A technique to reduce the processing time of defect detection in glass tubes

Gabriele Antonio De Vitis, Pierfrancesco Foglia, Cosimo Antonio Prete

Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italia
gabrieleantonio.devitis@ing.unipi.it, foglia@iet.unipi.it,
antonio.prete@unipi.it

Abstract. The evolution of the glass production process requires high accuracy in defects detection and faster production lines. Both requirements result in a reduction in the processing time of defect detection in case of real-time inspection. In this paper, we present an algorithm for defect detection in glass tubes that allows such reduction. The main idea is based on the reduce the image areas to investigate by exploiting the features of images. In our experiment, we utilized two algorithms that have been successfully applied in the inspection of pharmaceutical glass tube: Canny algorithm and MAGDDA. The proposed solution, applied on both algorithms, doesn't compromise the quality of detection and allows us to achieve a performance gain of 66% in terms of processing time, and 3 times in term of throughput (frames per second), in comparison with standard implementations. An automatic procedure has been developed to estimate optimal parameters for the algorithm by considering the specific production process.

Keywords: Defect detection, glass tube production, real time inspection, image processing, inspection systems.

1 Introduction

The evolution of the glass production process requires both high accuracy in defects detection and faster production lines.

The general schema of inspection systems for semi-finished glass production, based on machine vision [1][2][3][4], is constituted by the Image Acquisition Subsystem and the Host Computer (Fig. 1). The Image Acquisition Subsystem is devoted to the acquisition of the digitized images (frames); key components of such system are a LED-based illuminator, a line scan camera, and a frame grabber, which groups together single sequential lines captured by the camera into a single frame, transferring it to the Host Computer. The Host Computer implements defect detection and classification algorithms (in the Defect Detection and Classification subsystems). Discard decisions, sent [5] to a Cutting and Discarding Machine, are taken considering process parameters, some of which are settled via a high usable operator GUI [6]. Defect detection and classification algorithms usually are applied only to a part of the acquired image (Region Of Interest – ROI), as only a portion of the image represents the semi-finished glass.

The inspection system works in pipeline, and the Image Acquisition Subsystem feeds the pipeline at a rate which is determined by the sampling rate of the line scan camera divided by the number of lines in a frame. The Defect Detection and Classification module must work with the same rate, to avoid frame loss, i.e., the sampling rate of the line scan camera enforces an upper bound to the processing time of defect detection and classification algorithms. The current requirement of increasing the production speed involves the use of line scan cameras with increased sampling rate to keep constant or to improve the accuracy of defects detection. Consequently, the increase of production speed determines the need to reduce the processing time of defect detection and classification algorithms.

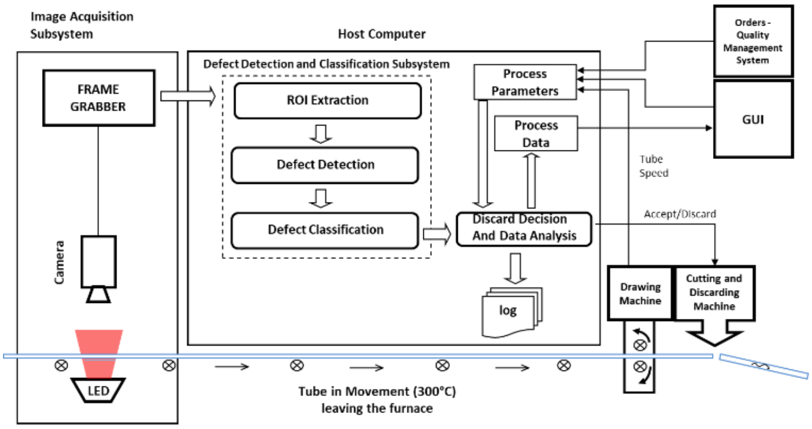


Image Acquisition Subsystem
Host Computer
Orders - Quality Management System
Defect Detection and Classification Subsystem
FRAME GRABBER
ROI Extraction
Process Parameters
Process Data
GUI
Defect Detection
Tube Speed
Defect Classification
Discard Decision And Data Analysis
Accept/Discard
Camera
log
Drawing Machine
Cutting and Discarding Machine
LED
Tube in Movement (300°C) leaving the furnace

Fig. 1. The architecture of an inspection system for glass production.

In this paper, we propose an algorithm to reduce the processing time of defect detection by means of the size reduction of the images to investigate. We easy exclude the subareas that can be assumed to not include defects, by exploiting Detrend Standard Deviation of columns luminous intensity and double thresholding with hysteresis.

We consider the critical production of glass tubes, converted into pharmaceutical containers such as vials, syringes and carpules. In this case, to achieve a 360 degrees inspection, 3 cameras and led illuminators are utilized in the system. Due to imperfections in the raw materials used in the furnace, the glass tube may have defects such as knot inclusions (blobs) or flexible fragments called lamellae, which can cause subsequent problems and pharmaceutical recalls [7][8][9]. The main classes of defects relevant for pharmaceutical glass production [1][10] due to critical size features and their significant effects on the final quality of the tubes are:

1) air lines, due to the presence of air bubbles in the furnace which are pulled by the drawing machine; they appear as darker lines of long dimensions (Fig. 2) with a

back illuminator, with the end parts thinner than the center one. This line, when it is too close to the tube surface, breaks and therefore is thinner and more difficult to detect.

2) knot inclusion (blobs), due to imperfections in the raw materials used in the furnace; they appear on the tube surface as circular lenses, while they appear on the captured image as dark patches, orthogonal to the frame (Fig. 3).

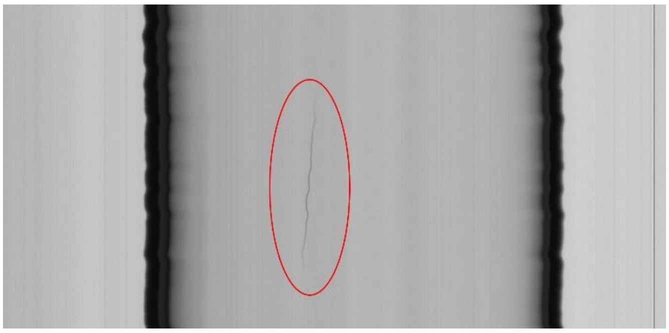


Fig. 2. Example of an air line defect in a glass tube. We must observe that the line is straight but appears as curved and irregular due to the oscillations and rotations of the tube.

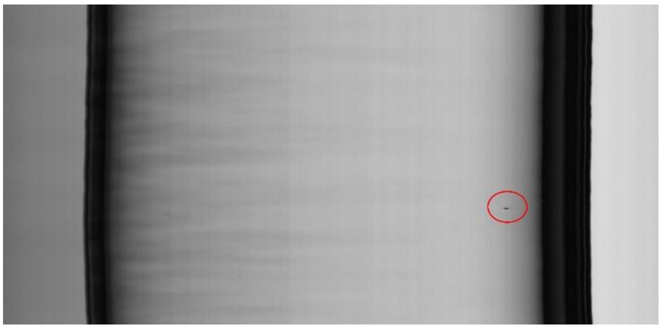


Fig. 3. Example of a blob defect in a glass tube (highlighted by a red oval).

In our experiment, we utilized two algorithms that have been successfully applied in the inspection of pharmaceutical glass tube: the Canny algorithm [11] and the MAGDDA [12]. With Canny, the image is first smoothed with a Gaussian filter and then gradient magnitude is computed at each pixel; edges (marked pixels) are determined via non-maximum suppression, double threshold, and hysteresis. MAGDDA algorithm [12] works at row level and apply a moving average filter of an assigned windows size WS to the image.

The reduced area to be investigated has a direct impact on the processing time of the further stages of detection, and in the overall processing time. Results of the experimental evaluation show that the proposed solution achieves a 66% reduction in processing time during the detection/classification phase and improvement of 3x in term of frames per seconds, preserving the same accuracy in defect detection of standard solutions.

We characterize also the effects of the parameters of the algorithm on processing time and quality of detection and propose a procedure to determine upper bound to such parameters.

2 State of the art

As usually done in inspection systems, the set of elaborations performed on each frame can mainly be divided into 3 stages [2][13][14].

1. Image preprocessing
2. Defects detection
3. Defects classification

In the pre-processing phase, algorithms are used to prepare the image for the following stages, with the aim of reducing detection errors due to the acquisition process and/or speeding up the calculation by excluding the regions not to be investigated. The steps generally adopted concern noise reduction, contrast enhancement [15], elimination of unwanted regions and identification of the Region Of Interest (ROI).

State-of-the-art ROI extraction techniques consist in identifying the visible part of the glass inside the frame. In the case of the glass tube, ROI extraction consists in identifying the visible part of the tube inside the frame (called hereinafter internal part of the tube, Fig. 4), excluding also the edges of the tube. They appear dark as the light rays of the illuminator, having a critical angle of incidence on them according to Snell's law, are reflected on the glass tube and do not affect the camera sensor. An algorithm to extract the internal part of the tube has been proposed in [1], and it is based on detecting the minimums of columns luminous intensity and moving from them to detect not visible part of the tube.

In the defect detection stage, algorithms are used to determine image regions whose pixels may identify a defect. To extract these regions, segmentation techniques are adopted [13], typically based on thresholds [3] or on edge detection [11][16].

The classification stage consists of algorithms that extract a series of characteristics of the segmented regions, eventually including them within predetermined classes of defects.

State of the art techniques for feature/defect detection and extraction are the edge detection techniques [16]. Edge detection aims to identify points in a digital image where the image brightness changes sharply compared to the rest. Among the various edge detection techniques, the algorithm proposed by Canny [11] (Canny algorithm) is considered the ideal one for images with noise [16]. The image is first smoothed with a Gaussian filter and then gradient magnitude is computed at each pixel of the smoothed image; edges are determined by applying non-maximum suppression, double threshold, and hysteresis. The algorithm has been usefully adopted in various applications domains (inspection of semiconductor wafer surface [17], detection of defects in satin glass [18], measuring icing shape on conductor [19], studies on bubble formation in co-fed gas-liquid flows [20]) and it has been also adopted for the detection of defects in glass tube inspection systems [1].

Other techniques for defect detection and extraction are based on thresholds, that can be global (fixed for the whole image) or local, i.e. they can be variable in different regions of the image [21].

As for global thresholds techniques, [3] presents an inspection method for float glass fabrication. The authors utilize a benchmark image to remove bright and dull stripes that are present in their glasses. Then, they utilize adaptive global thresholds based on the OTSU algorithm [22] to separate distortions from defects. The OTSU algorithm selects threshold values (one for each image) that maximize the inter-class variance of the image histogram [23]. It is useful for separating background from defects/foreground and produces satisfactory results when images present bimodal or multimodal histograms [13]. It has been successfully utilized in [13] to derive a configurable industrial vision system for surface inspection of transparent parts (in particular, it has been tested on headlamp lens) and again in [24] to detach defects from the background in a float glass defect classificatory and in [25] or glass inspection vision systems.

By considering the characteristics of the tube glass production, the use of single or multiple constant thresholds does not allow the detection of defects (Fig.6b highlights that the luminous intensity of columns belonging to the defect is similar to the ones of columns not including a defect). Besides, techniques based on background subtraction or other template matching techniques [3][26] cannot be utilized due to the tube vibration and the not perfect circular section of the tube ("sausage" shape).

As for local thresholds techniques, the Niblack's [21] binarization method is a local adaptive thresholding technique, based on varying threshold over the image by using local mean value and the standard deviation of gray level evaluated in a window centered in each pixel. This method can separate the object or text from the background effectively in the areas near to the object. Niblack method is one of the document segmentation methods and has shown good results in segmenting text from the background. Anyway, it can be applied also to images without text [27] and has been applied in a vision system for auto seeding and for observing the surface of the melt in the Ky method for the Sapphire Crystal Growth Process [28].

All these techniques extract a ROI from the acquired image. This extraction removes the part of the image that is not part of the inspection object (for example the portion of image that does not contain the tube, as in [1], or the image of the roller conveyor that is separated from the glass in [18]) but do not use information on defects to further reduce the ROI. Only in [25] is presented a technique that identifies not defective areas within the ROI (background) using a threshold on the local variance calculated in a window centered on a pixel. In the paper, statistics on areas without error are used to automatically calculate global thresholds to perform segmentation, more accurately than OTSU. In our work, we use the knowledge of the no defective areas to reduce the size of the ROI and improve the execution time. Unlike the previous work, in our case, it is not possible to use the standard deviation for the lighting effects. We besides use a double threshold mechanism with hysteresis to further limit the size of the ROI, and we perform columns elaboration as they have an almost constant distribution of luminous intensity, thus avoiding the introduction of further expensive filtering.

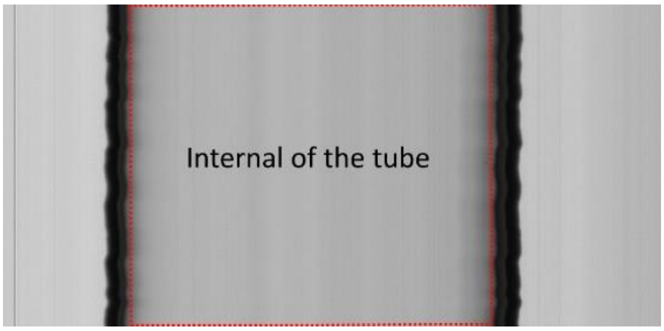


Fig. 4. Image taken by the line-scan camera. The array of CCD sensors in the line scan camera is orthogonal to the direction of the movement of the tube. The internal of the tube is also highlighted. It represents the portion of the image which is further analyzed for detecting defects. The frame is composed of 1000x2048 pixels.

3 Rational

State-of-the-art techniques for defect detection consider the ROI as the input image for their elaboration. Since not all the portions of the ROI include defects, the application of defect detection algorithm to these portions is useless and wastes processing time. Fig. 5 shows the processing time of the 3 stages of detection when the Canny algorithm is applied to a defective image. Bar Canny Whole ROI represents the processing time when Canny is applied to the whole ROI, while Canny Defective ROI represents the processing time when Canny is applied only to the columns of the ROI that includes the defect (defective ROI). As can be seen, it is possible to reduce the processing time of about 60%. As our main requirement is to reduce the overall processing time, our idea is to remove, from the ROI, portions where it is possible to easily predict that no defects are present (reducing the size of the ROI).

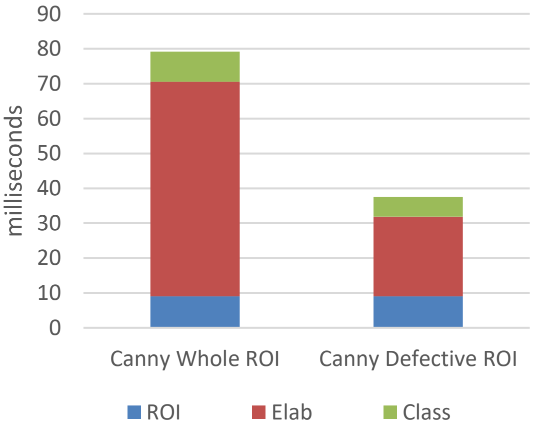


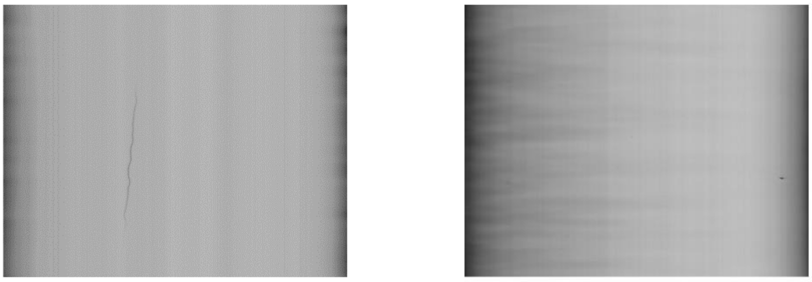
Fig. 5. Processing Time on an Intel i7-940 CPU (2.93 GHz) on a defective image. Canny Whole ROI represents the processing time when the Canny algorithm is applied to the whole

ROI. Canny defective ROI represents the processing time when Canny is applied only to the columns of the ROI that include the defects.

To reduce the size of the ROI, we observe that the luminous intensity inside a column is almost constant except in pixels where there is noise or defects. So, the standard deviation of the column may be used as an indicator of defect presence on that column, as shown in Fig. 6b. Therefore, values of the standard deviation for column below a specific threshold indicate that the column can be excluded by the next processing. Anyway, the standard deviation of the columns in a glass object is influenced by the alignment of the illuminator with the tube. In case of not perfect alignment, usual in glass tube production, the standard deviation shows an increasing trend from one edge of the tube to the other, which can be approximated with a linear trend (Fig. 6d). In order not to have any influence from this factor, we consider the values of the standard deviation of the columns removing their linear trend (Detrended Standard Deviation - DSD).

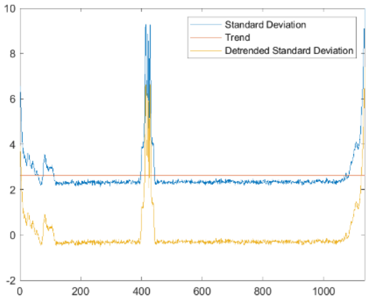
Another relevant requirement concerns the ability to accurately detect the size of defects because it is a parameter for taking discard decisions. We have experimented that luminous intensity of the defects presents high values near the central area of the defects but tends to decrease away from it. Therefore, not all the columns including a defect present high standard deviation value. For these reasons, to detect these columns, we apply two hysteresis thresholds: t_L and t_H with $t_L<t_H$. Columns with DSD values less than t_L are not considered to belong to ROI, columns with values greater than t_H are considered to belong to ROI and columns with values between t_L and t_H are considered to belong to ROI only if they are adjacent to columns that belong to ROI (Fig. 7).

Using the DSD criterion, areas near the edge of the tube are often classified as ROI, as they have peaks of DSD greater than the peaks of the DSD of the columns where defects are placed. These values are caused by many effects as the vibration of the moving tube or the imperfect circular shape of the tube surface. In these areas, there may be defects that if included in the ROI will be recognized by the Defect Detection and Classification Subsystem. Two solutions are then possible. The first is to exclude the ROIs located near the edges. This solution is not destructive because areas near the edges for a camera appear in the central area for one of the other two cameras positioned at 120 degrees from it and the tube. The second solution is to include these areas in the ROI for the next stage of Defect Detection.



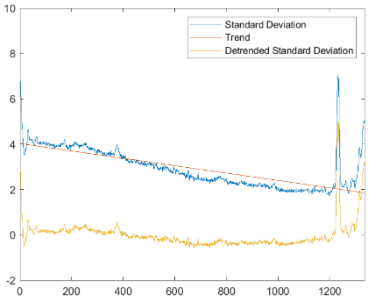
(a)

(c)



(b)

Fig. 6a-b. a) Internal part of the tube including an air line. b) Standard Deviation, Linear Trend and DSD of columns of Fig. a).

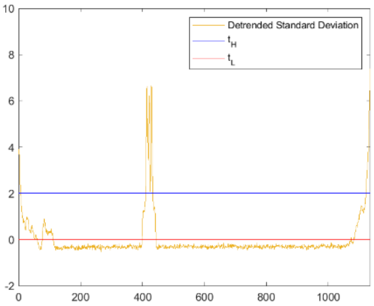


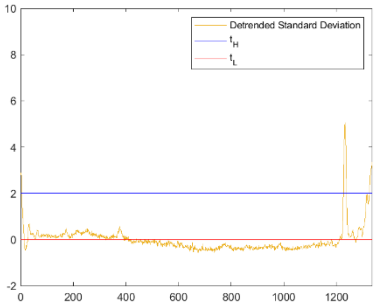
(d)

Fig. 6c-d. c) Internal part of the tube including a blob. d) Standard Deviation, Linear Trend and DSD of columns of Fig. c).

4 Algorithm and choice of the thresholds

The proposed algorithm (Detrended Standard Deviation ROI Reduction algorithm, DSDRR), starting from the image representing the internal part of the tube, calculates for each column the DSD. Then, the algorithm finds the columns whose DSD values are greater than the t_H threshold and promotes all these columns as belonging to the ROI. Next, for each of the columns belonging to the ROI, the algorithm finds the adjacent columns whose DSD values are greater than t_L and promotes also these columns as belonging to the ROI. Finally, if ROI of areas close to the edge of the tube must be excluded from the analysis, the algorithm removes from the ROIs the columns adjacent to the first and the last column.





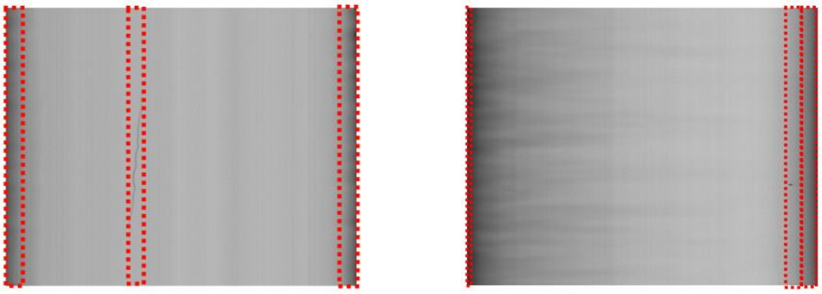


Fig. 7. Application of two thresholds with hysteresis to the DSD of columns of Fig 6a and 6c. Above is shown the ROIs of images (in red boxes) in Figure 6a and 6c as results by applying the proposed algorithm.

The choice of the threshold values determines algorithm performance in terms of detection. A bad choice may cause the exclusion of defects in the ROIs (causing a false negative in detection), or it can generate too large ROI with no reduction of the overall processing time.

The choice of threshold t_H must guarantee that at least one column of a defect belongs to the ROI, i.e. the DSD for that column must be higher than t_H. So, to ensure that all the defects are correctly included in the ROI, the threshold t_H must be lower than the maximum value of the DSD of the columns of each defect. Too high values of t_H can exclude from the ROI areas that include defects, and too low values of t_H, therefore, lead to extremely large ROI, that is, high processing times.

The threshold t_L must guarantee that all the columns of a defect belong to the ROI, otherwise portions of the defect are not detected, therefore, causing the incorrect measurement of the defect size. Too high value of t_L could exclude a portion of the shape of the defect from the ROI, and too low values of t_L could cause again the ROI to include the entire internal area of the tube. A possible solution is to set it to a value lower than the minimum value of the DSD of the columns of each defect. This ensures that, if a defect is detected using threshold t_H, its entire shape is always included in the ROI.

The thresholds t_L and t_H used by the algorithm are related also to properties of defects that depend on production-related parameters (such as glass tube size, diameter, thickness, opacity etc). In an industrial application scenario, these parameters change with batches of production, so it is important to adapt the thresholds to the changes in production. We have developed a procedure that suggests upper bounds for the thresholds that must be adopted when the production parameters change.

The procedure consists of:

1) collecting a number of frames from the acquisition system during the starting phase of a new batch of production,

2) applying an algorithm for the detection and classification of defects to each frame,

3) for each classified defect, calculating the maximum value and minimum value of the DSD of the columns of the defect and adding them respectively to the sets maxDSD and minDSD,

4) the suggested upper bounds are: $t_L < \min(\text{minDSD})$ and $t_H < \min(\text{maxDSD})$.

Table 1. Host Computer.

	Hardware Configuration		
Processor	Intel® Core™ i7-940 Processor (8MB Cache, 2.93 GHz)		
RAM	8 GB		
	Defect detection system		
Algorithm name	Without DSDRR	DSDRR with edges	DSDRR no edges
ROI extraction	Internal part	Internal part + DSDRR including edge with t_L=0, t_H=2	Internal part + DSDRR excluding edge with t_L=0, t_H=2
Defect Detection	Canny (35,80)		MAGDDA (ws = 145, k=10.7)
Implementation	OPENCV		

5 Results

The inspection system of Fig. 1 is working on production lines of a glass tube foundry [1]. The image processing stages have been implemented in a task activated by the frame grabber when a new frame is ready in main memory. Table I summarizes the main features of the Host Computer, the algorithms utilized for the various stages of defect detection and their configuration parameters. All the image processing algorithms have been implemented using the OpenCV library [29] and compiled with the Visual Studio compiler.

System performance has been evaluated using a dataset of 30 frames acquired during the production phase. All frames are composed of 1024 lines acquired by the line scan camera sensor (2K pixels). These frames contain 6 air lines and 10 blobs; one of these frames contains 3 blobs and another one contains 2 air lines while 17 frames do not have any type of defects. The machine on which tests are executed has a configuration similar to the production one and is equipped with an Intel Core i7-940 CPU running at 2.93 GHz. As for processing time, we perform 1000 executions of the whole dataset of images [30], and we take the average execution time spent by each stage (ROI extraction/Defect detection/Defect classification) and the maximum total processing time.

In our experiment, we utilized two algorithms that have been successfully applied in the inspection of pharmaceutical glass tube: the Canny algorithm [11] and the MAGDDA [12]. With Canny, the image is first smoothed with a Gaussian filter and then gradient magnitude is computed at each pixel; edges (marked pixels) are determined via non-maximum suppression, double threshold, and hysteresis. MAGDDA algorithm [12] works at row level and apply a moving average filter of an assigned windows size WS to the ROI. Then applies a fixed threshold (k) to mark the pixel. As for the Defect Classification stage, we adopt an algorithm that groups adjacent marked pixels using connected-components labeling [31] and builds, for each group, the smallest rectangle that contains them. The rectangle features permit to individuate blobs and air lines.

In the tested image-set, the autotuning method suggests for t_H a value less than 2,0362 and for t_L a value less than 0,0126. We set t_H = 2 and t_L = 0. For Canny, we find that the optimal thresholds are 35, 80.

Table 2. Number of recognized defects/Defective Frames and their classification

	Expected value	Without DSDRR DSDRR with edge Canny	DSDRR no edge Canny	Without DSDRR DSDRR with edge MAGDDA	DSDRR no edge MAGDDA
	TP	TP/FP (FN)	TP/FP (FN)	TP/FP (FN)	TP/FP (FN)
Blobs	10	10/5 (0)	9/5 (1)	10/6 (0)	9/6 (1)
Air lines	6	6/0 (0)	6/0 (0)	6/0 (0)	6/0 (0)
Defective Frames	13	13/2 (0)	12/2 (1)	13/2 (0)	12/2 (1)

Table 3. Air line length [pixels]

Frame	Without DSDRR DSDRR with edge DSDRR no edge Canny	Without DSDRR DSDRR with edge DSDRR no edge MAGDDA
1.1	81 / 81 / 81	72 / 72 / 72
1.2	473 / 473 / 473	456 / 456 / 456
2.1	709 / 709 / 709	758 / 758 / 758
2.1	337 / 337 / 337	402 / 402 / 402
2.6	502 / 502 / 502	509 / 509 / 509
3.9	450 / 450 / 450	513 / 513 / 513

With this setting, in terms of defect detection, the accuracy of each defect detection algorithm is the same with or without the application of the DSDRR algorithm when including also the edges of the tube (Table 2 and Table 3 for air lines; the same results are achieved for blob area). When the DSDRR algorithm excludes the edges, there is a false negative due to a defect near the edge of the tube whose columns are excluded from the ROI. Anyway, the system is equipped with three cameras to guarantee a 360 degrees inspection of the tube, and defects near edge of the tube are located by one of the other cameras near the center of the image.

The DSDRR algorithm is effective in reducing the ROI that is processed by the Defect Detection and Classification algorithms. DSDRR calculates 70 ROIs. 53 of these are in the proximity of the edge of tube (and 1 of these contains a defect), 14 contain defects (3 defects are in the same columns/ROI). Compared to the total area, the ROI is reduced on average by 88%, to a maximum of 96% and at least b69%.

Excluding the parts near the edges of the tube, the ROI is reduced on average by 98% and at most by 100% (in frames without defects) and at least by 82%.

The reduced area of the ROI has a direct impact on the processing time of the further stages (Table 4) of detection and in the overall processing time (Fig. 8). As shown in Table 4, the average processing time of Canny without DSDRR is about 61 ms while the execution time of Canny with DSDRR with edges is about 20 ms and 10 ms with DSDRR no edges, with a decrease of 67% and 83%. For MAGDDA, the processing time without DSDRR is 8 ms while with DSDRR with edges is about 3 ms and 2 ms with DSDRR no edges. To further reduce the processing time, these results suggest excluding the investigation of the image edges. Indeed, the system is equipped with three cameras to guarantee a 360 degrees inspection of the tube, and defects near an image edge for a camera are located by one of the other cameras near the center of the image. So, the inspection quality is guaranteed.

Similar improvements can be observed for the processing time of the Classification stage, as noisy pixels that do not belong to DSDRR ROI are not considered by the classification algorithm. The DSDRR algorithm increases the processing time for ROI extraction (from about 8 to 10 ms). Despite this, the total maximum processing time for Canny moves from 90 to 37/30 ms (with a 3x increase in throughput) and for MAGDDA from 34 to 19/17 ms (with a 2x increase in throughput).

Table 4. Average Processing Time of defect detection phases (ms)

		ROI	ELAB	CLASS
Canny	Without DSDRR	7,845	61,538	9,395
	DSDRR with edge	10,217	20,403	3,945
	DSDRR no edge	10,331	10,437	2,554
MAGDDA	Without DSDRR	7,695	8,333	8,114
	DSDRR with edge	10,221	2,817	3,924
	DSDRR no edge	10,331	1,442	2,446

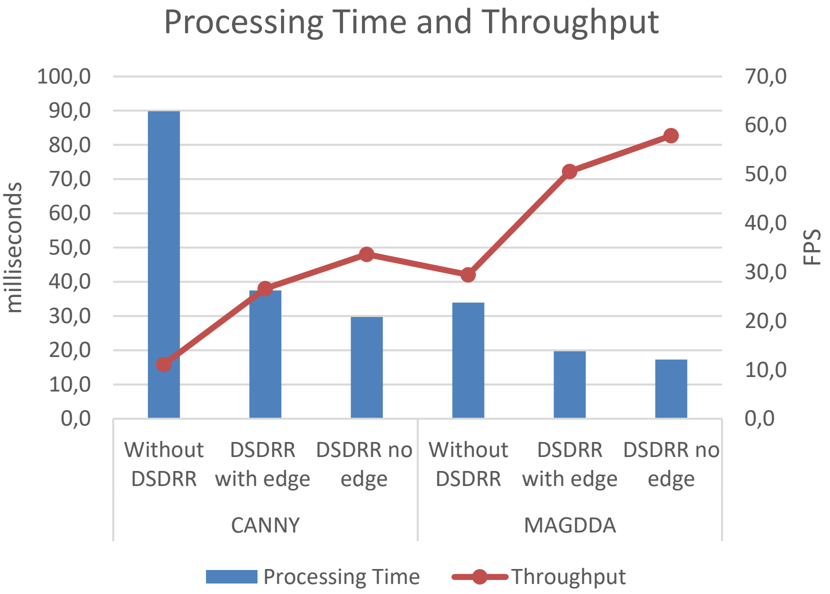


Fig.8. Processing Time and Throughput.

6 Conclusion

A vision system can be exploited to inspect in real-time the glass tube quality during the production process. Improvements in such process and the need to increase the detection accuracy suggest the adoption of a solution that reduces the processing time of all the steps involved in defect detection and classification. A classical approach for dealing with inspection consists in extracting the whole internal part of the tube (ROI) and passing it to the defect detection and classification algorithm. By considering the features of defects and properties that are relevant in the images, we presented an algorithm that further reduces the ROI area by excluding columns that can be determined to not include defects.

We consider the production of glass tubes for pharmaceutical uses. Experimental results indicate that our proposal does not change the quality of detection of the system and significantly improves processing time of both defect detection and classification stages. The overall throughput (frames per second) is improved up to 3 times with respect to standard solutions.

As for future works, we plan to test our algorithm in other application domains and to investigate strategies to parallelize the algorithms by considering CMPs and GPU architectures and related memory systems [32].

7 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).

References

1. P. Foglia, C.A. Prete, M. Zanda, An inspection system for pharmaceutical glass tubes, WSEAS Transactions on Systems, Vol. 14, Art. #12, PP. 123-136, 2015.
2. Kumar, Ajay. "Computer-vision-based fabric defect detection: A survey. "IEEE trans. on ind, electronics 55.1 (2008): 348-363.
3. Peng, Xiangqian, et al. An online defects inspection method for float glass fabrication based on machine vision. The Inter. Journal of Advanced Man. Technology 39.11-12 (2008): 1180-1189.
4. Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc., 2008.
5. S. Campanelli, P. Foglia, C.A. Prete, An architecture to integrate IEC 61131-3 systems in an IEC 61499 distributed solution, Computers in Industry, Vol. 72, Sept. 2015, pp. 47-67.
6. P. Foglia, C. A. Prete, M. Zanda, (2008, May). Relating GSR signals to traditional usability metrics: Case study with an anthropomorphic web assistant. In Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE (pp. 1814-1818).
7. Reynolds G, Peskiest D. Glass delamination and breakage, new answers for a growing problem; BioProcess International 9(11):52-57, 2011.
8. Iacocca R.G., Toltl N., et al.. Factors Affecting the Chemical Durability of Glass Used in the Pharmaceutical Industry. AAPS PharmSciTech, v.11(3):1340-1349, 2010.
9. Schaut, Robert A., and W. Porter Weeks. "Historical review of glasses used for parenteral packaging." PDA journal of pharmaceutical science and technology, 71.4 (2017): 279-296.
10. Berry H. Pharmaceutical aspects of glass and rubber; J. of Pharmacy and Pharmacology, v.5(11):1008-1023; Wiley, 2011.
11. J.F. Canny, A computational approach to edge detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986
12. G.A. De Vitis, P. Foglia, C.A. Prete, “A Special Purpose Algorithm to Inspect Glass Tubes in the Production Phase”, TR-DII-2018-01, University of Pisa, 2018
13. Martínez, S. Satorres, et al. "An industrial vision system for surface quality inspection of transparent parts." The International Journal of Advanced Manufacturing Technology 68.5-8 (2013): 1123-1136
14. Malamas, N., et al. "A survey on industrial vision systems, applications and tools."Image and vision computing 21.2 (2003): 171-188.
15. Li, Di, L. Liang, and W. Zhang. "Defect inspection and extraction of the mobile phone cover glass based on the principal components analysis." The International Journal of Advanced Manufacturing Technology 73.9-12 (2014): 1605-1614.
16. Kumar, Mukesh, Rohini Saxena. "Algorithm and technique on various edge detection: A survey."Signal & Image Processing 4.3 (2013): 65.
17. N.G. Shankar, Z.W. Zhong, Defect detection on semiconductor wafer surfaces, Microelectronic Engineering, 77 (3–4), 337-346, 2005

18. Adamo, Francesco, et al. "A low-cost inspection system for online defects assessment in satin glass." Measurement 42.9 (2009): 1304-1311.
19. X. Huang, et al, An Online Technology for Measuring Icing Shape on Conductor Based on Vision and Force Sensors, IEEE Trans. on Instrumentation and Measurement, 66 (12) 3180-3189, 2017.
20. M.M. de Beer, et al., Bubble formation in co-fed gas–liquid flows in a rotor-stator spinning disc reactor, International Journal of Multiphase Flow, Volume 83, 2016, pp 142-152.
21. Saxena, Lalit Prakash. "Niblack's binarization method and its modifications to real-time applications: a review." Artificial Intelligence Review (2017): 1-33.
22. Otsu N, A threshold selection using an iterative selection method. IEEE Trans Syst Man Cybern (1979) 9:62–66
23. Z. Wakaf, Hamid A. Jalab, Defect detection based on extreme edge of defective region histogram, Journal of King Saud University - Computer and Information Sciences, Vol. 30(1), 2018, pp. 33-40
24. L. Huai-guang et al., A classification method of glass defect based on multiresolution and information fusion, The International Journal of Advanced Manufacturing Technology, Vol. 56 (9), pp. 1079-1090, 2011.
25. M. Aminzadeh, T. Kurfess, Automatic thresholding for defect detection by background histogram mode extents, Journal of Manufacturing Systems, Vol. 37(1), pp. 83-92, 2015.
26. H. Kong, et al., "Accurate and Efficient Inspection of Speckle and Scratch Defects on Surfaces of Planar Products," IEEE Trans. on Industrial Informatics, vol. 13, no. 4, pp. 1855-1865, Aug. 2017.
27. Farid, S., and F. Ahmed. "Application of Niblack's method on images." Emerging Technologies, 2009. ICET 2009. International Conference on. IEEE, 2009.
28. Kim, Churl Min, Sung Ryul Kim, and Jung Hwan Ahn. "Development of Auto-Seeding System Using Image Processing Technology in the Sapphire Crystal Growth Process via the Kyropoulos Method." Applied Sciences 7.4 (2017): 371.
29. https://opencv.org/
30. J. Abella, M. Padilla, et al., Measurement-Based Worst-Case Execution Time Estimation Using the Coefficient of Variation, ACM Trans. Des. Autom. Electron. Syst.22, 4, Article 72 (June 2017)
31. C. Grana, D. Borghesani, and R. Cucchiara, "Optimized Block-based Connected Components Labeling with Decision Trees," IEEE Trans. on Image Processing, vol. 19(6), pp. 1596–1609, 2010.
32. Bartolini, S., Foglia, P., Prete, C. A. (2017). Exploring the relationship between architectures and management policies in the design of NUCA-based chip multicore systems. Future Generation Computer Systems, 78(2), 481-501.