Pattern Recognition Letters 101 (2018) 60–66

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Comparison of bubble detectors and size distribution estimators

Jarmo Ilonen[a], Roman Juránek[b], Tuomas Eerola[a,*], Lasse Lensu[a], Markéta Dubská[b], Pavel Zemčík[b], Heikki Kälviäinen[a]

[a] Machine Vision and Pattern Recognition Laboratory (MVPR), School of Engineering Science, Lappeenranta University of Technology, P.O. Box 20, FI-53851 Lappeenranta, Finland
[b] Faculty of Information Technology, Brno University of Technology, Božetechnova 2, Brno 61266, Czech Republic

ARTICLE INFO

Article history:
Received 14 March 2017
Available online 22 November 2017

Keywords:
Bubble detection
Size distribution estimation
Circular arrangements
Boosting-based detection
Convolutional neural networks
Pulping

ABSTRACT

Detection, counting and characterization of bubbles, that is, transparent objects in a liquid, is important in many industrial applications. These applications include monitoring of pulp delignification and multiphase dispersion processes common in the chemical, pharmaceutical, and food industries. Typically the aim is to measure the bubble size distribution. In this paper, we present a comprehensive comparison of bubble detection methods for challenging industrial image data. Moreover, we compare the detection-based methods to a direct bubble size distribution estimation method that does not require the detection of individual bubbles. The experiments showed that the approach based on a convolutional neural network (CNN) outperforms the other methods in detection accuracy. However, the boosting-based approaches were remarkably faster to compute. The power spectrum approach for direct bubble size distribution estimation produced accurate distributions and it is fast to compute, but it does not provide the spatial locations of the bubbles. Selecting the most suitable method depends on the specific application.

© 2017 The Author(s). Published by Elsevier B.V.
This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

This paper focuses on estimating the size distribution of bubbles, or more generally, transparent approximately spherical objects in a liquid. The research is driven originally by the pulpmaking industry, in particular the development of the pulp delignification process. Pulp delignification with oxygen is a very energy-intensive operation. To optimize and control the process, it is essential to be able to characterize the process, especially the sizes of the oxygen bubbles. The recent progress in camera and illumination technologies has made it possible to capture images inside the process machines. Mutikainen et al. [29] present an imaging setup applied to the pulp mill environment. From the produced images (see Fig. 1), the bubble size distribution could be determined by manually marking the individual bubbles. However, manual analysis of the images is very time-consuming, which motivates the development of automatic methods for estimating the bubble size distribution.

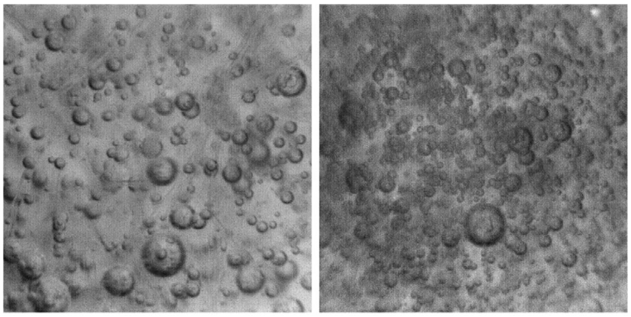
The automatic determination of the bubble size distribution can be performed either by detecting and characterizing each bubble in an image or by estimating the size distribution directly using, for example, frequency information. Of these two approaches, the one based on detection is more common. The bubble detection problem is not easy to solve because the bubbles are transparent and the illumination conditions inside the process machines are challenging, which causes the bubble appearance to vary. In the images, the bubbles appear as roughly circular objects which motivates to solve the problem by detecting circles.

Two common approaches are used to detect circular objects: geometry-based and appearance-based approaches. In the geometry-based approach, a circular model parameterized by its center and radius is fitted to the image edge map. These methods typically utilize a voting technique, such as the Hough Transform (HT) [10] or its modifications [23]. To address the main weaknesses of the HT, such as computational complexity and storage requirements, several revisions of the basic HT have been proposed (see e.g. [27]). However, despite the progress the geometry-based approaches suffer from a large number of false positives and are sensitive to noise. Moreover, they often fail to detect small blob-like bubbles that do not have a ridge edge expected by the model [35].

The appearance-based approaches use typically a sliding window in combination with a classifier/regressor in order to detect the objects of interest. This class of methods include boosting-based detectors [9,36], HOG-based detectors [8,12], or recently emerging convolutional networks [15]. These approaches require a

[*] Corresponding author.
E-mail address: tuomas.eerola@lut.fi (T. Eerola).

https://doi.org/10.1016/j.patrec.2017.11.014
0167-8655/© 2017 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)



(a) (b)

Fig. 1. Examples of pulp suspension images with different process variables: (a) 1000 rpm; (b) 1380 rpm.

large amount of annotated training data covering all possible conditions and appearances of target objects. Insufficient data makes detector performance worse, causing lower detection rates and higher false positive rates.

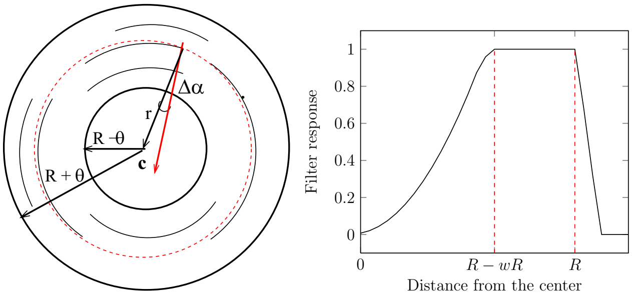
Several bubble detection methods that focus on resolving the overlaps between bubbles exist. For example, Karn et al. [21] proposed a method that utilizes morphological operations and the watershed transform to detect individual bubbles and Zhong et al. [39] makes use of concave points. However, these methods are developed for high quality images where the bubbles are clearly distinguishable from the background. Therefore, the methods are not applicable for noisy industrial images.

The problem of cell segmentation shares common characteristics with bubble detection. Methods developed for cell segmentation include intensity thresholding, morphological operations, region growing, feature detection, and deformable model fitting [26]. There are, however, some important differences between the bubble detection problems and the cell segmentation problems: the cells contain more complex shapes while typical bubbles are close to perfect circles, cells have smaller variation in size assuming only certain cell type is considered, a small bubble can appear fully inside larger bubble, and bubbles are more transparent.

The cell detection is often solved as blob detection problem. Bernardis and Yu [4] proposed a method to detect blobs defined as small round regions in an image. The detection is carried out by solving a region segmentation problem where the blobs form one region and the areas between blobs form the other. Arteta et al. [1] further developed a similar blob detection approach by proposing a more complex statistical model that uses detected blobs as object candidates. The blob detection approach, however, cannot be applied to bubble detection as such since bubbles typically have distinguishable edges, but are otherwise transparent. Therefore, bubble detection should be based on edge point, instead of detection of uniform blobs.

Direct methods for the bubble size distribution estimation also exist (e.g., [19]). They do not require individual bubbles to be detected. Such methods usually rely on the assumption that the image does not contain other objects besides bubbles and the size distribution is estimated by analyzing the frequency information. The direct methods are often fast to compute and are not sensitive to frequent overlaps between bubbles, but may become unreliable if the conditions (e.g., lighting or the amount of bubbles) change considerably compared to the method training phase.

Optical granulometry is a process of computing a size distribution of grains from images. The classical approach obtains the size distribution by applying a series of morphological opening operations [24]. It is, however, only applicable for binary images. More sophisticated modifications of the original method have been proposed since (see, e.g., [33]) and have been applied to, for example, biometric image analysis [16] and grain size analysis of sediments [14]. However, the granulometry based methods are not likely suitable for estimating the size distribution of the bubbles since the bubbles appear as overlapping rings rather than regions with approximately constant color.



(a) (b)

Fig. 2. CCA: (a) The model; (b) The weight function for the edge distance from the center.

In this work, different promising methods to detect bubbles and to estimate the bubble size distribution are compared. The set of methods is selected so that the different approaches to solve the problem are covered including geometry-based methods, appearance-based methods with both hand-crafted and learned features, as well as, a direct method for the size distribution estimation. Both specific methods developed particularly for bubbles or circular objects as well as general purpose object detectors are included. The experiments were carried out using a novel and extensive data set consisting of challenging images from real industrial environment.

2. Bubble detectors

2.1. Concentric Circular Arrangements

Strokina et al. [35] proposed a method to detect bubbles formulated as detection of Concentric Circular Arrangements (CCAs). CCA $H = (\mathcal{A}, c, r, \theta)$ is a set of concentric circular arcs $\mathcal{A}$ with a center c located within an annulus of radius r and width 2θ (see Fig. 2(a)). An arc is a connected component $A = (P, r_a)$ parameterized by a pixel list P and a radius r_a. The score of a CCA is comprised by the contribution of separate pixels located within the CCA area. The contribution of a pixel $p \in A$ with an orientation α_p to the support of a CCA hypothesis with a center c is computed as

$$f_p = g(\Delta\alpha) \cdot q(r_a) \tag{1}$$

where $\Delta\alpha = \left|\alpha_p - \arctan\left(\frac{|x_c - x_p|}{|y_c - y_p|}\right)\right|$ describes if the orientation of the pixel is consistent with the CCA model (see Fig. 2(a)). The weight function for the pixel orientation $g(\Delta\alpha) = 1/(1 + \Delta\alpha)$ puts less weight to the edges with inconsistent orientation. The function $q(r_a)$ weights the pixel input depending on its location within the CCA area. The weight function $q(r)$ for the edge distance from a center used in this work is presented in Fig. 2(b). Its shape reflects the fact that the inner structure of a bubble also votes for a hypothesis.

The CCA-based bubble detection [35] starts with the computation of the oriented edge map by filtering image with second derivative zero-mean oriented Gaussian filters in eight directions followed by non-maximum suppression [6]. After that, the CCA hypotheses are generated by sampling from the edge map using procedure similar to RANSAC [13]. A CCA hypothesis is generated from the arc if it satisfies the following requirements: (i) it receives sufficient support from the pixels located within the CCA area and (ii) the support comes from a sufficient number of directions. The center location and the radius of the hypothesis are optimized by maximizing the support with the Neadler–Mead Simplex method [30]. Finally, the non-maximum suppression is performed on the CCA parameters to prevent multiple hypotheses for a single bubble.

2.2. Boosting-based detection

Object detection based on Boosting (AdaBoost algorithm or its modifications) is one of the most widely known and used technique today. It has been successfully applied to detection of faces [36], pedestrians [9], and other object categories including bubbles in a preliminary study [34]. The main advantages of this method are the simplicity of implementation and the speed of detection of objects. Detectors based on AdaBoost were successfully implemented on GPUs and programmable hardware [38]. During the last decade, much attention has been given to different aspects of the method in order to handle multiview detection, improve computational performance and detection accuracy, specifically as follows: Training algorithm – Discrete/Real AdaBoost [36], GentleBoost, VectorBoost [18], etc; detector structure – cascade [36], soft-cascade [5,37], constant soft-cascade [9] and various tree structures [18]; weak classifiers – Haar features, Local Binary Patters, Local Rank Differences, decision trees; and to underlying image features used for classification – grayscale image, HOG, decorelation filters, etc. The particular choice depends on the task at hand.

The classification model H is an additive sequence of weak classifiers h evaluated on input features $\mathbf{x}$ as

$$H(\mathbf{x}) = \sum_{t=1}^{T} h_t(\mathbf{x}) \tag{2}$$

where the training selects h by greedy optimization [36].

The detector works on the sliding window principle. On pyramidal image representation, each level is scanned with the detector, and each image sub-window is classified by the model H. Classification speed can be improved by making the rejection decision after the evaluation of every h_t based on the tentative sum of H [9]. Background areas (which are easy to classify), are rejected by the first few weak classifiers, and more time is spent on the areas which are harder to classify. Finally, positive responses are clustered by a non-maxima suppression method in order to obtain the locations and sizes of detected objects.

In this work, we consider Soft-cascade detectors operating on grayscale images or more advanced aggregate feature channels [9] (HOG+L channels specifically). As weak classifiers we use shallow decision trees, as they have proved to be efficient in many tasks [3,25], and Local Binary Patterns, as they outperform the commonly used Haar features and were efficiently implemented on hardware platforms [38]. Fig. 3 shows the overview of our detector architecture.

2.3. Convolutional neural networks

Convolutional Neural Networks (CNNs) have been used to achieve state-of-the-art performance in various image analysis applications, such as image recognition [17] and object detection [32]. A typical CNN consists of convolutional layers that are formed by neurons which look at small overlapping regions of the input image/feature map, a pooling layer used for downsampling, and fully connected layers for the final decision making. Unlike the traditional methods where the decision is made using preselected features, CNNs learn the filters used for feature extraction.

In order the solve the bubble detection problem, the CNNs can be applied in a sliding window manner. For each window, the CNN is applied to compute the probability whether the window contains a bubble or not. This creates a probability map for bubble detections (detection image). To detect bubbles with different sizes, the detection image is computed in multiple scales. A real bubble creates a circular patch of high values, usually in few neighboring scales. To detect these, the detection image is, first, thresholded with a threshold t_c to create binary images. Connected components that do not meet the following conditions are rejected: 1) area is larger than t_a, 2) the proportion of the pixels in the convex hull that are also in the region is larger than t_s, and 3) the ratio of minor and major axis is larger than t_m. This means that only roughly circular blobs with a size large enough are accepted. After this the detection images in different scales are scaled to a common scale, and a detection cube with detection images from all scales is created. The centroids of the blobs in the detection cube are then detected. The location of the centroid in the depth direction represents the scale (radius) of the detected bubble. Finally, bubbles which are found only in one scale are rejected. The final step is not applied for the smallest scales (highest resolution) due to the fact that a circular arrangement of small bubbles occasionally looks like a large bubble, but they are usually detected only in one scale.

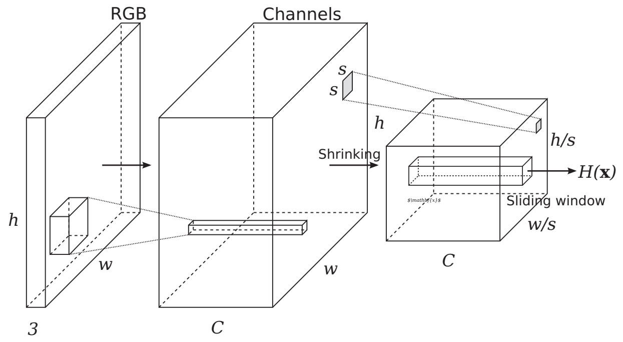


Fig. 3. Feature channels extraction from the input image. First, C channels are extracted and then shrinking by factor s is applied. We use HOG+L channels and $s = 2$ for ACF and WladBoost+Trees models, and Gray channel and $s = 1$ for WaldBoost+LBP.

It should be noted that much faster CNN-based object detection methods exist, such as Faster R-CNN [32] that uses computationally efficient ways to create region proposals that are classified by using a CNN, and You Only Look Once (YOLO) [31] that uses a CNN to predict bounding boxes by only processing the image once. However, the sliding window approach, while slow to compute, can be seen as a method to obtain the upper limit for the performance of approaches based on region proposals. If the same CNN is used for classification, R-CNN can exceed the performance of the sliding window approach only if the method to generate the region proposals does not miss any bubble that the sliding window approach found. YOLO is not suitable for large images with hundreds of objects (e.g., bubbles) due to the limited number of predicted bounding boxes and known problems on detecting objects with a small size [31].

3. Bubble size distribution estimators

Given the radii of the detected bubbles, computing the bubble size distribution is straightforward. However, it is possible to estimate the bubble size distribution directly from the image without detecting the bubbles. In [11,22], a method was presented for determining the bubble size distribution using a stationary planar boolean model and two-point set statistics. The method, however, requires a binary image where the bubbles and background are separated, and it is, therefore, unsuitable for noisy industrial image data.

3.1. Power spectrum

Ilonen et al. [19] proposed a method to estimate the bubble size distribution directly from the images. This approach makes the detection of bubbles unnecessary and, therefore, the problems related to overlapping bubbles and computation time caused by large amount of bubbles can be avoided.

Fourier transform of two signals (here bubbles) is the same as taking the Fourier transform for them separately and adding them up. Therefore, the Fourier transform of an image consisting of bubbles is the same as Fourier transforms of images of separate bubbles. However, the power spectrum of two combined signals is not the same as their separate power spectra combined because the phase (location) difference can cause them to nullify each other. In the case of a large set of bubbles located randomly, it is reasonable to argue that their phases overlap predictably on average so that the distribution can still be determined with a good accuracy from the power spectrum.

The method consists of the following steps:

1. Calculate the power spectrum of an image using L frequency bands.
2. Use principal component analysis (PCA) to reduce the data to M dimensions.
3. Use multivariate linear regression to learn the dependency between the power spectrum and bubble count or volume distribution.

The power spectrum is a vector of the portion of a signal power falling into specified frequency bins. It is acquired by applying the 2-D discrete Fourier transform to the signal (image) and computing the energy belonging to L linearly spaced frequency bins, producing a vector P_i for an image i. The frequency range is limited at the lower end because the maximum size of the bubbles is known, and therefore, the lowest possible frequencies caused by actual bubbles is defined.

To reduce the dimensionality, PCA is used. The principal components are calculated from the matrix containing the power spectra of N images, $P_i, i = [1 \ldots N]$. The M principal components can then be used to reduce the dimensionality of the original power spectrum P_i to p_i which is a vector with M components.

Multivariate linear regression [7] is used to find out the relationship between vector p_i and the bubble count or volume distribution D_i in the image. The distributions used are histograms with K bins, with $K = 10$ in the example and all the experiments. The multivariate linear regression is defined as

$$\begin{pmatrix} D_{1,1} & \ldots & D_{1,K} \\ & \vdots & \\ D_{N,1} & \ldots & D_{N,K} \end{pmatrix} = \begin{pmatrix} p_{1,1} & \ldots & p_{1,M} & 1 \\ & \vdots & & 1 \\ p_{N,1} & \ldots & p_{N,M} & 1 \end{pmatrix} \mathbf{X} + \epsilon \tag{3}$$

where $\mathbf{X}$ is the $M + 1 \times K$ matrix to be estimated and ϵ is the noise term. $\mathbf{X}$ is solved as a linear least squares estimation problem. The distribution D_j for a new image j can be estimated by calculating the power spectrum P_j, using PCA to reduce its dimensionality and getting p_j and then

$$D_j = \left(p_{j,1} \ldots p_{j,M} 1\right)\mathbf{X}. \tag{4}$$

4. Experiments

4.1. Data

Images were gathered from the oxygen delignification process of a pulp fiber line that provides a difficult imaging environment resulting challenging data for bubble detectors and size distribution estimation methods. The imaging setup developed for the purpose has been described in [28] and [29]. The experiments were performed with varied mixer rotor speeds from 900 rpm to 1380 rpm (see examples in Fig. 1). Increasing the rotor speed increases the amount of bubbles and decreases their size. Therefore, the images taken with a high rotor speed are more difficult to analyze. All other process variables were kept constant during each measurement session. 20 images were captured with 6 different rotor speeds. The size of the original images was 2588 × 1940, but only the central 1482 × 1482 pixel areas of the images were used. The ground truth (bubble contours) was manually marked by an expert. Due to the large amount of bubbles in each image and low image quality, the ground truth is imperfect. However, it provides a representative set of bubbles to train the detection methods and to estimate the bubble size distribution. The images were divided into separate training and test sets so that the training set contained 72 images and the test set 48 images.

4.2. Parameter tuning and method training

4.2.1. Circular Hough Transform

As a baseline method we selected the phase coding based circular Hough transform (CHT) method by Atherton and Kerbyson [2]. CHT parameters were selected by using a grid search and minimizing the volume estimation error in the training set.

4.2.2. Concentric Circular Arrangements

CCA parameters were selected by using a grid search and minimizing the volume estimation error in the training set. The following values were selected: the number of CCA sectors $N = 10$, the number of sectors where the support of the hypothesis is not zero $K_s = 4$, the maximum radius of bubbles $R_{\max} = 1.52$ mm, the filter response threshold $T_{filt} = 0.93$, the width of the annulus $\theta = 0.5R$, the minimum length of an arc from which a hypothesis is generated $L_{\min} = 15$, the parameter of the cost-function $W = 0.6R$.

4.2.3. Boosting-based detectors

We trained three models that differ in the training algorithms used and features they extract from the input image.

The *ACF* model was trained for reference. We used Piotr's Image and Video Toolbox[1] with default settings, except for the channels and smoothing. We used HOG+L channels as no color information is available in our data, and we oversmoothened images with a 5 pixel triangle kernel prior to shrinking and feature channel computation in order to handle the noise in images. The model size was 52 × 52 pixels with the shrink factor of 2. We applied 3 rounds of bootstrapping with detectors of length 32, 128 and 512, loading 10,000 hard negatives in each round. The final model contained 2048 decision trees of depth 2.

The *WaldBoost+Trees* model is similar to *ACF*. We used the same settings for the channels and model size. The differences are twofold: *1)* training algorithm – WaldBoost [37] – bootstraps hard negative samples in each training round, and *2)* splits in the decision trees are pixel differences instead of simple pixel values.

The *WaldBoost+LBP* model is the simplest one. Here we used just the grayscale image, Local Binary Pattern features, and no shrinking. The model size was 26 × 26 pixels. The LBP features were composed from 3 × 3 adjacent cells, each of size up to 2 × 2 pixels. The feature is parametrized by its position in a sliding window and the size of its blocks.

During detection, a pyramid with 8 levels per octave is created, feature channels are calculated, and shrinking is applied. Each pyramid level is scanned with the trained model, and positive responses are clustered using a simple overlap-based non-maxima suppression algorithm, producing final locations of the detected bubbles.

[1] http://vision.ucsd.edu/~pdollar/toolbox/doc/ , version 3.40.

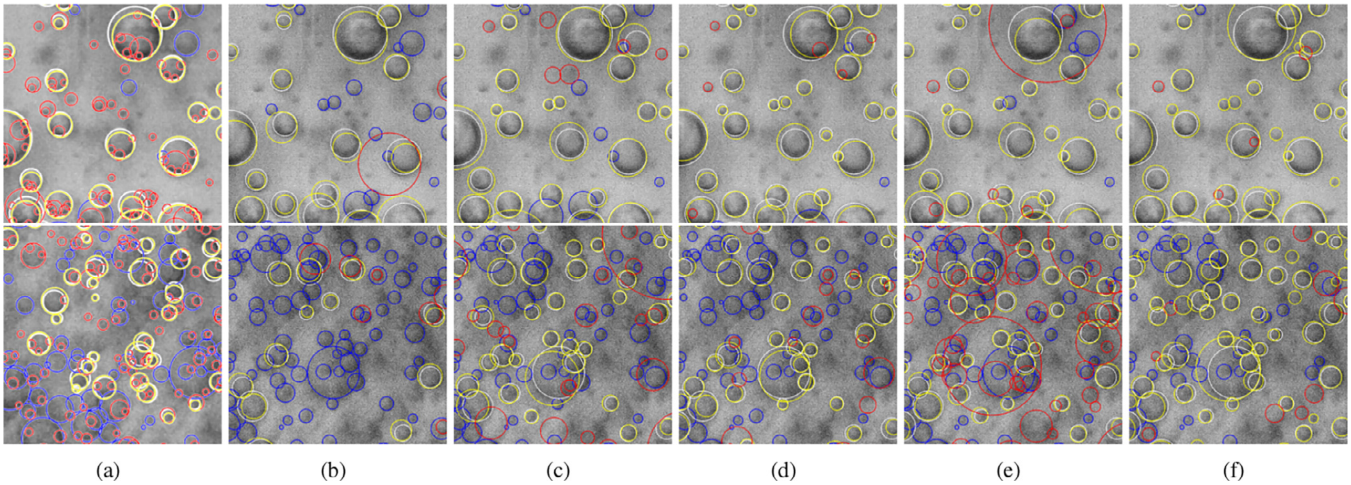


Fig. 4. Examples of bubble detection results with low rotor speed (1000 rpm; top row) and high rotor speed (1380 rpm; bottom row) images. Original images are cropped for visualization purposes. The ground truth markings are in white, true positives in yellow, false negative in blue, and false positive in red: (a) CHT; (b) CCA; (c) WaldBoost+LBP; (d) WaldBoost+Trees; (e) ACF; (f) CNN.

4.2.4. Convolutional neural networks

To train the CNN, the Caffe framework [20] was used. The first convolution layer had 40 outputs with the kernel size of 9 and the second convolution layer had 100 outputs with size 5. Max-pooling was used. Training was performed using image patches of size 28 × 28. If the bubble radius was small enough, the patch was taken directly from the image. For larger bubbles, a larger patch was cropped and resized. From each ground truth bubble in the training data, 10 positive examples were generated (two scales with factors 1.25 and 1.5, shifts in 4 directions and one without a shift). Negative examples (background) were created by selecting 20 random patches so that they were not too close to the actual bubbles. To avoid learning that a bubble border is a bubble, additional 10 randomly selected patches from the border of each large bubble (radius larger than 40 pixels) were included to the negative examples. This way the training data contained 148,860 positive and 317,732 negative examples.

For the bubble detection, a 28 × 28 pixel window for the lowest scale and a scaling factor of $\sqrt{2}$ to create multiple scales were used. The number of scales was 9. Moreover, the following parameter values were used: the threshold for detection images $t_c = 0.9$, the area threshold for detections $t_a = 20$, the proportion of the pixels in the convex hull that are also in the region $t_s = 0.8$, the ratio of minor and major axis $t_m = 0.7$. The rule to reject bubbles that were detected only on one scale was ignored in the 4 smallest scales.

4.2.5. Power spectrum

The power spectrum was calculated from 19 linearly spaced frequency bins, $f = [0.05, 0.5]$, and dimensionality was reduced with PCA to 5.

4.3. Bubble detection

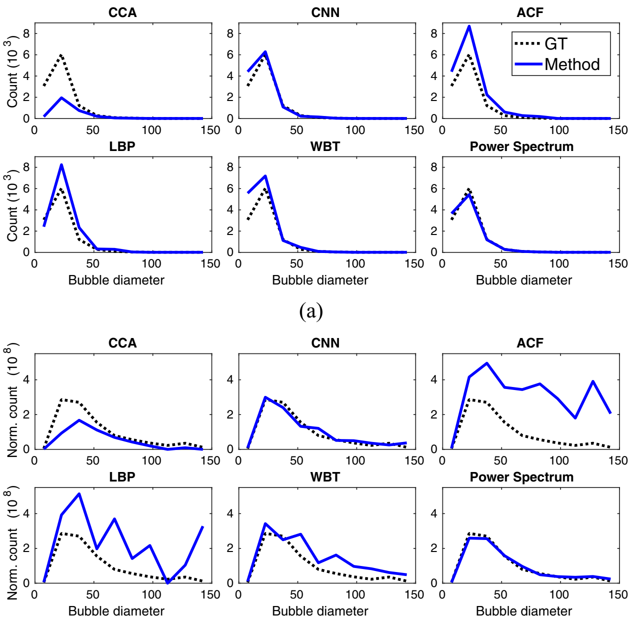
Examples of bubble detection results are shown in Fig. 4. It was noted that the methods found several apparent bubbles that were not by the expert. In order to reliably estimate the true positive and false positive detections, all the unmarked detections were given to the expert for validation who made a binary decision whether the detection was a correct detection or background. The new improved ground truth was used only for the evaluation and not for method training. To compare the methods, three performance measures, True Positive Rate (TPR) and Positive Predictive Value (PPV), and F-measure (the harmonic mean of TPR and PPV) were used. A detection was considered as a true positive (TP) if the overlap (intersect divided by union) between the detected bubble (circle) and the ground truth bubble was higher than 0.5. Multiple detections were not allowed, i.e. only the detection with the greatest overlap is considered as TP, others are treated as false positives. The results with the improved ground truth are shown in Table 1. As it can be seen, the CNN outperformed the other methods by detecting more correct bubbles and producing less false detections. It should be noted that the numbers of false negatives can be slightly underestimated assuming neither the expert nor any of the methods detected certain bubbles. Therefore, the true positive rates can be slightly overestimated.

Table 1
Detection results over all test images.

Method	TPR	PPV	F-measure
CHT	54.69%	18.80%	0.280
CCA	17.81%	73.07%	0.286
CNN	**79.72%**	**84.75%**	**0.822**
ACF	56.69%	44.31%	0.497
WaldBoost+LBP	56.49%	53.12%	0.548
WaldBoost+Trees	58.96%	52.35%	0.563

4.4. Bubble size distribution estimation

The results on bubble size distribution estimation are shown in Figs. 5–7. CHT produces a huge amount of false positives with small bubble sizes. Therefore, CHT is left out from the figures to keep the presentation clear. Chi-squared distances between the estimated distributions and the ground truth distributions are shown in Table 2. Both the distances between the combined distributions over all rotor speeds (Combined) and the mean distances over distributions for different rotor speeds (Mean) are listed. Among the bubble detection methods, the CNN produced the most accurate bubble size distribution. WaldBoost+LBP, ACF, and WaldBoost+Trees suffered from a large amount of false positives. This, together with the small numbers of bubbles in the larger size groups causes noise to the size distribution estimation witch appears as a zigzag pattern in the histograms. Moreover, a limited number of detection scales (window sizes) might cause the sliding window based methods to struggle with certain bubble sizes. ACF resulted a count distribution with a low chi-squared distance to the ground truth although it overestimates the number of bubbles.



(a)

(b)

Fig. 5. Combined size distribution estimates for all rotor speeds: (a) Bubble count; (b) Count normalized by volume.

Table 2
Chi-squared distances of the estimated size distributions from the ground truth.

	Combined		Mean	
Method	Count	Volume	Count	Volume
CHT	0.0805	0.0309	0.0818	0.0468
CCA	0.0706	0.0228	0.0581	0.0311
CNN	**0.0085**	**0.0052**	**0.0103**	**0.0086**
ACF	**0.0022**	0.0638	**0.0093**	0.0785
WaldBoost+LBP	0.0171	0.0523	0.0215	0.0664
WaldBoost+Trees	0.0153	0.0223	0.0247	0.0302
Power Spectrum	**0.0060**	**0.0017**	**0.0136**	**0.0149**

This is due to the fact that the chi-squared distances are computed for normalized histograms (the sum of bins is equal to 1), i.e., it measures the shape of the histograms instead of the count estimates. ACF fails to estimate the volume distribution. The power spectrum based bubble size distribution estimator provided an accurate estimate over all rotor speeds. However, as can be seen from Fig. 6 it slightly underestimates the number of the bubbles with high rotor speeds, and overestimates the number with low rotor speeds. Also, since the method estimates the distribution directly from the image, the user cannot be sure if the method is working correctly.

4.5. Execution times

The exact execution times of different methods are not comparable due to the different programming environments (MATLAB, C, etc.) and degrees of performance optimization. However, indicative conclusions can be made. ACF and WaldBoost+LBP can be efficiently implemented on PC and programmable hardware allowing real-time performance. The current implementation of the CNN takes approximately one hour per image to compute. Although the method performance could be optimized considerably, it is still not possible to reach real-time performance with large images with several hundreds small bubbles, and the CNN is, therefore, only valuable in off-line use. The execution times for CCA and CHT are

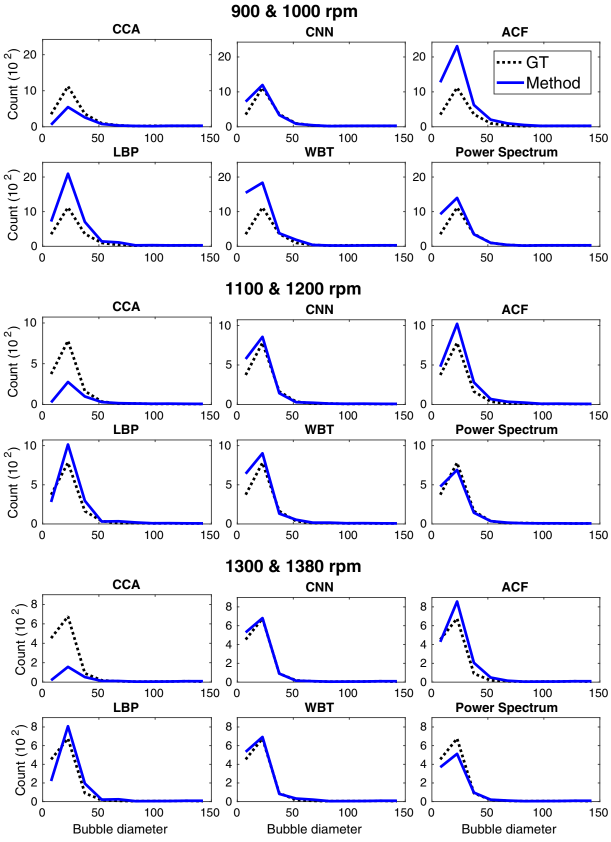


Fig. 6. Size distribution (bubble count) estimates for different rotor speeds.

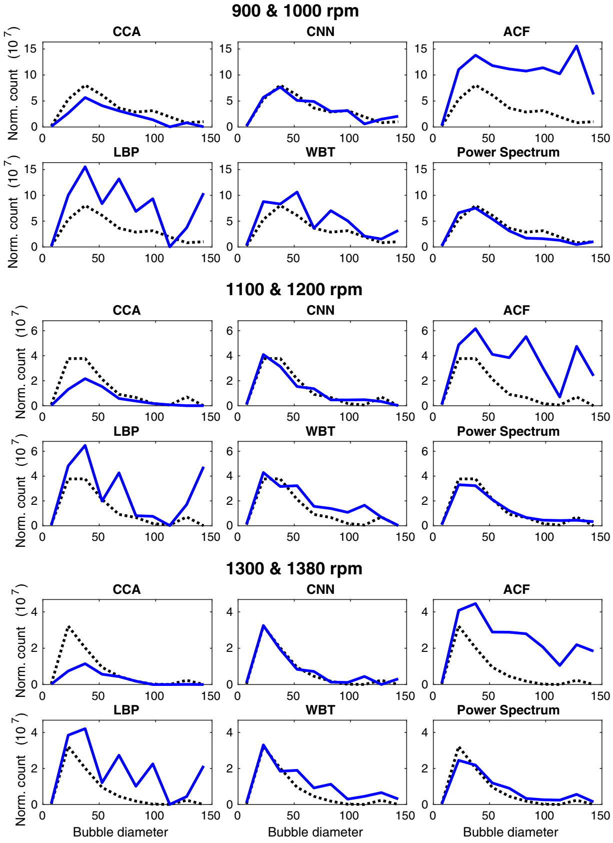


Fig. 7. Size distribution (count normalized by volume) estimates for different rotor speeds.

few seconds per image. The power spectrum method is fast to compute and it is also fast to train making online training possible.

5. Conclusions

We compared various bubble detection and size distribution estimation methods with industrial image data from a pulpmaking process. The CNN-based approach outperformed the other methods in the bubble detection task and provided an accurate estimate for the bubble size distribution. However, the method is slow making it unsuitable for in-line use. The WaldBoost+LBP and WaldBoost+Trees methods are fast to compute and provided reasonably accurate bubble detection results. Out of these two, WaldBoost+Trees was able to detect more bubbles, but it suffered from a slightly higher number of false positives. However, the amount of false positives was similar in different bubble size categories making WaldBoost+Trees a better method for the size distribution estimation. The power spectrum approach for direct bubble size distribution estimation produced accurate distributions and it is fast to compute. However, it does not provide the spatial locations of the bubbles. Moreover, from the user perspective, the method is essentially a black box, and therefore, the user cannot be sure if the method is working correctly. This is not the case with the bubble detection methods where the user can easily see if the method is detecting bubbles correctly. Selecting the most suitable method depends on the application. If real-time performance is not needed, a CNN is the best choice. For in-line measurements WaldBoost+Trees and the power spectrum based size distribution estimator provide the most promising solutions.

Acknowledgments

The research was carried out in the PulpVision project (TEKES Projects No. 70010/10 and 70040/11) funded by the European Union and the participating companies. The work was further supported by the Technology Agency of the Czech Republic (TA CR) Competence Centres project V3C Visual Computing Competence Center (no. TE01020415), and the Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602.

References

[1] C. Arteta, V. Lempitsky, J.A. Noble, A. Zisserman, Detecting overlapping instances in microscopy images using extremal region trees, Med. Image Anal. 27 (2015) 3–16.
[2] T.J. Atherton, D.J. Kerbyson, Size invariant circle detection, Image Vis. Comput. 17 (11) (1999) 795–803.
[3] R. Benenson, M. Omran, J. Hosang, B. Schiele, Ten years of pedestrian detection, what have we learned? ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving, 2014.
[4] E. Bernardis, S. Yu, Finding dots: Segmentation as popping out regions from boundaries, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 199–206, doi:10.1109/CVPR.2010.5540210.
[5] L. Bourdev, J. Brandt, Robust object detection via soft cascade, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
[6] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.
[7] R. Christensen, Plane Answers to Complex Questions: The Theory of Linear Models, Springer, 2011.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 886–893.
[9] P. Dollár, R. Appel, S. Belongie, P. Perona, Fast feature pyramids for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1532–1545.
[10] R.O. Duda, P.E. Hart, Using the hough transform to detect lines and curves in pictures, Commun. ACM 15 (1972) 11–15.
[11] X. Emery, W. Kracht, Ivaro Egaa, F. Garrido, Using two-point set statistics to estimate the diameter distribution in boolean models with circular grains, Math. Geosci. 44 (7) (2012) 805–822.
[12] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, Cascade object detection with deformable part models, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2241–2248.
[13] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[14] P. Francus, An image-analysis technique to measure grain-size variation in thin sections of soft clastic sediments, Sediment. Geol. 121 (3) (1998) 289–298.
[15] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.
[16] J. Goutsias, S. Batman, Morphological Methods for Biomedical Image Analysis, SPIE Press, 2000, pp. 175–272.
[17] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[18] C. Huang, H. Ai, Y. Li, S. Lao, Vector boosting for rotation invariant multi-view face detection, in: IEEE International Conference on Computer Vision (ICCV), 2005, pp. 446–453.
[19] J. Ilonen, T. Eerola, H. Mutikainen, L. Lensu, J. Käyhkö, H. Kälviäinen, Estimation of bubble size distribution based on power spectrum, in: Iberoamerican Congress on Pattern Recognition, 2014, pp. 38–45.
[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM International Conference on Multimedia, 2014, pp. 675–678.
[21] A. Karn, C. Ellis, R. Arndt, J. Hong, An integrative image measurement technique for dense bubbly flows with a wide size distribution, Chem. Eng. Sci. 122 (2015) 240–249.
[22] W. Kracht, X. Emery, C. Paredes, A stochastic approach for measuring bubble size distribution via image analysis, Int. J. Miner. Process. 121 (10) (2013) 6–11.
[23] V. Kyrki, H. Kälviäinen, Combination of local and global line extraction, J. Real-Time Imaging 6 (2) (2000) 79–91.
[24] G. Matheron, G. Matheron, G. Matheron, G. Matheron, Random sets and integral geometry(1975).
[25] M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool, Face detection without bells and whistles, in: European Conference on Computer Vision (ECCV), 2014, pp. 720–735.
[26] E. Meijering, Cell segmentation: 50 years down the road, IEEE Signal Process. Mag. 29 (5) (2012) 140–145.
[27] P. Mukhopadhyay, B.B. Chaudhuri, A survey of hough transform, Pattern Recognit. 48 (3) (2015) 993–1010.
[28] H. Mutikainen, K. Peltonen, T. Tirri, J. Kayhko, Characterisation of oxygen dispersion in mediumconsistency pulp mixing, Appita J. 67 (1) (2014) 37.
[29] H. Mutikainen, N. Strokina, T. Eerola, L. Lensu, Kälviäinen, J. Käyhkö, Online measurement of the bubble size distribution in medium-consistency oxygen delignification, Appita J. 68 (2) (2015) 159–164.
[30] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1992.
[31] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[32] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks., IEEE Trans. Pattern Anal. Mach. Intell. (2016).
[33] K. Sivakumar, J. Goutsias, Discrete morphological size distributions and densities: estimation techniques and applications, J. Electron. Imaging 6 (1) (1997) 31–53.
[34] N. Strokina, R. Juránek, T. Eerola, P. Zemčík, H. Kälviäinen, L. Lensu, Comparison of appearance-based and geometry-based bubble detectors, in: Proccedings of International Conference on Computer Vision and Graphics, 2014, pp. 610–617.
[35] N. Strokina, J. Matas, T. Eerola, L. Lensu, H. Kälviäinen, Detection of bubbles as concentric circular arrangements, Mach. Vis. Appl. 27 (3) (2016) 387–396.
[36] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[37] J. Šochman, J. Matas, WaldBoost – learning for time constrained sequential detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
[38] P. Zemčík, R. Juránek, M. Musil, P. Musil, M. Hradiš, High performance architecture for object detection in streamed videos, in: International Conference on Field Programmable Logic and Applications (FPL), 2013, pp. 1–4.
[39] S. Zhong, X. Zou, Z. Zhang, H. Tian, A flexible image analysis method for measuring bubble parameters, Chem. Eng. Sci. 141 (2016) 143–153.