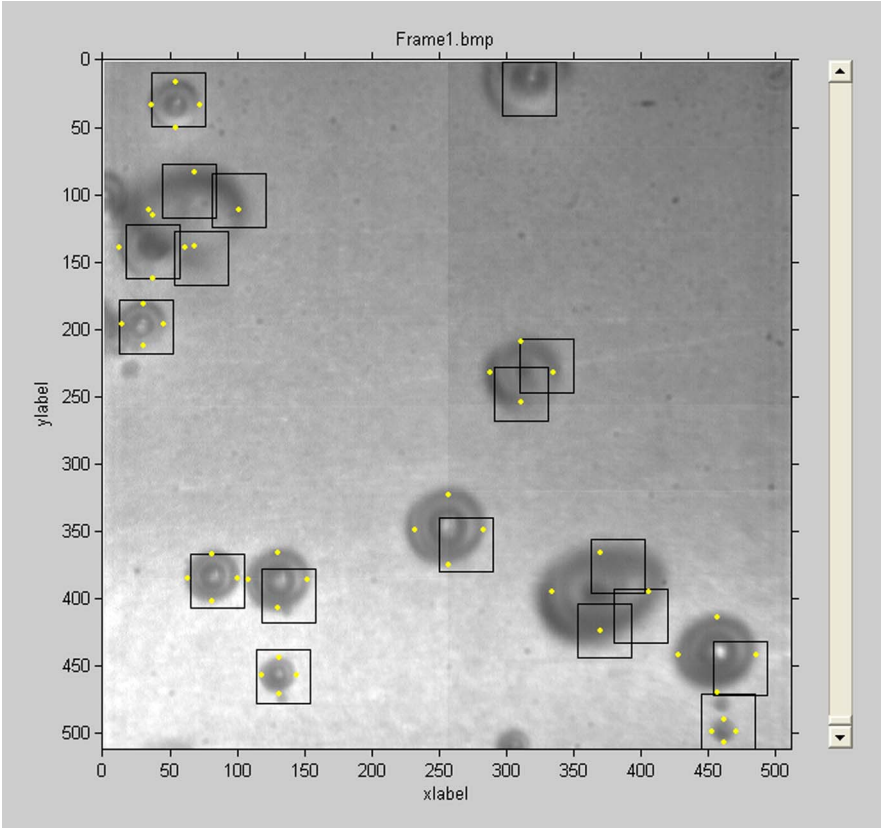

Fig. 10. The tracking result.

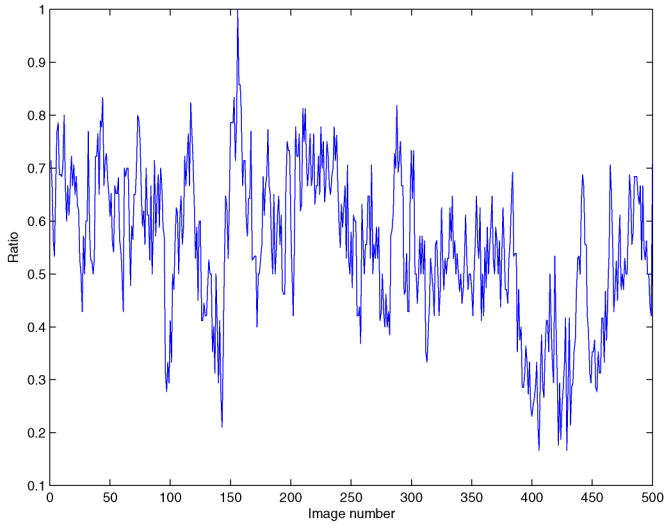

Fig. 11. The identification ratio.

sites; all these being important parameters for interpreting high-speed image sequences in our cooperative research group for exploring heat transfer phenomena.

Acknowledgements

The financial support for this research from Deutsche Forschungsgemeinschaft (**DFG**) is gratefully acknowledged. The authors thank Prof. D. Gorenflo, University of Paderborn, for providing the bubble images. The first author thanks Dr. A. Luke for valuable discussions.

References

[1] S. Caplanis, Wärmeübertragung und Blasenbildung an Hochleistungs—Verdampferrohren, PhD thesis, Paderborn University, 1997.
[2] A. Luke, Beitrag zum Einfluss der Mikrostruktur von Heizflächen auf den Wärmeübergang beim Blasensieden, PhD thesis, Paderborn University, 1996.
[3] D. Cheng, H. Burkhardt, Bubble tracking in image sequences, Int. J. Thermal Sci. 42 (2003) 647–655.
[4] D. Cheng, H. Burkhardt, Bubble recognition from image sequences, in: Eurotherm Seminar, No. 69, Inverse problem and experimental design in thermal and mechanical engineering, Poitiers, France, March 2001.
[5] D. Cheng, H. Burkhardt, Automatic bubble nucleation sites identification in an image sequence, in: Eurotherm Seminar, No. 71, Visualization, Imaging and Data Analysis in Convective Heat And Mass Transfer, University of Reims, France, 2002, pp. 351–358.
[6] O. Borchers, Zweiphasen-Particle-Tracking-Velcimetry (PTV) zur detaillierten Analyse der Hydrodynamik von Blasensäulenreaktoren, PhD thesis, Institut für Chemische Verfahrenstechnik der Universität Stuttgart, Germany, Juni 2002.
[7] F. Hering, C. Leue, D. Wierzimok, B. Jähne, Particle tracking velocimetry beneath water waves. Part I: Visualization and tracking algorithms, Experiments in Fluids 23 (1997) 472–482.
[8] D. Cheng, H. Burkhardt, Image analysis of bubbles in high speed image sequences, in: Proceedings of Thermophysical Properties and Transfer Processes of New Refrigerants, Paderborn, Germany, October 2001, pp. B5–4.
[9] Matlab 7.0, 2003.
[10] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. of Intl. Conf. on Computer Vision, ICCV, Vancouver, Canada, 2001.
[11] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, New York, 1999.
[12] B. Schölkopf, A.J. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
[13] N. Diehl, H. Burkhardt, Planar motion estimation with a fast converging algorithm, in: Proc. of Intl. Conf. on Pattern Recognition, ICPR, Paris, France, 1986.
[14] H. Burkhardt, H. Moll, Ein modifiziertes Newton–Raphson-schema zur modelladaptiven identifikation von laufzeiten, Technical Report, University of Karlsruhe, 1978.
[15] H. Burkhardt, H. Moll, A modified Newton–Raphson search for the model-adaptive identification of delays, in: 5th IFAC Symposium on Identification and System Parameter Estimation, 1979, pp. 1279–1286.
[16] N. Diehl, Methoden zur allgemeinen Bewegungsschätzung in Bildfolgen, PhD thesis, University of Hamburg, Harburg, 1988.
[17] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Analysis Machine Intelligence (PAMI) 25 (5) (2003) 564–577.
[18] H. Chen, T. Liu, Trust-region methods for real-time contour tracking, in: Proc. of Intl. Conf. on Computer Vision (ICCV), vol. II, 2001, pp. 717–722.
[19] T.-L. Liu, H.-T. Chen, Real-time tracking using trust-region methods, Trans. Pattern Anal. Machine Intelligence (PAMI) 26 (3) (2004) 397–402.