

A pre-processing technique to decrease inspection time in glass tube production lines

Gabriele Antonio De Vitis | Pierfrancesco Foglia  | Cosimo Antonio Prete

Dipartimento di Ingegneria dell'Informazione,
Università di Pisa, Via Diotisalvi 2, Pisa, Italy

Correspondence

Pierfrancesco Foglia, Dipartimento di Ingegneria
dell'Informazione, Università di Pisa, Via Diotisalvi 2,
56100 Pisa Italy.
Email: pierfrancesco.foglia@unipi.it

Funding: This work has been partially supported
by the Italian Ministry of Education and Research
(MIUR) in the framework of the CrossLab project
(Departments of Excellence – LAB Advanced
Manufacturing and LAB Cloud Computing, Big data
& Cybersecurity).

Funding information

Italian Ministry of Education and Research; CrossLab
project (Departments of Excellence – LAB Advanced
Manufacturing and LAB Cloud Computing, Big data
& Cybersecurity)

Abstract

In case of glass tube for pharmaceutical applications, high-quality defect detection is made via inspection systems based on computer vision. The processing must guarantee real-time inspection and meet increasing rate and quality requirements. Defect detection in glass tubes is complicated by aspects that hamper the efficiency of state-of-the-art techniques. This paper presents a pre-processing algorithm which excludes portions of the image where defects are surely absent. The approach decreases the time for defect detection and classification phases (any detection algorithm can be applied), as they are applied only in high-probability candidate sub-image. We derive a methodology to get robust values of algorithm's parameters during production. The algorithm relies on detrended standard deviation and double threshold hysteresis, which solve issues related to the misalignment between illuminator and acquisition camera, and enable a robust detection despite rotation, vibration, and irregularities of tubes. We consider Canny, MAGDDA, and Niblack algorithms. The solution keeps the detection quality of such algorithms and reaches a 4.69× throughput gain. It represents a methodology to obtain defect detection in time-constrained environments through a software-only approach, and can be exploited in parallel/accelerated solutions and in contexts where a linear camera is utilized on both flat and uneven surfaces.

1 | INTRODUCTION

Glass tubes, to be used as containers for pharmaceutical use such as vials, syringes, and carpules, are typically made of borosilicate glass, which is widely considered the most traditional and effective choice for parenteral drug containers or delivery systems [1, 2]. Such kind of glass can present production defects like knot inclusions (blobs), lines of air, or flexible fragments called lamellae, and these defects can produce subsequent problems and pharmaceutical recalls [1, 3, 4]. Glass issues have been related to numerous product recalls, and over 100 million units of drugs packaged in borosilicate glass containers have been withdrawn from the market [5–7]. Therefore, pharmaceutical glass manufacturers must accurately inspect the quality of the tubes, to accept only the ones having defects, if any, below the threshold that prevents them from degrading the final quality of the product.

An automatic inspection system can be used, based on a machine vision system with a back illuminator (Figure 1). On

average, glass has no patterns and it has monochromic and translucency dominant characteristics. However, glass defects produce variations in refraction properties around them and can cause local variations in light intensity in glass images, which image processing techniques [8] aim to detect.

Machine vision has been successfully adopted for performing inspections in many industrial processes [8, 9–16]. The classical approach consists in extracting the image parts containing the object of inspection [9, 11]. Subsequently, algorithms are applied to identify image regions whose pixels may belong to a defect [8, 14, 17, 18]. The final phase consists of the classification of defects, in which characteristics of the identified regions are derived as to eventually assign defects to predetermined classes.

Current evolution of glass production process involves both high accuracy in defect detection and faster production lines. It is necessary to design new systems aiming at doubling the current production speed and improving current defect-detection quality. Therefore, the throughput

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *IET Image Processing* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

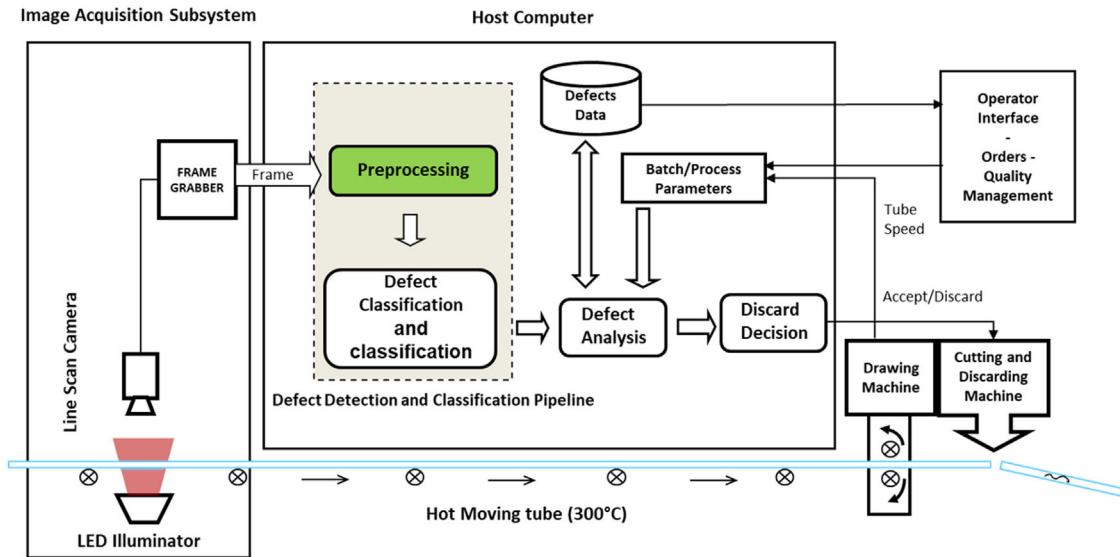


FIGURE 1 The inspection system

requirement of the detection is getting more and more critical over time.

From the vision standpoint of defect detection, we must consider the following major production issues: (a) The cylindrical shape of the glass produces darker areas near the edges of the tube profile image, (b) the tube is partly suspended, it rotates and vibrates and, (c) in some instants, the illuminator is not perfectly centred with the tube, (d) the glass is not perfectly cylindrical but has irregularities (it has a “sausage-like” shape). All these aspects prevent using techniques known in the literature [8, 17–19, 20] effectively.

With the aim of reducing the processing time required by defect detection and classification phases, in this paper we propose a fast preprocessing approach to exclude from detection the portions of the image where no defects are present. To exploit such an approach, we develop an algorithm (DSDRR, Detrended Standard Deviation Region of interest Reduction algorithm), based on detrended standard deviation (DSD) and double threshold hysteresis. The detrended standard deviation detects the image portions that may contain a defect, solving the issues related to alignment between the illuminator and the acquisition camera. Double thresholds with hysteresis allow robust detection of the image portions despite rotation, vibration, and geometric irregularity of the tube. The algorithm relies on parameters that affect the detection quality and throughput. We have refined a methodology, to be applied in the setup process of a new production batch, which determines the values that ensure robustness in the detection at the cost of a minimum temporary degradation of the throughput. These values are then kept constant throughout the full-throughput production batch.

The comparative evaluation of DSDRR has been carried out considering three different detection algorithms available in literature: Canny, MAGDDA, and Niblack. Results in all the cases show that the algorithm keeps the same inspection quality,

reduces the inspection area by 88% on average, and achieves a throughput improvement by at least 1.97 \times and up to 4.69 \times in more computational demanding cases.

The proposal represents a methodology to obtain defect detection in time constrained environment by acting only on software, so it can be extended to application domains where computation and/or energy resources are limited. Differently from other proposal in literature, this technique does not reduce the resolution of the images, which could prevent or severely hamper defect detection. It can also be useful in environments where high-performance resources are available, as it reduces computation time, the communication cost and power consumption.

2 | THE INSPECTION SYSTEM

The most relevant classes of defects for pharmaceutical glass production [9, 21], are the following, and their criticality derives from the size of defect features as well as the considerable impact on the final quality of the product. Specifically:

1. Knot inclusion (Blobs), caused by imperfections in the raw materials used in the furnace; they appear on the tube surface as circular lenses, while they appear on the image with a back illuminator as dark patches (Figure 2).
2. Air lines, caused by air bubbles in the furnace which are pulled by the drawing machine; they appear as long dark lines (Figure 3), having their ends thinner than the central part.

Tubes cannot have air lines in their used section for some pharmaceutical applications, such as in vaccine syringes, because if they are too close to the surface of the tube, they can break (open line), and cause contamination of the content. The blob is related to the cosmetic look of the glass container, and, in the

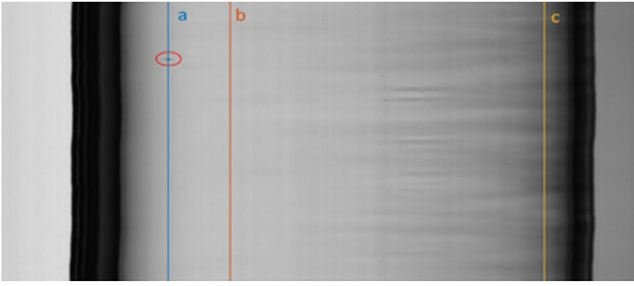


FIGURE 2 Blob defect (enclosed in an oval). The illuminator is not aligned. The right side of the tube shows the shape of sausage, the undulation of the edge shows the effects of vibration

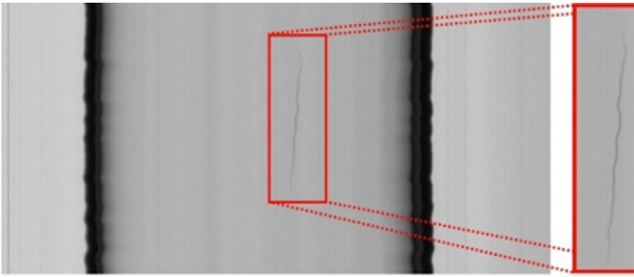


FIGURE 3 Air line defect. The line is straight, but the oscillations and the rotation of the tube give the line a curved and irregular shape

case of vials, can induce in consumers the idea of low quality of the contained product.

Such defects can be detected by an Inspection System, which has two macro-components: Image Acquisition Subsystem and Host Computer (Figure 1, [9]). The Image Acquisition Subsystem acquires the digitized images (frames). It includes a LED-based illuminator, a line scan camera in front of the illuminator, and a frame grabber, which (a) assembles individual sequential lines, captured by the camera, into a whole frame, and (b) transfers the frame to the Host Computer. Our system utilizes three cameras, with one illuminator each, as to cover all 360° of the tube's external surface.

The inspection is performed while the tube is moving from the furnace to the Drawing Machine. The subsequent Cutting and Discarding Machine (CDM) cuts the tube in slices, each about 1.5 m long. The Host Computer implements defect detection and classification algorithms and can send a discard command to the CDM. Discard policies are based on various factors, depending on the production order, and may include the number of air lines and blobs present in a tube slice, their (cumulative) length or area, and, the maximum size of defects.

The inspection task is made particularly challenging by the following factors: (i) The darker areas near the edges of the tube profile; (ii) the rotation and vibration of the tube, (iii) illuminator not aligned with the centre of the tube, (iv) the tube-shaped more like a “sausage” (the right side of Figure 2).

2.1 | Performance requirements

The line scan camera consists of a single array of sensors; each sensor in the line array covers a virtual rectangle on the glass tube. The virtual rectangle represents a pixel in the frame. The virtual rectangle has a dimension along the orthogonal direction of the tube (orthogonal size) that depends on sensor size and optical magnification, as in Equation (1) [22–24]. The dimension along the longitudinal direction (longitudinal size) can be derived from tube speed and line acquisition period, as in Equation (2) [22, 24, 25]. The frame size in the orthogonal direction is given by the orthogonal size of the virtual rectangle and the number of sensors (pixels) in the line array Equation (3) [23, 25].

$$\text{orthogonal size} = \text{sensor size} \times \text{magnification} \quad (1)$$

$$\text{longitudinal size} = \text{tube speed} \times \text{line acquisition period} \quad (2)$$

$$\begin{aligned} \text{orthogonal frame size} &= \text{orthogonal size} \\ &\times \#\text{pixelsinlinarray} \quad (3) \end{aligned}$$

The minimum orthogonal resolution required depends on the tube diameter. Higher resolution is required for smaller diameter tubes due to their higher curvature. For tubes having a diameter less than 30 mm, an orthogonal resolution of $15 \mu\text{m}$ is considered acceptable by experts. These requirements give a lower bound to the number of sensors in the line array. In fact, the number of sensors must be greater than the ratio between the orthogonal frame size and the orthogonal size Equation (4).

$$\#\text{pixelsinlinarray} \geq \text{orthogonal frame size} / \text{orthogonal size}$$

As orthogonal size is $15 \mu\text{m}$ and maximum orthogonal frame size is 30 mm, we obtain that:

$$\#\text{pixelsinlinarray} \geq 30 / 0.015 = 2000$$

Therefore, at least a 2k linear camera must be adopted to satisfy the requirement on orthogonal resolution.

In production, the Drawing Machine controls the tube speed, and, in our scenario, the speed is within the range 0.5–4 m/s (depending on the tube diameter). A value of 2 pixels per mm is considered by experts an acceptable value of longitudinal resolution. To obtain this longitudinal resolution at maximum tube speed (4 m/s), we utilize a 2k CCDs camera, with sensor size of $7 \mu\text{m}$ with a scan frequency of 8 kHz (the corresponding line acquisition period is $125 \mu\text{s}$). According to Equation (1) and (2), the orthogonal size is about $15 \mu\text{m}$ (the size of sensors is $7 \times 7 \mu\text{m}$ and the optical magnification is 1:2.1), and longitudinal size is about 0.5 mm when tube speed is of 4 m/s.

$$\text{orthogonal size} = 7 \mu\text{m} \times 2.1 \gg 15 \mu\text{m}$$

$$\text{longitudinal size} = 4 \text{ m/s} \times 10^{-6} \text{ s} = 0.5 \text{ mm}$$

To guarantee real-time inspection, the system must satisfy the following constraint Equation (5):

$$\text{processing time for a frame} < \text{frame acquisition period} \quad (5)$$

Acquisition period depends on the number of lines in a single frame and the line scan frequency of the camera Equation (6).

$$\begin{aligned} \text{frame acquis. period} &= \# \text{linesinaframe} \\ &\times \text{line acquisition period} \quad (6) \end{aligned}$$

In our case, frames are made up of 1000 lines so, according to Equation (6), the acquisition period for each frame is 125 ms and the processing time per frame must be less than 125 ms (5).

3 | RELATED WORK

In inspection system, each frame is typically subjected to three processing steps [14, 26, 27]: Pre-processing, defect detection and classification.

In the pre-processing phase, the image is elaborated for the subsequent phases, applying techniques for reducing acquisition noise [28] and allowing the acceleration of next phases through the exclusion of zones outside the Region of Interest (ROI, extraction) [29]. In glass tube inspection, ROI extraction consists in detecting the usable part of the tube within the frame, and comprises also the elimination of the tube's edges [9].

The defect detection phase aims at identifying the image regions whose pixels may belong to a defect, which is then classified in the following phase. The detection relies on segmentation techniques [14], which in turn are based on thresholds [8] or edge detection [17, 18]. In classification, a series of characteristics are extracted from the segmented regions, possibly including them within predetermined classes of defects.

Edge detection techniques [17] try to detect points in a digital image where the image brightness changes sharply compared to the surroundings. Canny algorithm [18] represents one of the best solutions for images with noise [17] and the algorithm has been successfully used in various applications domains ([10–13]).

In threshold-based techniques, the threshold can be global (a single threshold for the image) or local (different thresholds for different areas of the image [19]). Many global thresholds techniques are derived from the OTSU algorithm [30] that determines threshold values that maximize the inter-class variance of the image histogram [14]. Otsu algorithm and its variants have been exploited to perform inspection in various application domains [8, 14, 15].

As for local thresholds, the Niblack's [19] binarization method is a local adaptive technique, based on varying threshold over the image by using local mean value and the standard deviation of grey level evaluated in a window centred in each pixel. The application of Niblack has produced good results in

segmenting text from the background and has been used also in vision system for industrial processes [16].

By considering the circular shape of the tube, the detection of defects cannot be obtained with the use of constant thresholds, since the luminous intensities of pixels relative to a defect in an area have values similar to the pixels not related to defects in another area (Figure 6(a)). Besides, tube vibration and the its not perfect circular shape prevent using procedures based on background subtraction or template matching [8, 20]. In [31], a method is proposed for detecting defect in surface textures based on single dimensional local binary patterns, based on two steps. First, in training step, a single dimensional local binary patterns is applied on full defect-less surface images and a basic feature vector is calculated. Then, by image windowing and computing the non-similarity amount between these windows and the basic vector, a threshold is computed for defect-less surfaces. Finally, in the detection step, by using the defect-less threshold the defects are detected on real images. Anyway, by considering that tube is in continuous movement (web inspection) and rotates and vibrates, and that the illuminator is in some instant not perfect centred with the tube, it may be difficult to build an efficient training and detection step in the case of tube inspection. In [9], a Canny-based algorithm has been proposed to perform inspection in a system similar to the ones presented in this paper, and it was able to meet the performance requirements of that time.

Currently, a general trend in this type of image processing applications is the need to reduce the processing time of the algorithms used, due to the time and quality requirements of modern production lines, the increase in resolution required and the availability of high resolution data sources [32]. Image processing algorithms must then perform demanding computation on substantial amount of data, with high throughput, and often, with the need to meet real time requirements [32].

Techniques for reducing processing time and for achieving real-time performance adopted in literature are essentially the following:

- Reduction of the data elaborated by the algorithm
- Design of parallel implementations of the algorithms and execution on parallel machines
- Use of special hardware accelerators like FPGAs

For instance, processing time have been reduced by resizing the image under inspection [33]. The technique has been utilised in [34] to build an automated inspection system of surface defects based on machine vision. The authors utilize a pattern matching technique that is applied to images whose size is reduced via a Gaussian Pyramid decomposition. A similar technique has been utilized in [23] to reduce the computation time of one stage of their processing pipeline. In [33], for defect detection in web offset printing, a one dimensional down-sampling of the image is utilized to speed-up the identification of the first row of web offset images.

Parallel implementations can unleash significant speedups, but specific techniques must be adopted to code effective parallel algorithms that yield similar performance gains as the

invested computational and storage resources [35–37]. Parallel algorithm may be executed on conventional or GPGPU architectures (General Purpose Graphics Processing Units). For instance, Intel OpenCV library contains an optimized version of Canny detector, that exploits multithreading and multicores [38]. Different parallel implementation of Canny and other edge detection algorithms on GPGPU have been proposed [23, 39]. In [23], a high-resolution inspection platform for the detection of surface defects in touch panel glass, has been implemented based on GPGPUs. High-resolution acquisition systems generate large image data, and ad-hoc GPU kernels, that perform all the step of detections, allow achieving acceptable elaboration time. In general, performance of the developed algorithm depends on the size of the acquired image [39, 40], as also highlighted in [23].

An alternative solution to GPGPU is represented by the design of FPGA-based accelerators (Field Programmable Gate Array). A FPGA device is an array of logic blocks that can implement arbitrary parallel and pipelined operations, without the constraint of conventional multicores or GPGPUs [41, 42]. These devices are typically suitable for exploiting the specific parallelism of low-level image processing algorithms [43–45]. There are many implementations of defect and edge detection algorithms in literature [43, 44, 46], but also in the case of FPGA the processing time of an image depends on the image size [44].

As a summary, both in case of conventional, GPGPU or FPGA solutions, image resizing constitutes an important step for satisfying real-time requirements of the application, as it reduces processing time, and also communication costs of parallel solutions.

In our specific application, standard down-sampling techniques [33] cannot be adopted, as the reduced resolution degrades the quality of defect detection below mandatory requirements. The technique proposed in this paper reduces the size of the ROI image by excluding areas where no defects are present, and this approach differs from all other works in literature. A preliminary version of the idea appears in [47], where the effects of ROI reduction on processing resources are discussed.

4 | RATIONAL

In the ROI Extraction stage, two areas are excluded from the image: (i) The ones outside the tube image and (ii) dark areas near the edges of the tube itself. In general, the ROIs may contain large portions without any defect. Application of the detection algorithm on these portions is unnecessary and wastes elaboration time. Within the overall processing time of all the steps, the largest quote (about 77%) is spent in the detection, when Canny algorithm is applied. By applying Canny only to the areas containing defects, the time spent in detection is reduced by about 64%. In principle, if an image contains no defects, removing defect-free columns from the ROI implies that the processing time of defect detection drops to zero. Since our principal purpose is to decrease total processing time, our proposal consists in removing portions from the ROI where it is

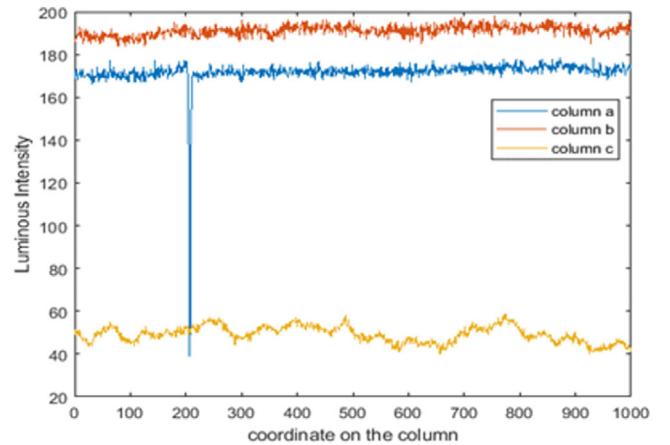


FIGURE 4 Value of luminous intensity along the three columns (a–c) of Figure 2. The presence of a pulse in column (b) is due to the presence of the blob, while the not perfect aligned illuminator emphasizes the irregularities of the tube (column (c))

easy (fast) to assess the absence of defects. We call the remaining ROI area Reduced ROI (RROI).

It's about understanding as identify the areas without defects without spending much time. We can observe that the luminous intensity along a column has small variation (column a, Figures 2 and 4) except in pixels where there are irregularities (column c) or defects (column b). The presence of the blob (column b) produces a significant impulsive variation, while the irregularity of the geometric shape (column c) produces non-impulsive variations of the pixel values. The standard deviation of each column can be used to detect the presence of defects in the column (Figure 5(a)), provided that the effects of non-perfect alignment of the illuminator with the acquisition camera are eliminated. In this case, in fact, there is a growing trend of the standard deviation from a side of the tube to the other (Figure 5(b)). This effect can be eliminated by removing the linear trend values from the values of the standard deviation (Detrended Standard Deviation, DSD).

Another significant requirement is implementing a robust detection strategy as to maintain its quality across various conditions. Rotations and vibrations soften the intensity and curve the shape of lines, making their ends less evident. The same phenomenon occurs in blobs causing a blurred outline along the longitudinal axis. Therefore, the luminous intensity of the defects is high in the proximity of the centre of the defect but tends to drop on the end. However, these final parts are those that determine the true size of the defect. The result is that only the central part produces an impulsive trend of the standard deviation, while the final parts might not be highlighted by the detection process.

To solve such issue, we use a hysteresis with two thresholds as to identify columns that may belong to a defect (t_L and t_H with $t_L < t_H$). The algorithm includes in the RROI: (i) Columns with DSD values greater than t_H , (ii) columns with DSD values between t_L and t_H adjacent to the columns already in the RROI (Figure 6).

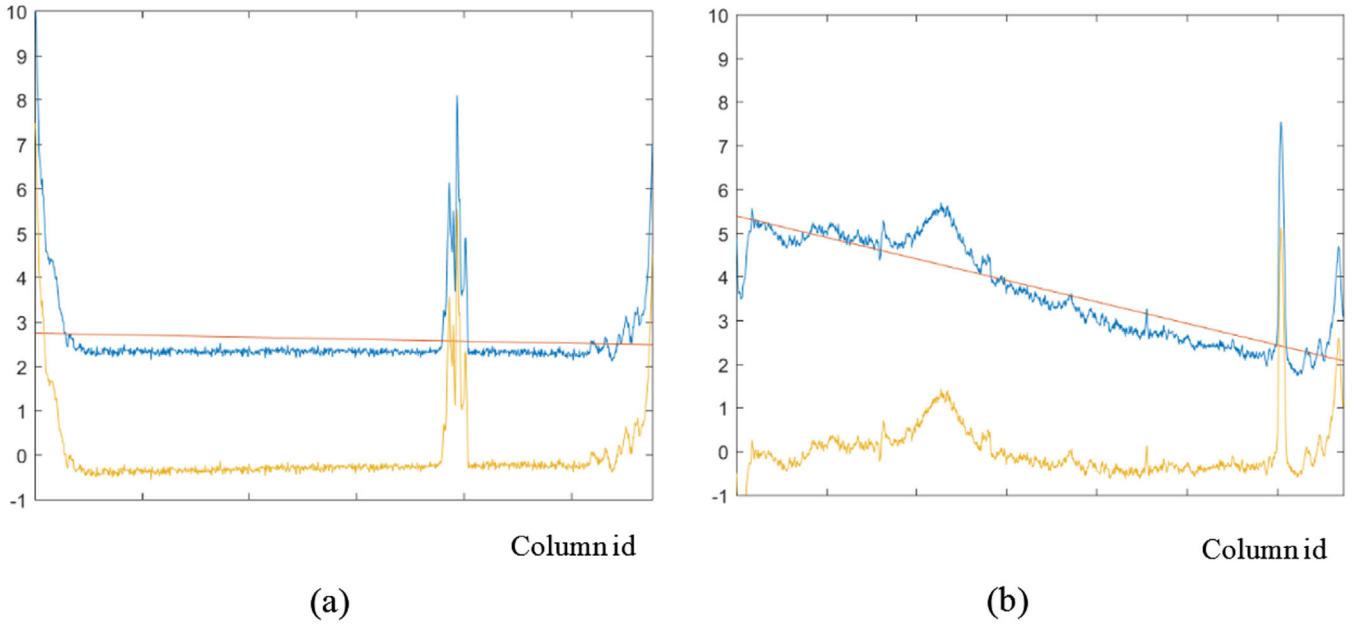


FIGURE 5 (a) Standard Deviation (blue), Trend (orange) and DSD (yellow) of the columns of a frame with an air line defect. (b) Standard Deviation (blue), Trend (orange) and DSD of the columns of a frame with a blob defect

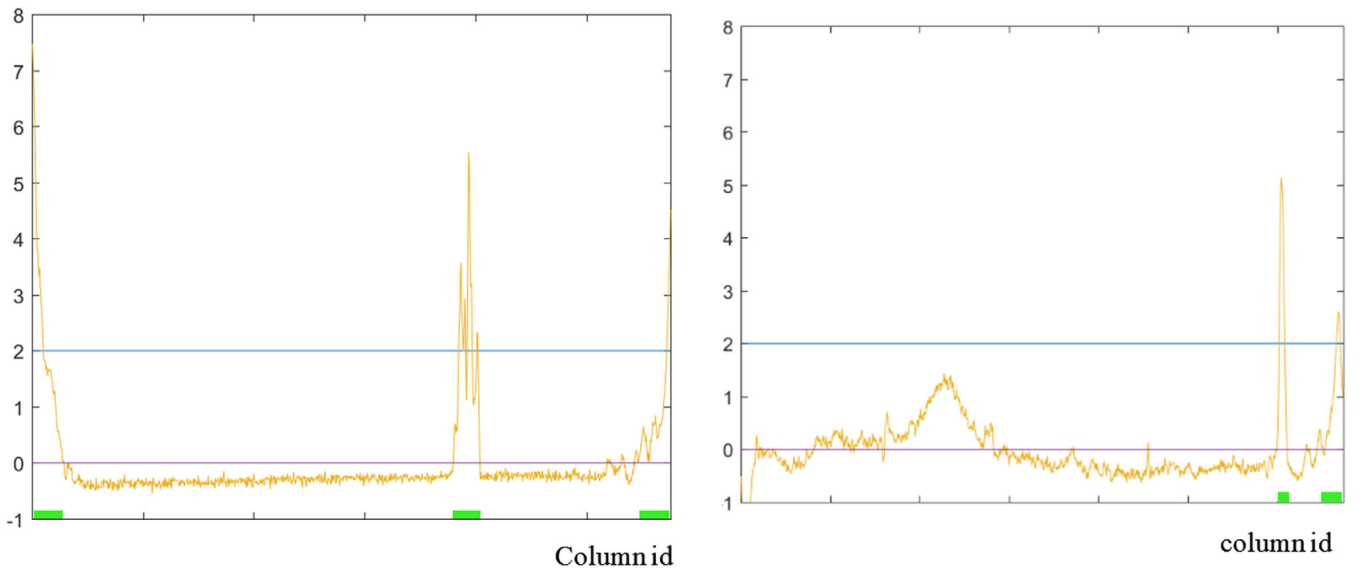


FIGURE 6 Proposed algorithm applied to the frames of Figures 5(a,b). The columns belonging to the resulting RROIs are highlighted in green. The blue line represents the tH value, the red line represents the tL value, the yellow line represents the DSD

Hysteresis with two thresholds has been utilized also in Canny [17], for managing weak edges that are connected to strong edges. The main differences with our approach are: (1) In Canny hysteresis is applied to the gradient magnitude of a pixel, while, in our approach, it is applied to the DSD of a column; and (2) points that pass the “hysteresis test” in Canny are marked as edges, while in our approach if a column passes the “hysteresis test” it does not mean that some pixels of the column belong to the defect. The column is indeed included in the RROI, and then processed in the next step of defect detection to determine if a defect is present on the RROI.

In [48, 49] is presented an image based automatic road crack detection method, in which image areas belonging to normal road surface and crack road surface are distinguished by using image sample variance (where a sample is a window of the image), and only image samples belonging to crack road surface are further elaborated for crack detection, improving also the accuracy of detection [49]. The variance is calculated for the set of sample windows that constitute the image, and a threshold is applied to detect the samples that include cracks. Unlike these works, in our proposal we utilized DSD of a column together with hysteresis. This is because data on a column are acquired by a single sensor, so it is expected that the acquired signal along

```

1. function adjacent_rroi_column(v,i,t,dir)
2.   while((i+dir>=1)&&(i+dir<=v.length())&&(v[i+dir]>=t))
3.     i=i+dir;
4.   end while
5.   return i;
6. end function
7.
8. function rroi_extraction(input_image, tL, tH)
9.   internal_image=extract_internal(input_image);
10.  std = std_column(internal_image);
11.  ds = detrend(std);
12.  RROI_list = empty list;
13.  i=1;
14.  while (i<=ds.length())
15.    if(ds[i]>tH)
16.      left_col= adjacent_RROI_column (ds, i, tL, -1);
17.      right_col= adjacent_RROI_column (ds, i, tL, +1);
18.      if ((left_col!=1) && (right_col!=ds.length()))
19.        insert into RROI_list the columns < left_col, right_col >
20.      end if
21.      i= right_col;
22.    else
23.      i=i+1;
24.    end if
25.  end while
26.  return RROI_list
27. end function

```

FIGURE 7 Proposed ROI reduction algorithm (DSDRR)

is almost constant for a defect free column, but, as stated above, we have to manage the effect of non-perfect alignment of the illuminator, and, due to rotations and vibrations, only the central part of the defects produces an impulsive trend of the standard deviation, while the final parts might not be highlighted by the detection process, that instead must be detected. The use of the image sample variance with a threshold could be not adequate, as end parts of the air lines or some portion of the blob may result below the threshold in this case, while they are marked as “weak defective column” by our algorithm. Besides, the aim of our technique is to improve the execution time, and not to improve the quality of detection.

5 | ALGORITHM

The algorithm proposed for defect detection (DSDRR, DSD Roi Reduction) computes the DSD of each column of the internal part of the tube (ROI). Afterward, the columns whose DSD values are greater than t_H and the ones adjacent to the RROI columns whose DSD values are greater than t_L are assigned to the RROI.

The algorithm logic is shown in Figure 7. The acquired frame is the input_image and the other inputs are the two thresholds (t_L and t_H). The algorithm returns a list of intervals of adjacent columns that represents the image RROI.

For simplicity, the code of some functions is not reported: Function extract_internal (input_image) extracts the internal part of the tube from the input_image using the algorithm proposed in [9]; std_column (image) returns the array of standard deviation values calculated for each column of the input

image; detrend (input_array) calculates the difference between the input_array and its linear trend by applying the linear least squares method; length() member function returns the number of components of an array. Function adjacent_RROI_column (v, i, t, dir) finds the adjacent columns that belong to a RROI (i.e. their DSD is higher than t) given a column belonging to the RROI. Given an array v , an index i , a threshold t and a direction dir (1 for right and -1 for left), the function moves, from the index i and with the direction dir, on the array until a value lower than the threshold t is found. It returns the last index, that is, the last column in the direction dir that meets the condition v [column] $> t$.

The algorithm DSDRR (function rroi_extraction) performs the following steps:

- extracts the internal part of the image, invoking function extract_internal and assigns it to the variable internal_image (classic ROI, line 9);
- calculates the standard deviation for columns of the internal_image and assigns it to variable std (line 10);
- calculates the DSD of the image and assigns it to the array ds = detrend(std) (line 11);
- scans ds starting from index $i = 1$ until $ds[i]$ is higher than threshold t_H (lines 13–14). Column i belongs to the RROI;
- starting from i , it determines the columns adjacent to i that belong to the same RROI using threshold t_L . This is performed by invoking adjacent_RROI_column(v, i, t, dir) with parameters ds, i , t_L and dir (in the left and right direction, dir = -1 and dir = 1). Returned values are assigned to variables left and right (lines 16–17);
- the interval is added to the RROI_list only if left and right are not the indexes of the first or the last component of ds (lines 18–19) – this is an optional feature of the algorithm;
- updates index i setting it to right value (as columns from i to right have been already inspected) and repeats d) until the end of the array (line 21);
- returns the list of the columns belonging to the RROI (RROI_list) (line 26).

6 | ALGORITHM PARAMETERS

Threshold values of the proposed algorithm affect the quality of detection and throughput: in a real production environment there is the need to identify the values that provide the best quality and maximum throughput. An inappropriate choice of thresholds can exclude defects from the RROI and generate false negatives, or it can widen the RROI which induces loss of performance. To ensure that all defects belong to the RROI, threshold t_H must be lower than at least a DSD value of the column of each defect. Therefore, the threshold t_H must be chosen lower than the maximum value of the DSD of the columns of each defect, otherwise the algorithm may exclude areas that include defects. Too low values of t_H imply the inclusion in RROI of areas that do not include defects, resulting in unneeded higher computational load and thus decreased performance.

```

1. Set mDSD = <empty>
2. Set MDSD = <empty>
3. for each frame
4.   ROI is extracted
5.   DSD is calculated
6.   Elaboration and classification algorithms are applied
7.   for each classified defect
8.     Set D= DSD value of columns on which the defect is located
9.     M=max(D);
10.    Insert M into Set MDSD
11.    m = min(D);
12.    Insert m into Set mDSD
13.  end for
14. end for
15.  $t_{H\text{UPP}} = \min(\text{MDSD})$ 
16.  $t_{L\text{UPP}} = \min(\text{mDSD})$ 
17. Return <  $t_{L\text{UPP}}, t_{H\text{UPP}}$  >

```

FIGURE 8 The algorithm for determining upper bound to t_L and t_H

To ensure that the entire shape of each defect belongs to the RROI, and in order not to limit the quality of detection in terms of size, threshold t_L must be lower than the minimum value of the DSD of the columns of each defect. Too low values of t_L imply that areas without defects are included in RROI, resulting again in performance loss.

In conclusion, the values of t_H and t_L must satisfy the conditions:

$$t_L \leq \min(\text{minDSD_defects_columns}) \quad (7)$$

$$t_H \leq \min(\text{maxDSD_defects_columns}) \quad (8)$$

Many parameters related to the production batch (glass tube diameter, size, opacity, thickness, etc.) influence the choice of the optimal thresholds. Since these parameters change with different productions, we have defined a method that determines the best values for the thresholds depending on each condition. The method is performed at the change of production and consists in the collection of image frames during this phase and for each frame performs the following steps:

- (i) detects and classifies defects by a standard technique.
- (ii) for each defect it calculates the maximum and minimum values of the DSD of all the columns belonging to the defect and adds them to the MDSD and mDSD sets.
- (iii) determines the bounds (upper): $t_{L\text{UPP}} = \min(\text{mDSD})$ and $t_{H\text{UPP}} = \min(\text{MDSD})$.

The algorithm is described in Figure 8: in the initialization (line 1–2) MDSD and mDSD are set to empty. For each collected frame (line 3), the algorithm extracts the ROI (line 4), the DSD is evaluated (line 5) and the defect detection and classification algorithm is applied (6). For each classified defect (line 7), the DSD values of the defect columns are assigned to D (line 8), the minimum (m) and maximum (M) value of D are calculated and stored into MDSD and mDSD (line 9–12). The algorithm returns $t_{L\text{UPP}} = \min(\text{mDSD})$ and $t_{H\text{UPP}} = \min(\text{MDSD})$ (line 15–17).

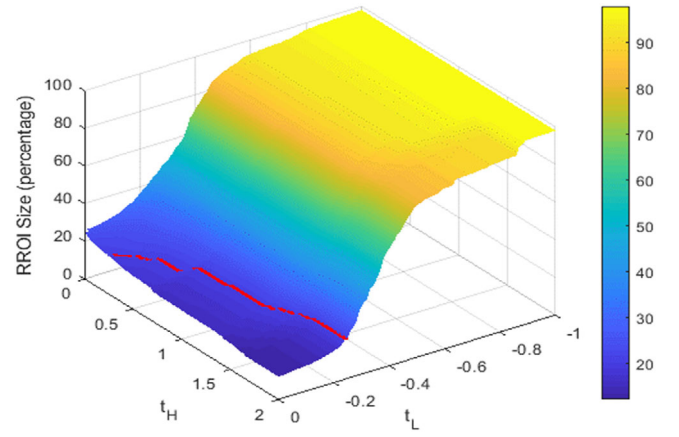


FIGURE 9 3D graph showing the sum of the RROI sizes in percentage (in terms of number of columns) applying the DSDRR according to t_L and t_H . The best value is obtained with $t_H = 2$ and $t_L = 0$ (optimal threshold), for which the RROI is reduced to 12.13% of the whole ROI. The red line represents combinations of t_H and t_L values for which the RROI size is equal to 20% of the whole ROI size

Such values represents the best value for t_L and t_H , so the DSDRR algorithm can work with such thresholds, that is, $t_L = t_{L\text{UPP}}$ and $t_H = t_{H\text{UPP}}$. We must observe that the tuning algorithm always identifies optimal values for the thresholds, since during the change of the production batch the glass tube is in a transitory state which determines the generation of a large number of defects (and in this phase the tubes are always discarded). When the phase ends, the tube is in a steady state and there is no need to change the thresholds until the next batch.

However, it is extremely useful to have a more robust choice of thresholds as to avoid false negative ($t_H < t_{H\text{UPP}}$) and to improve the quality of defect size ($t_L < t_{L\text{UPP}}$), even if it produces a limited reduction in the maximum throughput. So in the following, we characterize the dependence of the RROI size on t_H and t_L (for values lower than the upper bound identified by the algorithm) and we evaluate, in addition to the optimal solution obtained by the algorithm, also three cases of more relaxed thresholds.

Figure 9 shows the percentage of the RROI size determined by DSDRR for the experimental set of images described in Section 7.1, when varying the t_L and t_H parameters below the $t_{L\text{UPP}}$ and $t_{H\text{UPP}}$ bounds. Each image has a resolution of 2048×1000 pixels, and the Canny algorithm is applied for defect detection. Figures 10 and 11 show the relationship between each threshold and the size of RROI, while maintaining the other threshold fixed: results highlight that the size of RROI is particularly sensitive to t_L .

Table 1 shows the average processing time for (a) identifying the RROI (RROI column in Table 1, which includes the processing time to extract the ROI and the processing time to detect the RROI), (b) for defect detection (column Detect Detection in Table 1) and classification (column Class in Table 1) and (c) the maximum total processing time (i.e. the sum of the previous components, evaluated as explained in Section 7.1), in case of the optimal thresholds (UB). Results are also shown for 3 “relaxed” thresholds on the red line depicted in

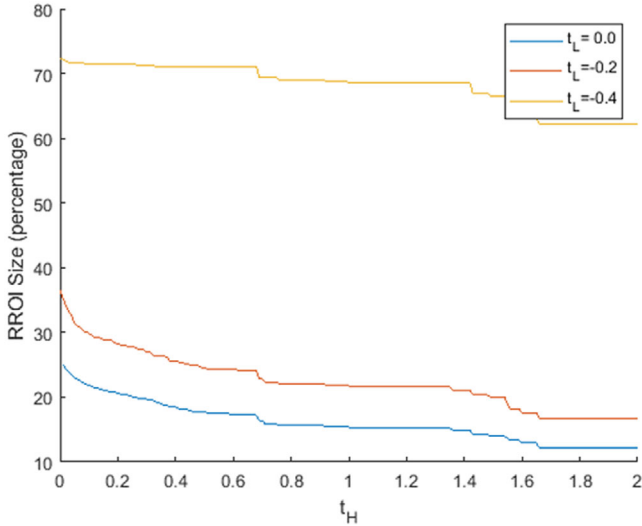


FIGURE 10 RROI size when changing t_H , for $t_L = 0, -0.2, -0.4$

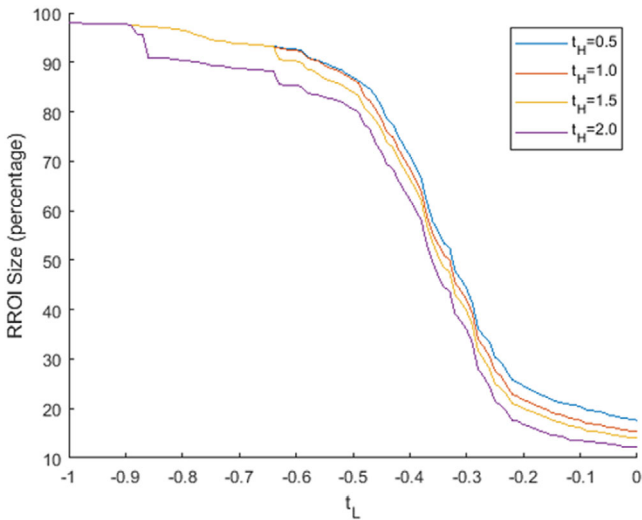


FIGURE 11 RROI size when changing t_L , for $t_H = 0.5, 1, 1.5, 2$

Figure 9. Choice A is more robust in measuring size ($t_L < t_{LUPP}$), C is useful to minimize false negatives ($t_H < t_{HUPP}$). B is a compromise solution ($t_L < t_{LUPP}$ and $t_H < t_{HUPP}$). FPS results underline the cost of more robust setups in terms of slightly reduced performance in case of the three “relaxed” solutions.

TABLE 1 Processing time (ms) and throughput

	RROI	Defect detection	Class.	Maximum total processing	FPS
UB $t_H = 2.03, t_L = 0.0$	10.217	20.403	3.945	37.514	26.657
A $t_H = 2.03, t_L = -0.23$	10.548	29.582	4.511	46.485	21.512
B $t_H = 0.69, t_L = -0.14$	10.667	30.522	4.597	47.714	20.958
C $t_H = 0.27, t_L = 0.01$	10.958	29.578	4.544	49.689	20.125

7 | RESULTS

7.1 | Experimental setup

The prototype inspection system of Figure 1 has been implemented and tested in a glass tube foundry. The image acquisition subsystem is composed by a Basler Racer RaL2048 linear camera [50], and a Matrox Solios eCL/XCL-B Camera Link frame grabber with PCI-X and PCIe connections [51]. Illumination is obtained with a COBRA Slim LED Line based on red light [52] in backlight configuration [53]. The illuminator guarantees a uniform source of illumination so that changes in the environmental conditions determine no effects on the light on the sensors. Three sensors, illuminators and cameras compose the system in order to reliably cover all 360° around the tube under inspection.

The Host computer is a PC based on an Intel i7-940 CPU at 2.93 GHz. The algorithm has been implemented in OpenCV [38] and the communication between the Host Computer and the Cutting and Discarding machine relies on a CIFX 50E-DP board [54].

Evaluations are made using a dataset of 30 frames of glass tubes captured in production. Each frame is composed of 1000 lines taken by the line scan camera sensor with 2K pixels. The dataset contains ten impurities (blobs) and six air lines; one frame contains two air lines, another three impurities. 17 frames have no defects.

The evaluation is performed according to four metrics: processing time, number and type of defects detected, pixel-length of air lines and blob area. The evaluation of processing times has been obtained by executing $400\times$ the complete dataset of images [55]; in each test, frames are randomly chosen, and three processor cores (each one devoted to a camera) perform the elaboration on images preloaded in RAM to reproduce real working condition and interference. We have measured the average processing time of each step (ROI extraction and preprocessing – ROI, which may include also the DSDRR preprocessing depending on the experiment, as explained in Section 7.2, Defect detection, Defect classification and the maximum total processing time (in the following MTPT) and these can be assumed as an estimate of the Worst-Case Execution Times [55].

For the number and type of defects, we considered the number of true positives (TP), that is, the number of defects that are properly classified by the algorithm; the number of false positives (FP), that is, defect detections not corresponding

TABLE 2 Algorithms used in the evaluation

Image pre-processing stage		
	Name	Features
Algorithms to extract the ROI	Without DSDRR	Extraction of the classical ROI according to [9]
	DSDRR	Extraction of the classical ROI according to [9] and its reduction via DSDRR with $t_L = 0.0$, $t_H = 2.03$
Defect Detection Stage		
	Name	Parameters
Algorithms for defect detection	Canny	Thresholds = (35, 80)
	MAGDDA	Windows Size (WS) = 145, Threshold $k = 10.7$
	Niblack	Windows Size (WS) = 20, Threshold $k = -1.7$
Classification Stage		
Algorithm	Connected-components labelling [56] based	

to real defects; and the number of false negatives (FN), that is, actual tube defects that are not recognized. Real defects (expected value) have been identified by human experts through a visual inspection of the tubes and constitute the ground truth.

7.2 | Considered algorithms and configurations

We have applied proposed DSDRR to three algorithms: Canny [18], MAGDDA [57], and Niblack's [19] (Table 2). In case of Canny, a Gaussian filter is also applied to smooth the image and then the magnitude of gradient at each pixel location is calculated. Pixels belonging to a defect are determined by non-maximum suppression, double thresholds and hysteresis. MAGDDA works at row level and uses a moving average filter with windows size WS. A fixed threshold (k) is applied to label the pixel. Niblack algorithm uses local thresholds, calculated as the local average of values, multiplied by a factor (k), the standard deviation of luminous intensity measured in a square window (WS size) around each pixel.

To evaluate the effectiveness of the technique, we considered two configurations (Table 2). In the first configuration (Without DSDRR), the ROI is extracted with the algorithm described in [9] and one of the three defect detection algorithms is applied to it. In the second configuration (DSDRR), after extracting the ROI with the technique [9], the preprocessing technique DSDRR is utilized to determine the RROI (Reduced ROI), to which the defect detection algorithms indicated above are applied.

In the classification phase, an algorithm connects adjacent labelled pixels using [56] and calculates, for each group, the smallest rectangle containing them. Defect classification is straightforward: (i) The ratio between height and base of the rectangle containing a defect, and (ii) the area size allows to classify the type of defect.

7.3 | Evaluations

The results show that the accuracy of Canny and MAGDDA algorithms, in terms of defect detection, does not change with the application of the DSDRR algorithm (Table 3). For Niblack, the introduction of the DSDRR algorithm improves detection as Niblack detects FP in areas removed from the RROI by DSDRR.

The DSDRR algorithm dramatically reduces the size of ROI. DSDRR detects only 17 RROIs: 14 of them contain defects and one contains three defects. RROI is on average the 2% of the original ROI, and at most the 18% and at least the 0% (in frames without defects).

Table 4 and Figure 12 show the processing time and the throughput obtained by applying the considered algorithms, with or without the application of DSDRR in the ROI elaboration preprocessing step. In Table 4, the best and worst execution times for each processing step are highlighted in green and red, respectively. A smaller RROI size decreases the required processing time in the core phases of elaboration (detection and classification) and on the total processing time (Figure 12 and Table 4). Moreover, for frames that do not include defects, the time for detection and classification is, in practice, zero.

Average detection time of Canny with no DSDRR preprocessing is around 62 ms while for Canny with DSDRR this execution time is about 11 ms (−82%). For MAGDDA, the average detection time with no DSDRR preprocessing is 8.33 ms while is 1.44 ms with DSDRR (−83%). For Niblack, the average detection time with no DSDRR preprocessing is 2.3 s while with DSDRR is about 551 ms (−76%). Relevant enhancements are reported also for the classification phase, due to the exclusion of noisy pixels from the RROI.

DSDRR algorithm increases the processing time (Table 4) for determining ROI (from 7.6 to 10.912 ms). Despite this, the MTPT in Canny goes from 89.8 to 29.7 ms (the corresponding throughput increases by 3.02×), and for MAGDDA from 33.9

TABLE 3 Defects/defective frames detected and classification

Expected value		BlobsTP/FP (FN)	Air linesTP/FP (FN)	Defective FramesTP/FP (FN)
		10 Blobs	6 Air Lines	13 Defective Frames
Canny	Without DSDRR	10/5 (0)	6/0 (0)	13/2 (0)
	DSDRR	10/5 (0)	6/0 (0)	13/2 (0)
MAGDDA	Without DSDRR	10/6 (0)	6/0 (0)	13/2 (0)
	DSDRR	10/6 (0)	6/0 (0)	13/2 (0)
Niblack	Without DSDRR	10/11 (0)	6/0 (0)	13/4 (0)
	DSDRR	10/6 (0)	6/0 (0)	13/2 (0)

TABLE 4 Processing time (ms) and throughput (FPS)

Algorithm		Average Execution Time				
		ROI	Defect detection	Class	MTPT	FPS
Canny	Without DSDRR	7.845	61.538	9.395	89.824	11.133
	DSDRR	10.331	10.437	2.554	29.745	33.619
MAGDDA	Without DSDRR	7.695	8.333	8.114	33.941	29.463
	DSDRR	10.331	1.442	2.446	17.270	57.904
Niblack	Without DSDRR	7.934	2,323.891	50.606	2,642.169	0.378
	DSDRR	10.912	551.255	33.165	563.865	1.773

to 17.2 ms (the corresponding throughput increases by 1.97X). For Niblack, the MTPT decreases from 2.6 s to 563.8 ms, inducing a 4.69x throughput increase.

In the previous evaluations we excluded from each ROI the regions near the edges of the tube, as they are inspected by the adjacent cameras. We have evaluated also the scenario in which DSDRR analyses and includes such areas, obtaining higher processing times (the RROI size usually increases), without altering the quality of inspection. The number of RROI areas becomes 70 (instead of 17 areas) and RROI average size increases from

2% to 12%, with a maximum of 31% and a minimum of 4%, instead of 0%. The throughput loss is about 21% for Canny, 13% for MAGDDA, 30% for Niblack.

Results of the evaluation highlight that DSDRR maintains the defect detection quality of each algorithms and in some cases can even improve it. Besides, it is effective in reducing the inspected area, inducing a throughput increase by at least 1.97X and up to 4.69X. On average, the benefits in throughput are higher for more complex algorithms (e.g. Niblack achieves the highest increase).

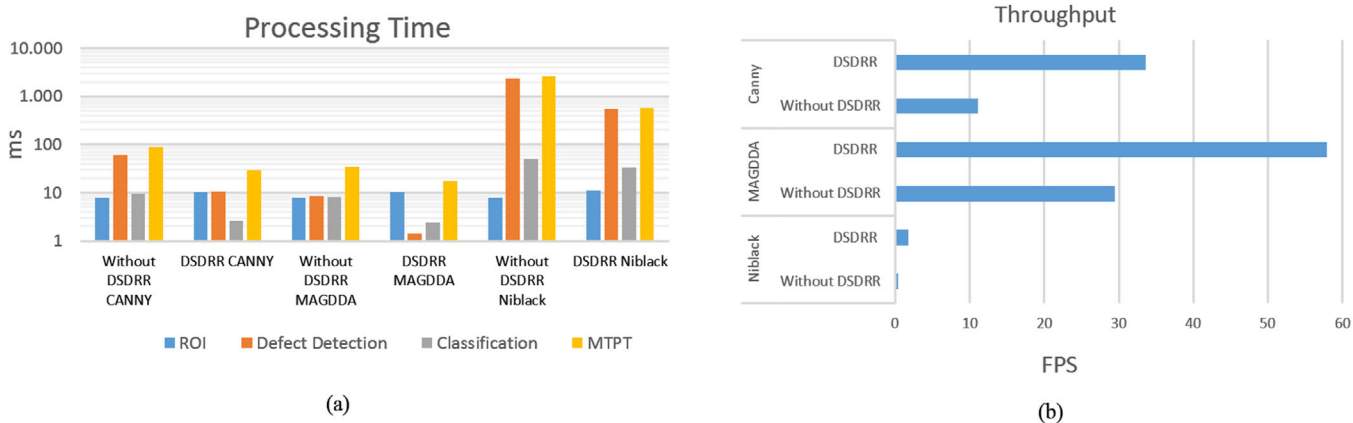


FIGURE 12 (a) Processing time and (b) throughput. In (a) ROI is the processing time (average) spent in determining the ROI/RROI, defect detection and classification represent the average processing times spent in the defect detection and classification stages. MTPT is the maximum total processing time experienced in the experiments. In the DSDRR configurations, ROI includes the processing time consumed to determine the ROI and to apply the DSDRR algorithm (Section 7.2)

The requirement of doubling the tube speed in the future production, while keeping the longitudinal size unchanged, imposes a scan frequency of 16 kHz, that is, a processing time/frame not exceeding 62.5 ms.

Results show that Canny algorithm, with DSDRR technique, can fulfil the performance targets of such scenario. Furthermore, also MAGDDA with DSDRR can keep up with the performance requirements, while doubling both speed and precision.

8 | CONCLUSIONS

The quality of pharmaceutical glass tubes can be checked in real-time during production by a vision system. Improvements in the production process and the need to improve accuracy in detection determine the need for fast solutions. The largest part of the elaboration (about 77%) is spent in defect detection and at least 64% of this time is spent looking for defects in defect-free areas. Our proposal is to reduce the ROI size, by removing the zones without defects through an additional fast-preprocessing phase relying on Detrended Standard Deviation (DSD) and a double threshold with a hysteresis. DSD compensates the effects of not perfect alignment of camera and illuminator, while the double threshold algorithm solves the issues related to vibration and rotation of the tube. The obtained algorithm (DSDRR) is effective in reducing the image area, by 98% on average, and in reducing the time required for defect detection and classification phases.

The algorithm has operating parameters that affect the quality of detection and throughput. We have tuned a methodology for identifying tradeoff values that ensure robust detection and optimal throughput and can be adopted during the setup process of a new production run.

DSDRR is orthogonal to the detection algorithm and so can be used in conjunction with any of them. We have applied DSDRR to three different state-of-the-art algorithms: Canny, MAGDDA, and Niblack. Results of the experimental evaluation suggest that DSDRR preserves the detection quality and significantly reduces the computational needs of the entire system. The only overhead resides in a slight increase in the RROI calculation time. The overall throughput is improved by at least 1.97× up to 4.69× in the most computationally demanding cases. As a general remark, reduction in processing time potentially allows either increasing of the detection quality by using higher sample-rate cameras or increasing the production speed without loss of precision and accuracy. Canny algorithm, with DSDRR preprocessing, fulfils the performance requirement deriving from doubling tube speed or doubling the resolution. MAGDDA with DSDRR can sustain the doubling of both.

With our proposal, the requirements imposed by the evolution of production processes are solved at software level. The proposal represents a methodology to obtain defect detection also in time-constrained environments, so it can be potentially extended also to other application domains where computation resources are limited. Likewise, it can be useful in environments where high performance resources are available,

because it can decrease computational effort and/or power consumption.

DSDRR approach can be utilized also in other application environments where a linear camera is used on both flat and uneven surfaces. As for future works, we are planning to apply it to satellite vision systems, where hardware resources can be quite constrained both in terms of computational capabilities and energy envelope. Another potential application of the proposed algorithm is in content based image retrieval systems, as a preprocessing technique to be used in the object cropping phase [58].

ACKNOWLEDGMENTS

This work has been partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence – LAB Advanced Manufacturing and LAB Cloud Computing, Big data & Cybersecurity).

ORCID

Pierfrancesco Foglia  <https://orcid.org/0000-0001-6432-4504>

REFERENCES

- Schaut, R.A., Weeks, W.P.: Historical review of glasses used for parenteral packaging. *PDA J. Pharm. Sci. Technol.* 71(4), 279–296 (2017)
- Berry, H.: Pharmaceutical aspects of glass and rubber. *J. Pharm. Pharmacol.* 5(11), 1008–1023 (2011)
- Reynolds, G., Peskiet, D.: Glass delamination and breakage, new answers for a growing problem. *BioProcess Int.* 9(11), 52–57 (2011)
- Iacocca, R.G., Tolt, N., Allgeier, M., et al.: Factors Affecting the Chemical Durability of Glass Used in the Pharmaceutical Industry. *AAPS PharmSciTech* 11(3), 1340–1349 (2010)
- Food and Drug Administration. Market withdrawals, & safety alerts search. <https://www.fda.gov/Safety/Recalls/> (2019). Accessed 23 March 2021
- Bee, J.S., et al.: Effects of surfaces and leachable on the stability of biopharmaceuticals. *J. Pharm. Sci.* 100, 4158–4170 (2011)
- DeGrazio, F., Paskiet, D.: The glass quandary. *Contract Pharma*, January 23 (2012): 2012-01
- Peng, X., et al.: An online defects inspection method for float glass fabrication based on machine vision. *Int. J. Adv. Manuf. Technol.* 39(11–12), 1180–1189 (2008)
- Foglia, P., Prete, C.A., Zanda, M.: An inspection system for pharmaceutical glass tubes. *WSEAS Trans. on Syst.* 14, 123–136 (2015)
- Shankar, N.G., Zhong, Z.W.: Defect detection on semiconductor wafer surfaces. *Microelectron. Eng.* 77(3–4), 337–346 (2005)
- Adamo, F., et al.: A low-cost inspection system for online defects assessment in satin glass. *Measurement* 42(9), 1304–1311 (2009)
- Huang, X., et al.: An online technology for measuring icing shape on conductor based on vision and force sensors. *IEEE Trans. Instrum. Meas.* 66(12), 3180–3189 (2017)
- de Beer, M.M., et al.: Bubble formation in co-fed gas–liquid flows in a rotor-stator spinning disc reactor. *Int. J. Multiphase Flow* 83, 142–152 (2016)
- Martínez, S.S., et al.: An industrial vision system for surface quality inspection of transparent parts. *Int. J. Adv. Manuf. Technol.* 68(5–8), 1123–1136 (2013)
- Huai-guang, L., et al.: A classification method of glass defect based on multiresolution and information fusion, *The Intn. J. Adv. Manuf. Technol.* 56(9), 1079–1090 (2011)
- Kim, C.M., et al.: Development of auto-seeding system using image processing technology in the sapphire crystal growth process via the Kyropoulos method. *Appl. Sci.* 7(4), 371 (2017)

17. Kumar, M., Saxena, R.: Algorithm and technique on various edge detection: A survey. *Signal Image Process.* 4(3), 65 (2013)
18. Canny, J.F.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8(6), 679–698 (1986)
19. Saxena, L.P.: Niblack's binarization method and its modifications to real-time applications: A review. *Artif. Intell. Rev.*, 51, 673–705 (2019)
20. Kong, H., et al.: Accurate and Efficient inspection of speckle and scratch defects on surfaces of planar products. *IEEE Trans. Ind. Inf.* 13(4), 1855–1865 (2017)
21. Berry, H.: Pharmaceutical aspects of glass and rubber. *J. Pharm. Pharmacol.* 5(11), 1008–1023 (2011)
22. Steger, C., Ulrich, M., Wiedemann, C.: *Machine Vision Algorithms and Applications* (2nd ed.). Wiley, Weinheim (2018)
23. Chang, M., Chen, B.-C.: Jacque Lynn Gabayno & Ming-Fu Chen development of an optical inspection platform for surface defect detection in touch panel glass. *Int. J. Optomechatronics* 10(2), 63–72 (2016)
24. Dalsa, T.: Understanding Line Scan Camera Applications. white paper, (2014). Available at https://www.imveurope.com/sites/default/files/2069_TD.LineScanApp_whitepaper.v4.pdf, Accessed 23 March 2021
25. <https://www.stemmer-imaging.com/en-dk/knowledge-base/cameras-technical-details-of-line-scan-cameras/visited> (2021). Accessed January 2021
26. Kumar, A.: Computer-vision-based fabric defect detection: A survey. *IEEE Trans. Ind. Electron.* 55(1), 348–363 (2008)
27. Malamas, N., et al.: A survey on industrial vision systems, applications and tools. *Image Vision Comput.* 21(2), 171–188 (2003)
28. Li, D., Liang, L.-Q., Zhang, W.-J.: Defect inspection and extraction of the mobile phone cover glass based on the principal components analysis. *Int. J. Adv. Manuf. Technol.* 73(9–12), 1605–1614 (2014)
29. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., Sebastopol, California (2008)
30. Otsu, N.: A threshold selection using an iterative selection method. *IEEE Trans. Syst. Man Cybern.* 9, 62–66 (1979)
31. Fekri-Ershad, S., Tajeripour, F.: A robust approach for surface defect detection based on one dimensional local binary patterns. *Indian J. Sci. Technol.* 5(8), 1–7, (2012)
32. Gentsos, C., et al.: Real-time canny edge detection parallel implementation for FPGAs. 2010 17th IEEE International Conference on Electronics, Circuits and Systems, Athens, pp. 499–502 (2010)
33. Szeliski, R.: *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, Berlin (2011)
34. El-Agamy, M., Awad, M.A., Sonbol, H.A.: Automated inspection of surface defects using machine vision. *Proceedings of the 17th International AMME Conference*. Cairo, Egypt (2016)
35. Grama, A., et al. *Introduction to parallel computing*. Pearson Education, London, England (2003)
36. Kambatla, K., et al.: Trends in big data analytics. *J. Parallel Distrib. Comput.* 74(7), 2561–2573 (2014)
37. Bartolini, S., Foglia, P., Prete, C.A.: Exploring the relationship between architectures and management policies in the design of NUCA-based chip multicore systems. *Future Gener. Comput. Syst.* 78, 481–501 (2018)
38. <https://opencv.org/>
39. Luo, Y., Duraiswami, R.: Canny edge detection on NVIDIA CUDA. 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, Alaska, pp. 1–8 (2008)
40. Emrani, Z., Bateni, S., Rabbani, H.: A new parallel approach for accelerating the gpu-based execution of edge detection algorithms. *J. Med. Signals Sensors* 7(1), 33 (2017)
41. Che, S., et al.: Accelerating compute-intensive applications with GPUs and FPGAs. 2008 Symposium on Application Specific Processors, Anaheim, CA, pp. 101–107 (2008)
42. Cope, B., et al.: Performance comparison of graphics processors to reconfigurable logic: A case study. *IEEE Trans. Comput.* 59(4), 433–448 (2010)
43. Jiang, J., Liu, C., Ling, S.: An FPGA implementation for real-time edge detection. *J. Real-Time Image Process.* 15(4), 787–797 (2018)
44. Xu, Q., et al.: A distributed canny edge detector: algorithm and FPGA implementation. *IEEE Trans. Image Process.* 23(7), 2944–2960 (2014)
45. Bailey, D.G.: Image processing using FPGAs. *J. Imaging* 5(5), 53 (2019)
46. Trieu, D.B.K., Maruyama, T.: Real-time color image segmentation based on mean shift algorithm using an FPGA. *J. Real-Time Image Process.* 10(2), 345–356 (2015)
47. De Vitis, G.A., Foglia, P., Prete, C.A.: A technique to reduce the processing time of defect detection in glass tubes. *Advances in Intelligent Systems and Computing*, 997, pp. 165–178 *Proceedings of the Computing Conference*, London, UK (2019). https://doi.org/10.1007/978-3-030-22871-2_13
48. Premachandra, C., et al.: Image based automatic road surface crack detection for achieving smooth driving on deformed roads. 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, pp. 4018–4023 (2013)
49. Premachandra, C., et al.: Road crack detection using color variance distribution and discriminant analysis for approaching smooth vehicle movement on non-smooth roads. *Int. J. Mach. Learn. Cyber.* 6, 545–553 (2015)
50. Basler Racer Camera Specification, available at <https://www.baslerweb.com/en/products/cameras/line-scancameras/racer/ral2048-48gm/>, Accessed 23 March 2021
51. Matrox Solios Frame Grabber Specification, available at <https://www.matrox.com/en/imaging/products/components/framegrabbers/solios-ecl-b>, Accessed 23 March 2021
52. COBRATM Slim Linescan Illuminator datasheet, available at <https://www.prophotonix.com/led-and-laser-products/led-products/led-line-lights/cobra-slim-led-line-light/>, Accessed 23 March 2021
53. Pernkopfa, F., O'Leary, P.: Image acquisition techniques for automatic visual inspection of metallic surfaces. *NDT & E Int.* 36(8), 609–617 (2003)
54. PC Cards cifX User Manual available at <https://www.hilscher.com/products/product-groups/pc-cards/pci/cifx-50-reeis/>
55. Abella, J., Padilla, M., et al.: Measurement-based worst-case execution time estimation using the coefficient of variation. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* 22(4), 1–29 (2017)
56. Grana, C., Borghesani, D., Cucchiara, R.: Optimized block-based connected components labeling with decision trees. *IEEE Trans. Image Process.* 19(6), 1596–1609 (2010)
57. De Vitis, G.A., Foglia, P., Prete, C.A.: Row level algorithm to improve real-time performance of glass tube defect detection in the production phase. *IET Image Proc.* 14(12), 2911–2921 (2020)
58. Tajeripour, F., Saberi, M., Ershad, S.F.: Developing a novel approach for content based image retrieval using modified local binary patterns and morphological transform. *Int. Arab J. Inf. Technol.* 12(6), 574–581 (2015)

How to cite this article: De Vitis G.A., Foglia P., Prete C.A. A pre-processing technique to decrease inspection time in glass tube production lines. *IET Image Process.* (2021); 1–13. <https://doi.org/10.1049/ipr2.12186>.