International Journal of Computer Integrated Manufacturing
Vol. 00, No. 00, Month 2009, 1–22

RESEARCH ARTICLE

Visual Inspection of Glass Bottlenecks

Miguel Carrasco[a*], Luis Pizarro[b] and Domingo Mery[a]

[a]*Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860 (143), Santiago, Chile*;
[b]*Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Saarland University, Building E1 1, 66041 Saarbrücken, Germany*

(*Received 00 Month 200x; final version received 00 Month 200x*)

Automated inspection using multiple views has been recently developed to automatically detect flaws in manufactured objects. The main idea of this strategy is that, unlike the noise that appears randomly in images, only the flaws remain stable in a sequence of images because they retain their position relative to the movement of the object being analyzed. In this paper the design of a prototype for image sequence acquisition and inspection of glass bottlenecks using this methodology is proposed. The inspection device considers light sources inside the bottle for better capture of potential defects. Using a single camera we record an image sequence of the bottleneck rotating the bottle along its principal axis. No camera calibration process is considered. We also present a novel approach for defect detection based on tracking potential flaws along multiple views. Our inspection system achieves over a 98% of true positives and 2.4% of false positives rates under uncalibrated image sequences. These results allow us to conclude that uncalibrated images sequences can be used effectively as a mean to evaluate the product quality where the images of the test object do not significantly differ from each other in the sequence.

Keywords: Automated visual inspection; Defect detection; Tracking in multiple views; Uncalibrated images; Glass bottlenecks.

1. Introduction

Visual inspection is defined as a quality control task that determines if a product deviates from a given set of specifications using visual data (Newman and Jain 1995). Inspection usually involves measurement of specific part features such as assembly integrity, surface finish and geometric dimensions. If the measurement lies within a determined tolerance,

*Corresponding author. Email addresses: mlcarras@puc.cl (M. Carrasco)

ISSN: 0951-192x print/ISSN 1362-3052 online

DOI: 10.1080/0951192YYxxxxxxxx
http://www.informaworld.com

the inspection process considers the product as accepted for use. In industrial environments, inspection is performed by human inspectors and/or automated visual inspection (AVI) systems. Human inspectors are not always consistent and effective evaluators of products because inspection tasks are monotonous and exhausting. Typically, there is one rejected in hundreds of accepted products. According to Drury et al. (Drury *et al.* 2004) the human visual inspection is at best 80% effective. In addition, achieving human '100%-inspection', where it is necessary to check every product thoroughly, typically requires high level of redundancy, thus increasing the cost and time for inspection. For instance, human visual inspection has been estimated to account for 10% or more of the total labor costs for manufactured products. For these reasons, in many applications a batch inspection is carried out. In this case, a representative set of products is selected in order to perform inferential reasoning about the total. Nevertheless, in certain applications a '100%-inspection' is required. This is the case of glass bottles fabricated for the wine industry, where it is necessary to ensure the safety of consumers. For this reason, it is necessary to check every part thoroughly.

Defects in glass bottles can arise from an incompletely reacted batch, from batch contaminants which fail to melt completely, from interactions of the melted material with glass-contact refractories and superstructure refractories, and by devitrification. If conditions are abnormal many defects can be produced and even just one defect of only 1-2mg in every 100g article can be enough to give 100% rejection rates. The source identification of these defects can then be a matter of urgency (Parker 2000).

Most of the methods proposed in the literature for automated glass bottle inspection attempt to identify defects in the lips, body and bottom of the bottles (Canivet *et al.* 1994, Firmin *et al.* 1997, Hamad *et al.* 1998, Ma *et al.* 2002, Shafait *et al.* 2004, Wang *et al.* 2005, Yan and Cui 2006, Duan *et al.* 2007, Yepeng *et al.* 2007, Katayama *et al.* 2008). Only few of them deal with the problem of detecting flaws in the bottleneck (Ma *et al.* 2002, Mery and Medina 2004). Nonetheless, that is the bottle part where most of the defects appear during fabrication, due to its narrow and hence difficult to manipulate structure. Encouraged by this difficulty, we focus our research on the inspection of necks in empty glass bottles.

The selection of the lighting for an inspection system is a crucial problem. Since natural lighting conditions are dynamic and change all the time it is not feasible to implement algorithms that are robust to illumination changes without burning important computational time (Kopparapu 2006). Therefore, the use of artificial lighting is a requisite for reaching good and uniform illumination for real-time inspection systems. There exist several studies, e.g. (Yi *et al.* 1995), concerning the placement of external light sources around the object under examination. However, we do not know any work reporting on light sources placed inside a glass bottle. We proposed the design of an electro-mechanical device for image acquisition and inspection of glass bottlenecks using an internal illuminating system. This allows us to obtain high-quality images for capturing very small defects, and to avoid the intrinsic reflections produced by external light sources.

In general, bottle inspection systems can be classified in two categories: Approaches that make use of a single view/camera for detecting flaws, e.g. (Canivet *et al.* 1994, Mery and Medina 2004, Wang *et al.* 2005, Yan and Cui 2006, Duan *et al.* 2007, Yepeng *et al.* 2007); and frameworks that exploit the utilisation of multiple views/cameras to reinforce the detection process, e.g. (Firmin *et al.* 1997, Hamad *et al.* 1998, Ma *et al.* 2002, Shafait *et al.* 2004, Katayama *et al.* 2008). Our proposed inspection device employs a single camera for image acquisition. However, it is able to emulate the use of multiple cameras by recording an image sequence of the bottleneck in successive rotations of the bottle along its principal axis.

Concerning the problem of detecting defects itself, we propose a novel methodology that performs tracking of potential flaws along the acquired image sequence. The key observation is that only real flaws can be successfully tracked, since they do induce spatio-temporal relations between the views where they appear. Conversely, potential defects that cannot be tracked will be considered as false alarms. We effectively track and thus identify real flaws by means of geometry of multiple views (Hartley and Zisserman 2000). In particular, we employ bifocal and trifocal analysis. Although several published methods work with multiple views, this is, to the best of our knowledge, the first work on tracking potential flaws along multiple views for inspecting glass bottles. In this research we have improved previous ideas presented in (Mery and Filbert 2002, Mery and Medina 2004, Carrasco and Mery 2006, Pizarro *et al.* 2008). The main improvements of this paper, apart from dealing with another application and proposing a new system for image acquisition are: (i) the tracking of uncalibrated sequences in objects where there is no clear difference between multiple views, (ii) the design of a novel inner light system to reveal defects more precisely in wine bottlenecks, (iii) the use of a robust feature extractor for defect matching, improving the classical NNDR algorithm (Mikolajczyk and Schmid 2005), and (iv) the introduction of fiduciary markers for computing accurate corresponding points between different views that rotates at the same angle and velocity with the object inspected. This paper also extends and improves a preliminary work (Carrasco *et al.* 2008).

Our paper is organised as follows. In Section 2 we present our prototype for image acquisition and inspection of glass bottlenecks. Section 3 describes our proposed algorithm for tracking real flaws along multiple views. Section 4 shows the performance achieved by our inspection system in comparison with other methods proposed in the literature. Finally, we summarise our contributions and succinctly describe some ongoing and future work in Section 5.

2. Electro-Mechanical System for Image Sequence Acquisition

In this section we describe an electro-mechanical device for image sequence acquisition of the bottleneck of an empty glass bottle. Two are the main components of our mechanism: an internal illumination system and an axle that rotates the bottle during acquisition. The proposed prototype is schematically shown in Fig. 1.

Fig. 1b shows a close-up of the illuminating tube placed inside the bottle. Four LEDs (T1 3.5v-20mA) emitting white light uniformly are located at the bottom of the tube. To improve light uniformity a reflecting layer has been fixed at the other extreme of the tube. To the best of our knowledge, there is no inspection system for glass bottles proposed in the literature that places the illumination system inside the bottle. Typically, bottle inspection systems utilise an illumination plate behind of the bottle conveyor system in order to reveal most defects in front of the inspector. Conversely, our novel design greatly improves the definition of the acquired images, increasing therefore the probability of capturing the smallest defects around the bottleneck. Another important characteristic of the illuminating tube is the set of fiduciary markers situated on both extremes. As we will see later, they allow us to compute accurate corresponding points between different views necessary to inspect the bottleneck in multiples views.

Fig. 2 shows a close-up of the electro-mechanical system that permits rotating the bottle and the light source simultaneously. An image sequence of the bottleneck is thus

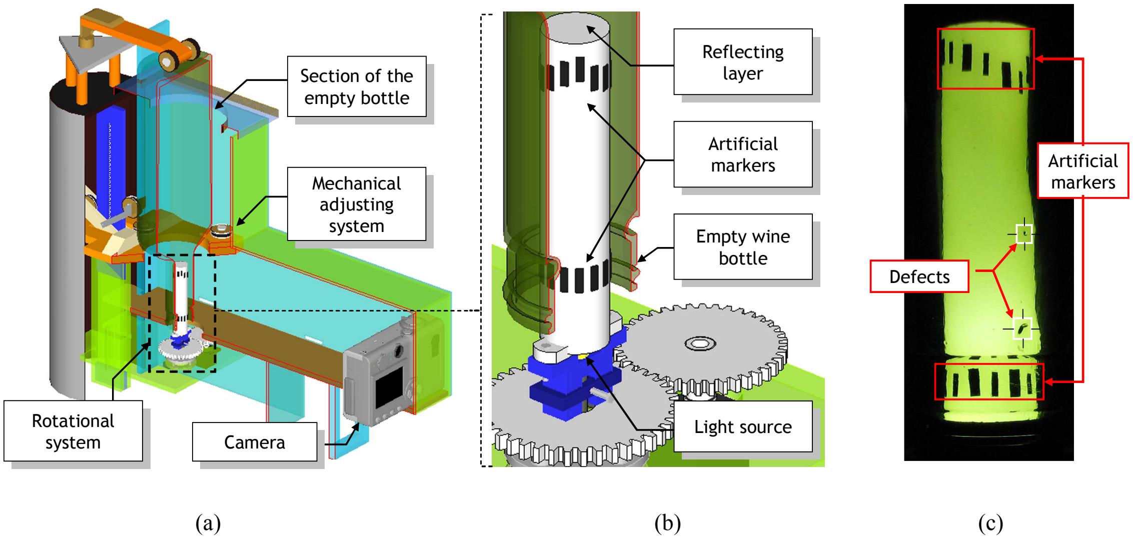


Figure 1. (a) Proposed electro-mechanical prototype for image acquisition; (b) Details of the illuminating tube; (c) Example image captured by the CCD camera.

composed by views taken at successive rotations by a configurable spin angle controlled by a step motor. The images are captured by a standard CCD sensor (see Fig. 1). In our experimental prototype we use a Canon S3 IS camera with resolution 2592 × 1944 pixels and a dynamic range of 24 bits. The camera is placed around 20 cm from the bottleneck. No additional light sources are utilised. The device also includes a mechanical adjusting system to adapt the inspection to different bottleneck lengths. An adjustable arm holds the bottle from its body, and a press mechanism pushes the bottle against the rotor to keep its vertical position.

The electro-mechanical system is commanded by a Basic Stamp micro-controller PIC16C57 connected to a standard personal computer via a RS232 communication port. The micro-controller is programmed in Pbasic(Martin 2005). For a specified spin angle α (in degrees) the micro-controller synchronises the step motor with the illumination system. The camera's acquisition process is triggered by a control system via Matlab. The image sequence consists thus of $\lfloor 360/\alpha \rfloor$ different views, where $\lfloor \cdot \rfloor$ is the floor function.

We have built our prototype considering an upside down bottle similar to (Wang and Asundi 2000), but it is also possible to assemble the system with a right side up bottle. Note that in contrast to several inspection systems that make use of multiple cameras (Firmin *et al.* 1997, Hamad *et al.* 1998, Ma *et al.* 2002, Shafait *et al.* 2004, Katayama *et al.* 2008) the proposed inspection mechanism employs a single camera only. This simplifies the scene's geometry and suffices to build a robust defect detection system by analysing the acquired image sequence. It is important to mention that no camera calibration procedure is considered at all[1]. In the next section we focus on the defect detection process in *uncalibrated* image sequences.

[1] Readers may refer to (Mery and Carrasco 2006) for a further discussion about the calibration problem in industrial environments.

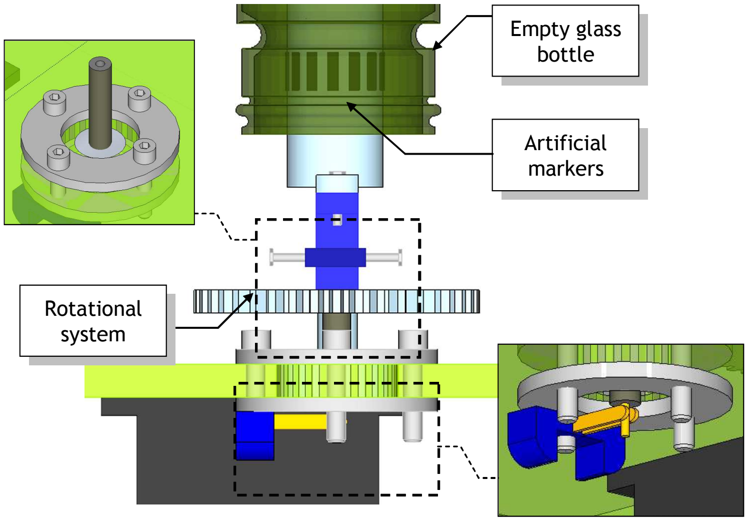


Figure 2. Details of the mechanical system that rotates the bottle and the illuminating tube simultaneously.

3. Detection of Real Flaws by Tracking in Multiple Views

In some applications a unique image might be enough for inspecting certain objects or materials. However, the use of multiple views of an image sequence can reinforce the diagnosis made with a single image. That is the case for example for low signal-to-noise ratio imaging systems, where the identification of real defects with poor contrast entails the appearance of numerous false alarms as well. Here, we aim at exploiting the redundant information present in the multiples views of the bottleneck in order to identify its defects. In fact, only real flaws can successfully be tracked along an image sequence. This is the main idea that will allow us to distinguish real defects from false alarms. Based on such observation, we propose a three-step methodology for detecting real flaws in bottlenecks: segmentation and feature extraction of potential flaws, computation of corresponding points, and tracking potential flaws. Fig. 3 shows a general overview of the proposed approach for image acquisition and defect detection.

3.1 *Segmentation and Feature Extraction of Potential Flaws*

The segmentation of potential defects for every image of the sequence is outlined in Fig. 4. Each image I_0 of the sequence is preprocessed by two independent filters. A noise removal (NR) filter is used for reducing the amount of noise intrinsic to any CCD sensor. For this purpose we employ a Gaussian filter with fixed kernel size (3×3) (Castleman 1996), obtaining an image mask I_1. A structure removal (SR) filter is used for blurring the structures present in the image. In this way, a uniform background image I_2 is obtained. Here we employ a Gaussian filter with kernel size $(n \times n)$, $n \gg 3$[1]. Subsequently, the absolute difference of the filtered images is binarised using the valley-emphasis method

[1] We efficiently compute Gaussian filtering with large spatial supports using the Fast Fourier Transform (FFT).

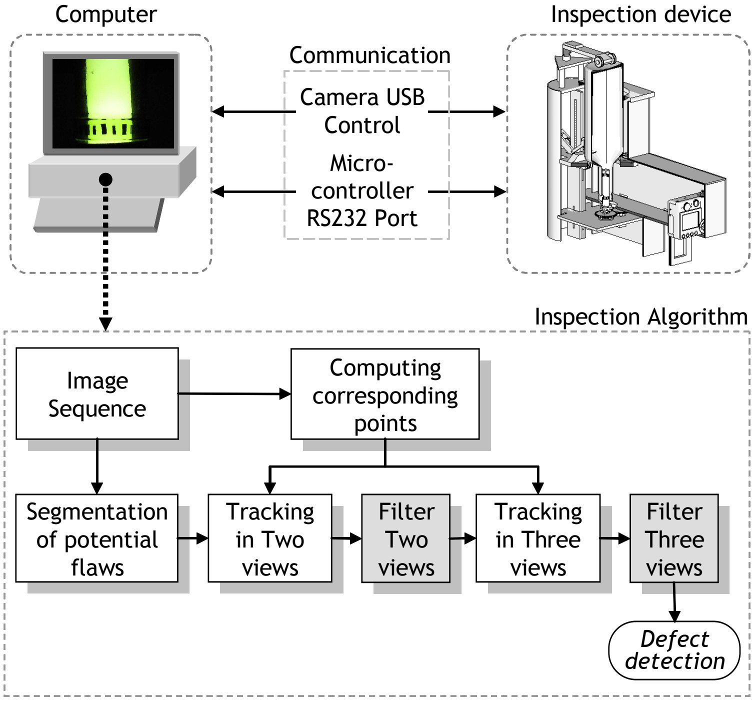


Figure 3. Proposed inspection system for glass bottlenecks.

(Hui-Fuang 2006), obtaining an image mask I_3 with potential defects. As a result, an optimal threshold level is obtaining. Finally, for each potential defect segmented in a region Ω_i a set of invariant measurements are computed and stored in a feature vector $\mathbf{z}^i$, as shown image I_4. Note that the segmentation stage can produce numerous potential defects, though a few (or maybe none) of them correspond to real ones. To maximise the probability of segmenting all real flaws the segmentation process is tunned by the parameter n that reflects the kernel size of the SR filter. Later on in the experimental section we discuss the effect of varying n. In the subsection 3.3 we will detail how potential defects are tracked in two and three different views. Such task is basically based on matching potential defects in the different views, which is highly demanding due to the large number of potential defects generated by the segmentation. The feature vectors of the potential flaws are essential to discriminate real defects from false alarms based on a later feature analysis procedure. This information, combined with the tracking in multiples views can allow us to improve the general tracking process, specially because the uncalibrated nature of the acquired images.

Several invariant features have been proposed in the literature for scalar and colour images (Hu 1962, Flusser and Suk 1993, Flusser *et al.* 1996, Sonka *et al.* 1999, Mindru *et al.* 2004, Lowe 2004, Bay *et al.* 2008). Due to the large difference of the above descriptors in its properties and uses, here we extracted multiple invariant features and a non-invariant features in order to evaluate them, and to compare its performances under multiple combinations as we shall see later. For the sake of completeness, we present a summary of each feature descriptor used in Table. 1.

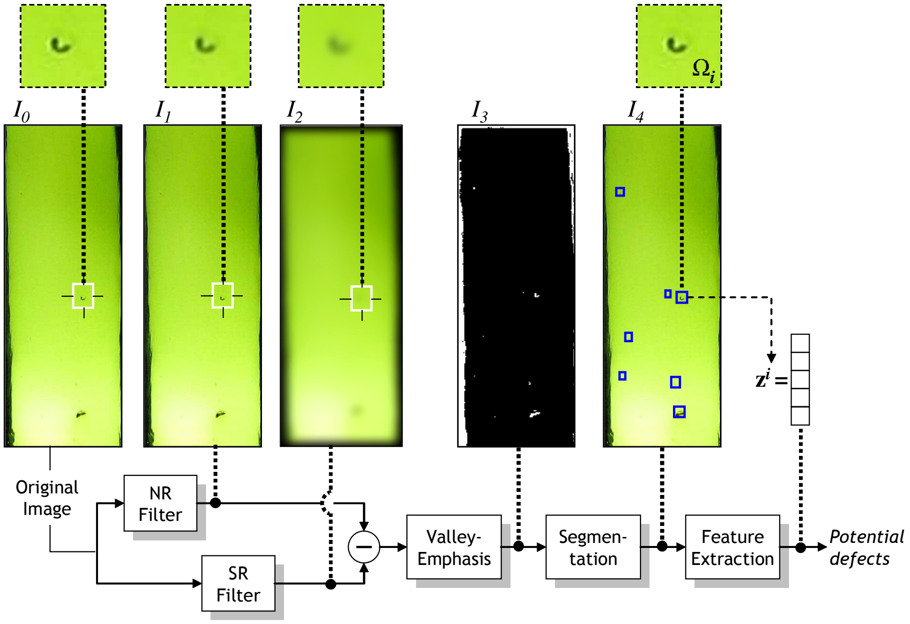


Figure 4. (a) Segmentation and feature extraction of potential defects.

3.2 *Computation of Corresponding Points*

Accurate corresponding points between every pair of views are necessary to match potential flaws along the image sequence. We can easily find corresponding points by placing equidistant fiduciary markers on both extremes of the illuminating source, as showed in Fig. 1. The lower fiduciary markers are positioned at the same vertical level, while the upper ones follow a sinusoidal wave. Using these markers we can compute a set of corresponding points between each pair of consecutive or non-consecutive views. This is schematically outlined in Fig. 5. The upper and lower fiduciary markers of each view are connected through vertical lines between their mass centre, which were also extracted in the segmentation step. Since the length of a vertical line connecting two particular markers remains constant along the image sequence, we know the relative position of these markers in different views. Therefore, the set of corresponding points between two views a and b is conformed by the relative positions of their markers. Such correspondences are later used to estimate the *fundamental matrix* $\mathbf{F}_{a,b}$ that relates any pair of points in the views a and b. Readers may refer to Hartley and Zisserman Hartley and Zisserman (2000) for further details on the fundamental matrix theory. In particular, we estimate $\mathbf{F}_{a,b}$ using a modified version of the algorithm proposed by Chen et al. Chen *et al.* (2000).

3.3 *Tracking of Potential Flaws*

After segmenting and extracting robust features for all potential flaws along the image sequence we now turn to the problem of separating real flaws from false alarms. The key observation is the fact that only real defects could be tracked along the image sequence.

Table 1. Features extracted from the identification step

Feature	Bands	Description
PSO*	Colour	The PSO* feature descriptor is invariant under photometric transformations of type scale and translation based on colour transformations (invariant to illuminations changes)(Mindru *et al.* 2004).
GPSO	Colour	The GPSO feature descriptor is invariant under affine geometric deformations and photometric transformations of type scale and translation based on colour transformations (Mindru *et al.* 2004).
GPD	Colour	The GPD feature descriptor is invariant under photometric transformations and all geometric invariants (Mindru *et al.* 2004).
Flusser-and-Suk	Grayscale	Flusser and Suk proposed a set of moments invariants under affine transformations (Flusser and Suk 1993), derived from second and third-order moments. Since they are designed for grayscale images, a simple grayscale transformation was performed.
Flusser, Suk-and-Saic	Grayscale	In the presence motion velocity or blurred images, Flusser et al., designed a set of 30 moment invariants up to the 7th order (Flusser *et al.* 1996).
Flusser, Suk-and-Saic	Grayscale	The above moments are not invariant to scale and rotated transformations. Therefore, Flusser et al. designed a set of four moments invariant to those transformations based on the normalized central moments and rotation invariants introduced by Hu (Hu 1962).
Hu (Gray)	Grayscale	The Hu feature descriptor is composed by seven moment invariants under rotation, scaling, translation and skew transformations (Hu 1962). Normally it is used for binary images, but it can be used in grayscale images by adding the pixel value for each position.
SURF	Grayscale	The SURF feature descriptor is composed by 64 descriptor invariants under scale and rotation transformations (Bay *et al.* 2008). It is used mainly by its robustness and speed against variations in scale and rotations. According to Bay et al, the SURF algorithm outperforms the SIFT descriptor (Lowe 1999).
Co-occurrence	Grayscale	The Co-occurrence descriptor uses the computation of the co-occurrence matrix (Haralick *et al.* 1973). In the literature, there are many descriptors based on the co-occurence matrix. Herein, we compute only the contrast, homogeneity and energy for each channel [R,G,B] (Castleman 1996).
CLP	Grayscale	The Crossing Line Profiles (CLP) uses straight lines crossing each segmented potential defect in the middle. The profile that contains the most similar gray levels in the extremes is defined as the best crossing line profile (BCLP). The features extracted corresponds to the first five harmonics of the fast Fourier transformation of BCLP (Mery 2003).

A real flaw entails a spatio-temporal relation in the different views where it appears, while a false alarm corresponds to a random event. An example of the tracking process with a real flaw detection is shown in Fig.6.

3.3.1 Tracking with two views

In the subsequent analysis we consider that the mass centre of the *i-th* potential defect in the *a-th* view is stored in homogenous coordinates as $\mathbf{m}_a^i = [x_a^i, y_a^i, 1]^\top$. If this potential defect is actually a real flaw it must have a corresponding point $\mathbf{m}_b^j$ in another consecutive or non-consecutive view b where a potential defect j was also segmented. According to the *principle of multiple view geometry* (Hartley and Zisserman 2000), the points $\mathbf{m}_a^i$ and $\mathbf{m}_b^j$ are correspondent if they are related by the fundamental matrix $\mathbf{F}_{a,b}$ such that

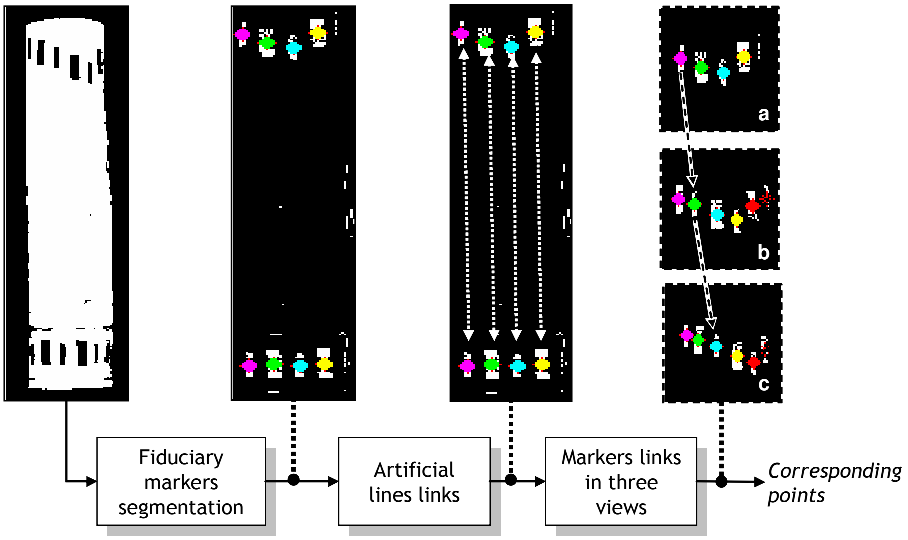
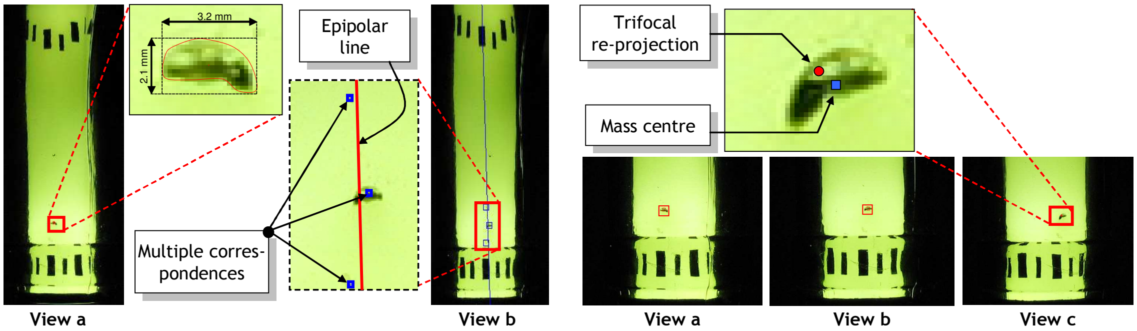


Figure 5. Segmentation of Fiduciary makers and computation of corresponding points in three views.



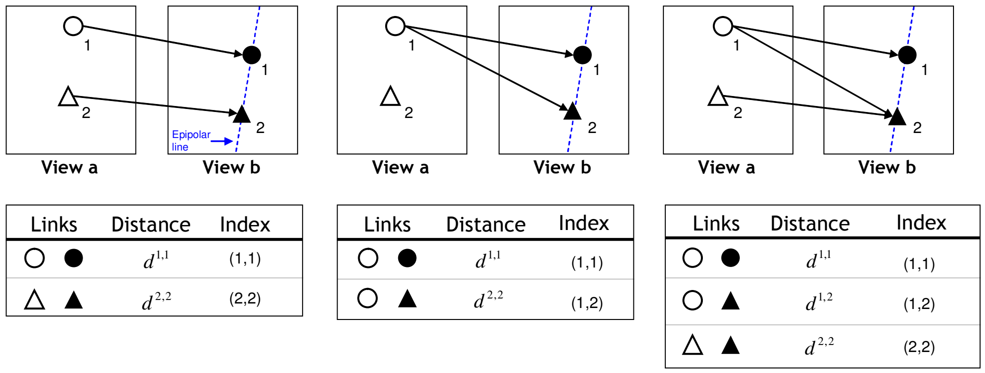
(a)

(b)

Figure 6. Examples of bifocal and trifocal correspondences in (a) and (b), respectively.

$$\mathbf{m}_b^{j\top}\,\mathbf{F}_{a,b}\,\mathbf{m}_a^i = 0\,.$$

This relation is known as *epipolar constraint*. It indicates that multiples points or one point $\mathbf{m}_b^j$, can only lie on the epipolar line of the point $\mathbf{m}_a^i$ defined as $\mathbf{l}_a^i = \mathbf{F}_{i,j}\,\mathbf{m}_a^i = [l_{a,x}^i, l_{a,y}^i, l_{a,z}^i]$. Herein, for simplicity we shown the case when the point $\mathbf{m}_b^j$ is corresponding with the point $\mathbf{m}_a^i$. Then, knowing $\mathbf{l}_a^i$ we identify a correspondence associated with the potential defect i as the potential defect j in the view b that satisfies the following condition



Links	Distance	Index
○ ●	$d^{1,1}$	(1,1)
△ ▲	$d^{2,2}$	(2,2)

(a)

Links	Distance	Index
○ ●	$d^{1,1}$	(1,1)
○ ▲	$d^{2,2}$	(1,2)

(b)

Links	Distance	Index
○ ●	$d^{1,1}$	(1,1)
○ ▲	$d^{1,2}$	(1,2)
△ ▲	$d^{2,2}$	(2,2)

(c)

Figure 7. Example of the potential matching in two views and estimation of the ratio with the Algorithm **1**. (a) The best case occurs when each potential defect is correctly linked; (b) When a potential defect has two options; (c) When multiple potential defects in the view *a* have matchings with one potential defect in the view *b*.

$$\frac{|\mathbf{m}_b^{j\top}\,\mathbf{F}_{a,b}\,\mathbf{m}_a^i|}{\sqrt{(l^i_{a,x})^2+(l^i_{a,y})^2}} < \varepsilon_1\,, \tag{1}$$

for small $\varepsilon_1 > 0$. The above condition measures the projective proximity of the potential flaws. If this constraint is fulfilled a potential matching is thus found in the two views. In this case it could be considered as a real flaw with a bifocal relationship. We want to emphasize that, in some cases, the above condition is not enough to ensure a correct matching between two views due to that multiple correspondences may occur, as shown in Fig. 7b-c. In order to filter most of wrong matchings founded in this step, we propose an effective method to quantify the feature similarity, described as follows.

3.3.2 Filter in two views

After condition (1) is fulfilled, a link of a potential flaw is founded in two consecutive views. Nevertheless, there are more than one possible combination. In fact, the problem of finding matchings can be grouped in three main classes: One-to-One, One-to-many, and Many-to-one. The filter in two views has as main objective to resolve these three cases or its combinations at the same time. In order to visualise our problem, we provide an explanation of these cases below:

i) **One-to-One:** The simple case of the matching problem is when there is one-to-one link between two potential defects in two views, as shown in Fig.7a. This does not mean that every link is correct (or a real flaw in correspondence), it only means that there is one possible option detected in both views by the tracking in two views.

ii) **One-to-Many:** In this case there are one-to-many links. Therefore the problem is to find the best neighbor based on the distance of the feature vector, as shown in Fig.7b. Additionally, it could be possible that a potential flaw does not have a matching in the second view. This happens when condition (1) is not fulfilled.

iii) **Many-to-One:** In this case there are multiple matchings corresponding to the same point in the view b, as shown in Fig. 7c. Normally it is not possible to deduce what matchings are correct or incorrect based only on the distance of multiple feature vectors. For this reason we propose a novel method to reduce wrong links.

For later analysis, we will introduce some notations in order to quantify correctly each feature similarity. Firstly, recall that $\mathbf{z}_a^i$ and $\mathbf{z}_b^j$ are the feature vectors of regions $\mathbf{m}_a^i$ and $\mathbf{m}_b^j$ respectively, extracted in the identification step, as shown in Fig. 4. In order to quantify which features are more suitable for our problem, we extracted multiple feature descriptors in the identification step (see Table 1). Based on this information, let $d_{a,b}^{i,j} = \|\mathbf{z}_a^i - \mathbf{z}_b^j\|$ be the distance of a feature vector of the potential defect i of the view a and the potential defect j of the view b. Using the above distance, let $\mathbf{D}_{a,b}$ be the global distance vector of all distances detected in both views, i.e, the vector $\mathbf{D}_{a,b}$ is a collection of distances for all pairs of matching detected in the tracking step. Since the number of matchings is variable for every two consecutive views, let p the length of the vector $\mathbf{D}_{a,b}$, defined as

$$\mathbf{D}_{a,b} = \begin{bmatrix} d_{a,b}^{1,1} \\ \vdots \\ d_{a,b}^{i,j} \\ \vdots \\ d_{a,b}^{n,m} \end{bmatrix}$$

where n and m are the total number of defects segmented in the view a and b respectively. Finally, let $\mathbf{X}_{a,b}$ be the vector composed by an array of indices for every matching detected in both views. $\mathbf{X}_{a,b}$ has the same length of vector $\mathbf{D}_{a,b}$ and it is defined as

$$\mathbf{X}_{a,b} = \begin{array}{c} 1 \\ \vdots \\ \\ \vdots \\ p \end{array} \begin{bmatrix} \{1,1\} \\ \vdots \\ \{i,j\} \\ \vdots \\ \{n,m\} \end{bmatrix}$$

Normally, to establish a matching between two (or more) descriptors is used the Nearest-Neighbor with Distance Ratio (NNDR) (Mikolajczyk and Schmid 2005). Although the NNDR criterion does not have the best performance, it has been selected because it is less expensive computationally (Sidibe *et al.* 2007). For the sake of completeness, we describe the NNDR criterion using the same notation described above, as shown in Algorithm **1**. As a result, the NNDR criterion returns an index vector $\mathbf{X'}_{a,b}$ of the same size of vector $\mathbf{X}_{a,b}$ containing the best matchings in two views. Unfortunately, the main disadvantage of the NNDR criterion is that only seeks matchings in one direction, i.e., trying to seek -for each feature descriptor from the left image- the nearest neighbor or the two nearest neighbors from the right image. The problem that can arise is when multiple point descriptors are related with an unique descriptor in the left image, as shown Fig.7c. In that case, the NNDR criterion does not provide a good performance,

Algorithm 1 Nearest Neighbor Distance Ratio (NNDR) algorithm for multiple correspondences

```
Input: $\mathbf{Z}_{a,b}$, $\mathbf{X}_{a,b}$ and $ratio$
Output: $\mathbf{X}'(p \times 1, \mathbb{R})$
$\mathbf{X}' \leftarrow \varnothing$
for all $t \in [1, \ldots, p]$ do
    $\mathbf{N} \leftarrow$ set of all index $w$ such that $\mathbf{X}(w) = \{t, j\}$, for all $j$
    $\mathbf{S}, \mathbf{I} \leftarrow sort\ (\mathbf{Z}_{a,b}(\mathbf{N}))$ {$\mathbf{S}$: for values; $\mathbf{I}$: for index re-order}
    if $\mathbf{S}(1) < \mathbf{S}(2) * ratio$ then
        $\mathbf{X}'(\mathbf{I}(\mathbf{1})) \leftarrow \{t, \mathbf{I}(\mathbf{1})\}$
        $\mathbf{X}'(\mathbf{I}(\mathbf{2})) \leftarrow \{\varnothing\}$
    else
        $\mathbf{X}'(\mathbf{I}(\mathbf{1})) \leftarrow \{t, \mathbf{I}(\mathbf{1})\}$
        $\mathbf{X}'(\mathbf{I}(\mathbf{2})) \leftarrow \{t, \mathbf{I}(\mathbf{2})\}$
    end if
end for
return $\mathbf{X}'$
```

Algorithm 2 **R**–distance. Calculate the ratio of feature descriptors in both views

```
Input: $\mathbf{Z}_{a,b}$, $\mathbf{X}_{a,b}$ such that $k > 0$
Output: $R(p \times 1, \mathbb{R})$
$R \leftarrow \varnothing$
for all $t \in [1, \ldots, p]$ do
    $\mathbf{N} \leftarrow$ set of all index $w$ such that $\mathbf{X}(w) = \{t, j\}$, for all $j$
    $\mathbf{M} \leftarrow$ set of all index $w$ such that $\mathbf{X}(w) = \{i, t\}$, for all $i$
    for all $n \in \mathbf{N}$ do
        if $|\mathbf{M}| = 1$ then
            $R(n) \leftarrow 1$
        else
            $R(\mathbf{M}) \leftarrow \frac{\mathbf{Z}_{a,b}(\mathbf{M})}{min(\mathbf{Z}_{a,b}(\mathbf{M}))}$
        end if
    end for
end for
return $R$
```

or in some cases, it will preserve wrong matchings. In order to improve this scheme, we propose a novel procedure for establishing a correct matching in two views, improving the original NNDR criterion. Firstly, we compute the ratio distance **R**, used for estimating the distance of feature vectors in both directions. The ratio **R** estimated is weight factor that adjusts the distance of the feature vector $\mathbf{D}_{a,b}$ of all matchings detected in both views. The computation of the **R** vector is shown in Algorithm **2**.

Once the **R** distance vector has been computed, the next step is to adjust the feature similarity vector $\mathbf{D}_{a,b}$. We call this procedure as expanded Nearest Neighbor Distance Ratio (eNNDR), because it allows us to increase the ratio difference of the original feature vector in both directions. Although the eNNDR procedure has been designed specially in those cases where there are many-to-one matchings, it can resolve any combination (one-to-many or one-to-one) described previously. We must recall that our procedure is used in conjunction with the NNDR criterion, nevertheless it obviates the original NNDR limitations. Using the same notation as in the previous NNDR procedure, the eNNDR

(1) Estimate the **R**-distance vector by applying the Algorithm **2** using the vector $\mathbf{Z}_{a,b}$ and the index $\mathbf{X}_{a,b}$, in order to seek the vector R (as output, the vector $\mathbf{R}$ has p rows).
(2) Compute the vector $\mathbf{Z'}_{a,b} = R^{\top} \cdot \mathbf{Z}_{a,b}$.
(3) Compute the NNDR criterion using the distance feature $\mathbf{Z'}_{a,b}$, as a result, we obtain a new vector $\mathbf{X'}_{a,b}$ containing the updated indices.

Figure 8. expanded Nearest Neighbor Distance Ratio (eNNDR) procedure to filter wrong matchings in two views

procedure requires firstly to compute the **R** distance. More formally, the eNNDR in two views is described in Fig. 8. As we will see in the experimental section, the eNNDR is able to outperform the performance of the NDDR criterion in its simple form.

3.3.3 Tracking with three views

After applying the filter in two views, the next procedure is to confirm that a bifocal correspondence represents indeed a real flaw. Nevertheless, there are still false alarms to reduce. For this reason, we search for a new correspondence in a third view with the help of trifocal tensors. Let $\mathbf{T} = (T_t^{rs})$ be a $3 \times 3 \times 3$ matrix representing the trifocal tensor that encodes the relative motion among the views a, b, c[1]. Then, we can estimate the hypothetical position of a defect k in a third view c using the correspondences $\mathbf{m}_a^i, \mathbf{m}_b^j$ and the tensor $\mathbf{T}$ as[2]

$$\widehat{\mathbf{m}}_c^k = \frac{1}{\mathbf{m}_a^{i\top}(T_{\cdot}^{13} - x_b^j\, T_{\cdot}^{33})} \begin{bmatrix} \mathbf{m}_a^{i\top}(T_{\cdot}^{11} - x_b^j\, T_{\cdot}^{31}) \\ \mathbf{m}_a^{i\top}(T_{\cdot}^{12} - x_b^j\, T_{\cdot}^{32}) \\ \mathbf{m}_a^{i\top}(T_{\cdot}^{13} - x_b^j\, T_{\cdot}^{33}) \end{bmatrix} .$$

We compare the estimated position $\widehat{\mathbf{m}}_c^k$ with all potential flaws $\mathbf{m}_c$ in the view c, and we choose the potential defect k that satisfies the following condition

$$\|\widehat{\mathbf{m}}_c^k - \mathbf{m}_c^k\| < \varepsilon_2 \,, \qquad (2)$$

In this case a potential defect is thus found in the three views, i.e., a real flaw with a trifocal correspondence has been detected based only on its projective proximity. Potential defects that do not find correspondence in three views are discarded and considered as false alarms. An example is shown in the Fig. 6b.

Even though the noise appears randomly in images, there are still some remaining false alarms in three views; most of them are wrong links. Again, the following step is to reduce these false alarms by means of a feature comparison.

[1] See (Hartley and Zisserman 2000) for details on the computation of the trifocal tensors.

[2] The estimated projection in the third view can be improved applying the point-line-point method proposed in (Hartley and Zisserman 2000, pp.373).

3.3.4 Filter in three views

Similarly as was defined in the tracking in two views, we define the values $z^{i,j}_{a,b}$, $z^{i,k}_{a,c}$ and $z^{j,k}_{b,c}$ to be the feature distance of each potential matching $\{i, j\}$, $\{i, k\}$ and $\{j, k\}$ between views $\{a, b\}$,$\{a, c\}$ and $\{b, c\}$ respectively. Also, let $\mathbf{Z}_{a,b}$, $\mathbf{Z}_{a,c}$, and $\mathbf{Z}_{b,c}$ be the global distance vector; and finally, let $\mathbf{X}_{a,b}$, $\mathbf{X}_{a,c}$ and $\mathbf{X}_{b,c}$ be the index vector of each matching detected in three views.

The final step of our method consists in filtering the remaining false alarms that have not been filtered out by the above intermediate steps. As well as the filter in two views, the filter in three views uses the feature similarity as a criterion function in order to discriminate and filter wrong links. Based on previous results, we propose an extended version of the eNNDR procedure in three views that uses a similar procedure in two views, described in Fig.9.

(1) Estimate the **R**-distance vector by applying the Algorithm **2** on vectors $\mathbf{Z}_{a,b}$, $\mathbf{Z}_{a,c}$, and $\mathbf{Z}_{b,c}$ in order to get vectors $R_{a,b}$, $R_{a,c}$ and $R_{b,c}$ respectively.
(2) Compute the vector $\mathbf{Z}_{a,b,c} = R^{\top}_{a,b} \cdot \mathbf{Z}_{a,b} \cdot R^{\top}_{a,c} \cdot \mathbf{Z}_{a,c} \cdot R^{\top}_{b,c} \cdot \mathbf{Z}_{b,c}$.
(3) Compute the NNDR criterion using the distance feature $\mathbf{Z}_{a,b,c}$, as a result, we get a new vector $\mathbf{X}_{a,b,c}$ containing the updated indices.

Figure 9. expanded Nearest Neighbor Distance Ratio (eNNDR) procedure to filter wrong matchings in three views

Finally, the matching set in three views with intermediate filters are stored in the vector $\mathbf{X}_{a,b,c}$. Our method does not use more views, primarily because most potential defects can only be tracked along three views, dismissing the possibility to use more views. Thus, more than three views do not imply an increment of the performance.

4. Experimental Evaluation

We now evaluate the performance of our proposed methodology for inspecting bottlenecks of empty wine bottles. In our experiments we use 120 color image sequences with several hundreds of real flaws. Each sequence consists of 3 views with a rotation angle $\alpha = 15$ degrees between them. From the recorded images we extract sub-images of the bottlenecks of 1000×250 pixels. The number of real flaws per image fluctuates between 0 and 4 with an average of 2.8, and the number of false alarms per image fluctuates between 0 and 10 with an average of 9.5 per image. The area of the smallest defect is around 9 pixels equivalent to 0.16 mm^2. The performance is assessed considering two standard indicators: *recall*, $\mathrm{r} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$ and *precision*, $\mathrm{p} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$. TP is the number of true positives or defects correctly classified. This occurs when bifocal (or trifocal) correspondences have been correctly established in two (or three) views. FN is the number of false negatives or existing real defects not detected by our tracking methodology. FP is the number of false positives. This is the case when flawless regions are incorrectly classified as defective. We can ideally expect that recall = 100% and precision = 100%.

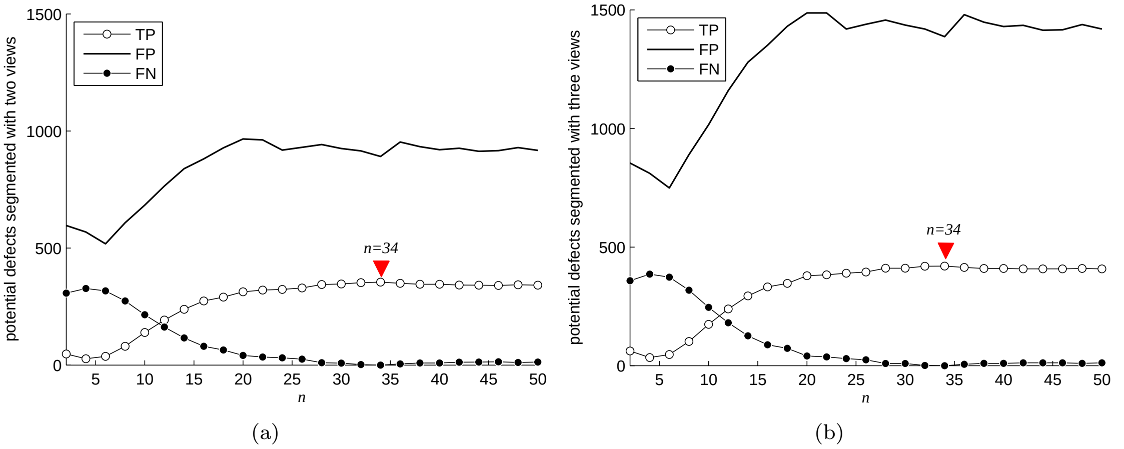


Figure 10. Influence of the kernel size ($n \times n$) of the SR filter on the performance rates of the proposed segmentation procedure using (a) bifocal analysis, (b) trifocal analysis.

For measuring these two indicators in one measure, we use the F-score indicator, where

$$F = \frac{2 \cdot p \cdot r}{p + r}.$$

In the following subsections, we will evaluate the performance of our method by changing different parameters. Specifically we experimented with the kernel size n, the ε parameter, the feature descriptor used, and finally the eNNDR algorithm versus the NNDR criterion. For the sake of clarity, we separate our evaluations in four steps by evaluating the above parameters. Finally, we make a comparison with respect to other existing systems of bottle inspection.

Evaluation 1

We also want to emphasize the importance of the segmentation task described in Section 3.1, where potential defects are automatically identified and characterised. The selection of the kernel size ($n \times n$) of the SR filter is crucial for obtaining the uniform background image from which the structures (potential defects) were removed. Fig. 10 shows the influence of the kernel size of the SR filter on the segmentation results for the bifocal and the trifocal analysis. As it is expected the number of detected real flaws augments as the kernel size increases until $n = 34$. In particular, at $n = 34$ the number of FN is zero and the number of TP is maximal, meaning that all existing real flaws have been segmented. After this value, the performance starts to decrease slowly. According to the previous analysis, the results indicates that there are in average 2.5 times more false alarms than real flaws, supporting the idea that multiples views must be used in order to discriminate correctly false alarms from real flaws. For later analysis, we set the kernel size to $n = 34$.

Evaluation 2

This evaluation shows the influence of the thresholds ε_1 and ε_2 used to establish bifocal and trifocal correspondences; for simplicity we fixed $\varepsilon_1 = \varepsilon_2 = \varepsilon$. In the same way, we study how each feature descriptor influences the performance of our tracking system (Fig.11 and Fig.12), by the inclusion of two independent filters after each tracking process. Namely, we compare ten feature descriptors, where eight of them are invariants and two are non-invariant (more details in Table 1). Likewise, for each descriptor, we evaluate and compare the effect of using: 1) our proposed eNNDR algorithm in two and in three views, and 2) the standard NNDR criterion only. We recall that the filter is used to discriminate wrong matchings from true matchings by means of a feature comparison.

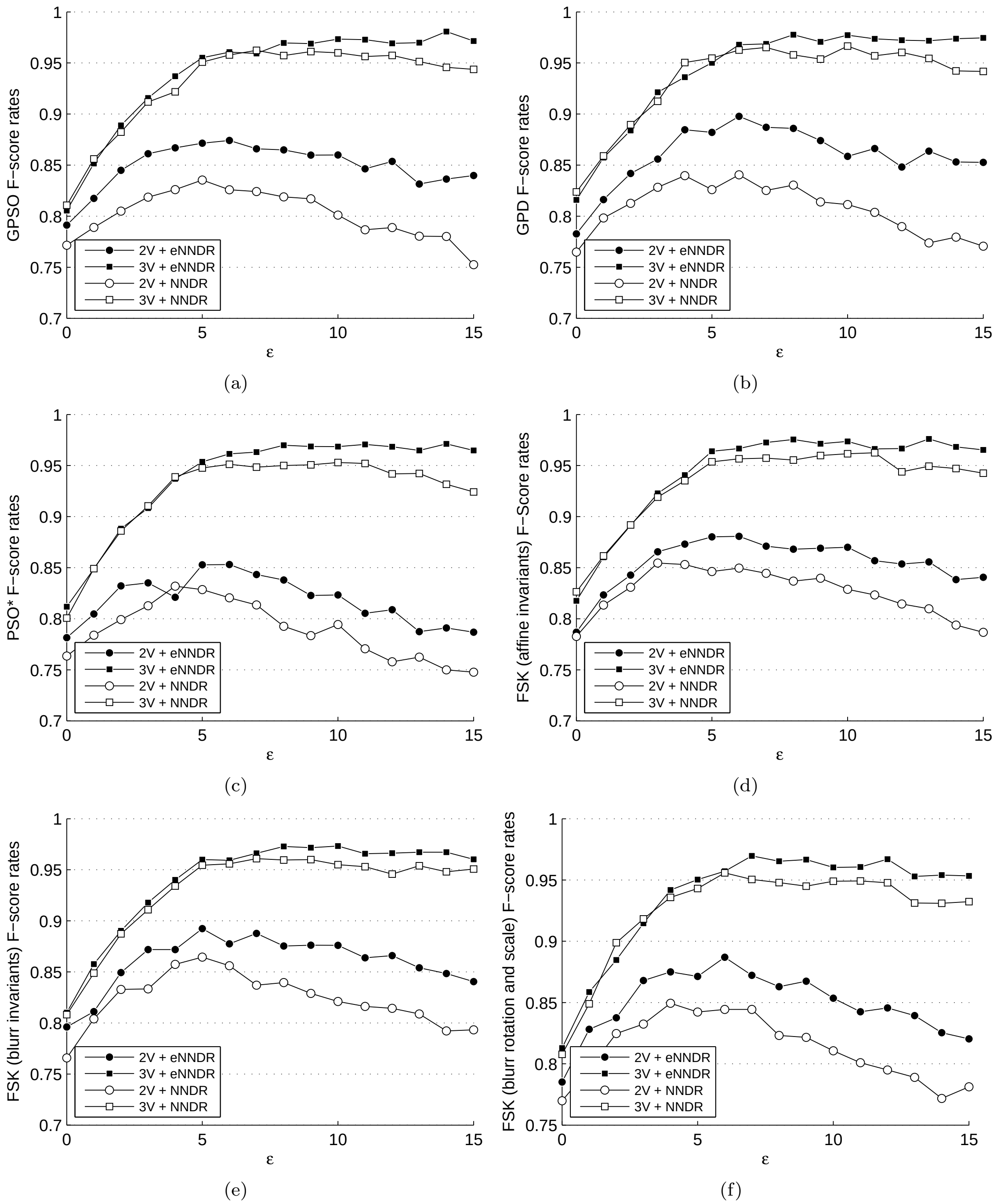
In order to put in context all previous descriptors used in this study, Fig.13 shows the mean performance of the tracking method using the bifocal and trifocal analysis. 1) Bifocal tracking. In this case, the tracking with two views reveals a clear tendency to decrease the performance as it increases the threshold ε. In this case the lower performance has been obtained by a non-invariant descriptor, namely the CLP descriptor; as we expected. Surprisingly, the best mean performance has been obtained by the co-matrix descriptor. This results can be explained by the use of images with small geometrical transformations only in two views. The second best mean performance has been obtained with the GPD feature descriptor due to its invariance against geometric and photometric transformations. 2) Trifocal tracking. In this case, there is no clear best descriptor because the performances of all descriptors have given good performances as it varies the threshold ε. In overall, it is worth mentioning that, as the threshold ε is increased over six pixels, the global performance is increased, allowing us to reduce most of false alarms. Although the threshold is high, this can be explained due to the uncalibrated nature of the acquired image sequences we are working with, which can produce imprecisions in computing the fundamental matrix and the trifocal tensors. In particular, the best mean performance has been achieved by the SURF descriptor, since there are more geometric deformations in three views than in two views.

Evaluation 3

Based on previous results, we show the advantage of using our proposed procedure to filter wrong links in two and three views, described previously. Fig.14 shows the mean F-score percentage of improvement for each feature descriptor using our proposed filter eNNDR with regard to use the NNDR criterion only. As we observe, eNNDRr can improve the classical NNDR criterion up to 6.7% in combination with the SURF descriptor. In overall, the mean of improvement is around at 4.6% using two views, and 1.4% using three views. Recall that our propose filter uses the NNDR criterion as a final step, in consequence these results clearly reveal that our method can deal with matchings problems; specifically when there are many-to-one links. Even though the NNDR criterion has not been designed to resolve this kind of problems, our procedure is able to transform the feature distance vector in order to be used by the NNDR criterion effectively.

Evaluation 4

Finally, in Table 2 we juxtapose our inspection results using the bifocal and trifocal analysis with other inspection systems proposed in the literature, indicating the use of single or multiple images. We denote as 2V (two views) and 3V (three views) the inspec-



(a)

(b)

(c)

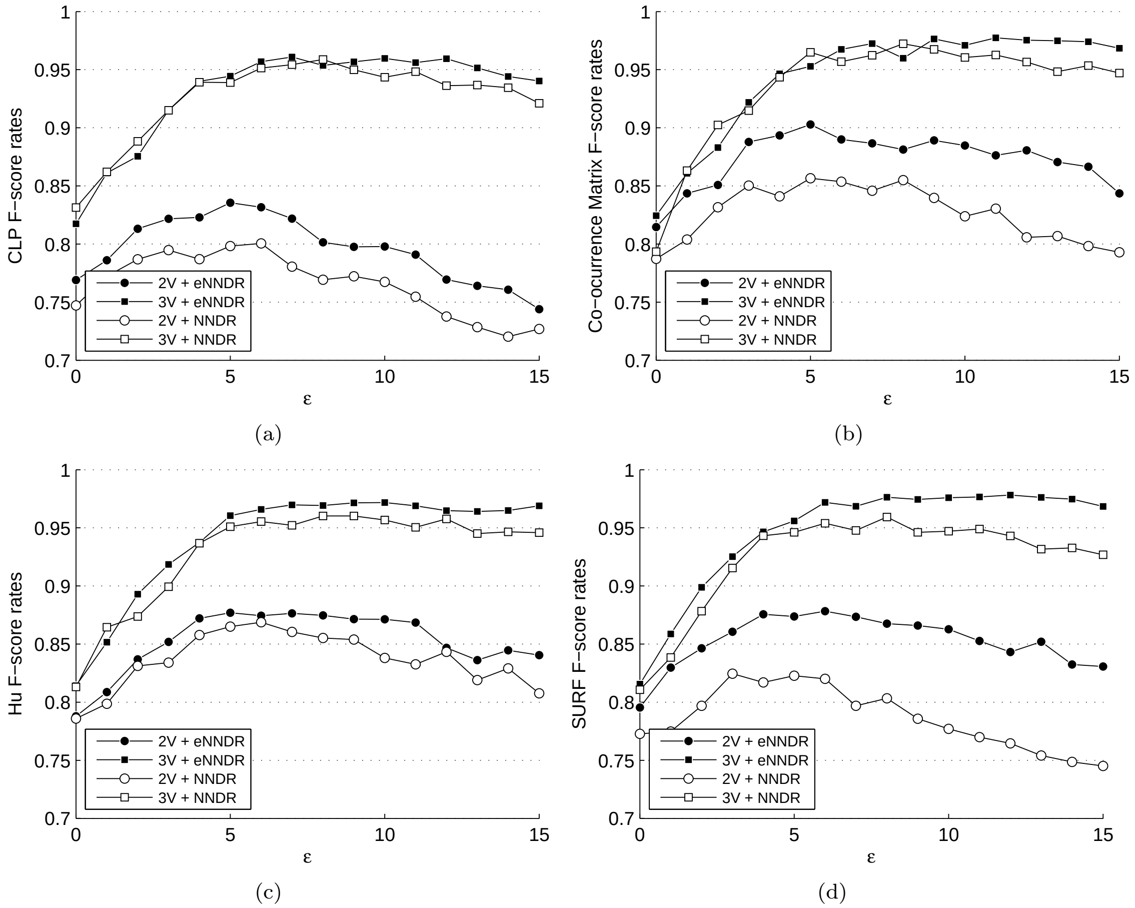
(d)

(e)

(f)

Figure 11. Influence of the thresholds $\varepsilon_1 = \varepsilon_2 = \varepsilon$ of the conditions (1)–(2) used to establish bifocal and trifocal correspondences, using the proposed eNNDR algorithm and the NNDR criterion.

tion results obtained when using only the geometric constraints (1) and (2), respectively. Similarly, 2V+eNNDR and 3V+eNNDR represent the inspection results using the eNNDR algorithm. According to the results presented in the above experiments, our best



(a)

(b)

(c)

(d)

Figure 12. Influence of the thresholds $\varepsilon_1 = \varepsilon_2 = \varepsilon$ of the conditions (1)–(2) used to establish bifocal and trifocal correspondences, using the proposed eNNDR algorithm and the NNDR criterion.

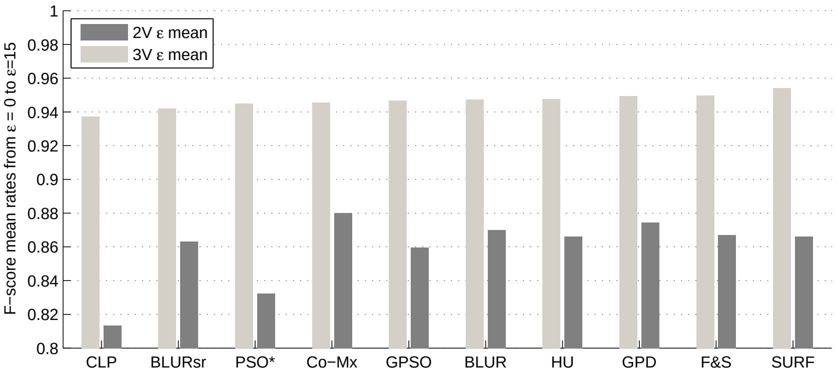


Figure 13. Mean performance rates from $\varepsilon = 1$ to $\varepsilon = 15$ for each feature descriptor listed in Table.1

performance has been achieved by the GPD descriptor for feature matching, by set-

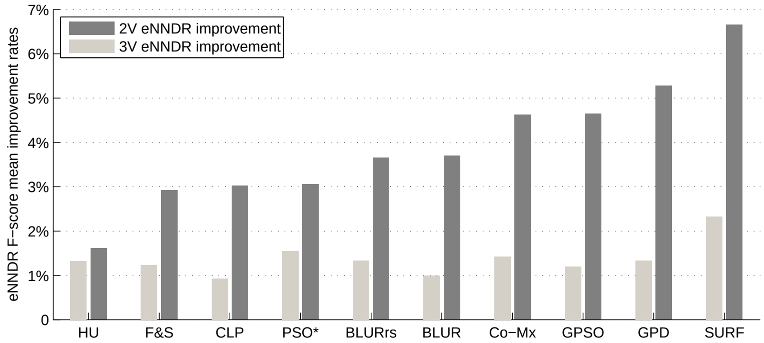


Figure 14. Mean improvement rates for each feature descriptor evaluated from $\varepsilon = 0$ up to $\varepsilon = 15$ using the eNNDR algorithm over the NNDR criterion.

Table 2. Comparison with other inspection systems proposed in the literature.

Inspected bottle part	Views	Tracking	TPR	FPR
neck (Mery and Medina 2004)	Single	No	85%	4%
lips, body, bottom (Duan *et al.* 2007)	Single	No	97%	>1%
lips (Wang *et al.* 2005)	Single	No	98%	0%
body (Firmin *et al.* 1997, Hamad *et al.* 1998)	Multiple	No	80%-85%	2%
lips, neck (Ma *et al.* 2002)	Multiple	No	98%	2%
body, bottom (Shafait *et al.* 2004)	Multiple	No	100%	>1%
neck (Our Method)	2V	Yes	99.9%	42.2%
	2V+eNNDR	Yes	98.4%	24.6%
	3V	Yes	98.7%	3.8%
	3V+eNNDR	Yes	**98.5%**	**2.4%**

ting parameters at $\varepsilon = 10$px and $n = 34$. Although other invariants and non-invariants descriptors have also achieved good results, we selected the GPD descriptor for four reasons. 1) by its good performance under two and three views, 2) by its high True Positive Rate (TPR), 3) by its low False Positive Rate (FPR), i.e., the capacity to discriminate correctly wrong links, and 4) by its high overall performance in combination with the filter proposed Fig.14

It is important to mention that Table 2 shows just a quantitative comparison, since the other methods proposed in the literature were tested on different images (or image sequences) and on different types of bottles, and they inspect one or several bottle parts (lips, mouth, bottleneck, body, bottom). Nevertheless, to the best of our knowledge, our methodology for glass bottle inspection is the first one that performs tracking of potential flaws in multiple views.

Concerning the real-time capabilities of our inspection system, the computational time required to process trifocal correspondences was 1.3 sec in average, using a Matlab 7.0's implementation running on a Pentium Centrino 2.0 GHz under Windows XP SP2. 34% of computational time is spent reading the images, 35% in the segmentation, 11% in the trifocal analysis, and 20% in other Matlab's internal operations.

5. Conclusions

Our principal contribution was twofold: First, we presented a prototyping design of an electro-mechanical device for acquiring image sequences of glass bottlenecks using a single camera. Its main novelty is the placement of the illuminating source inside the bottle, which greatly improves the definition of the inspected images. Second, we proposed a new methodology for detecting flaws in the bottleneck based on tracking potential defects along image sequences. Our inspection system achieves performance rates of 98.5% true positives and 2.4% false positives Thirdly, we improved the classical NNDR algorithm for featuring matching by proposing a new extended-NNDR (eNNDR) algorithm that it can handle featuring matching in a global way. Although these excellent results, our implementation is not yet competitive in terms of computational time. In this sense, several improvements are matter of ongoing work. For instance: the use of a fast industrial camera instead of a slow commercial camera together with a faster bottle rotating mechanism; the transfer of our Matlab implementations to a highly efficient low-level programming language; and the exploitation of multigrid implementation strategies. Additional future work includes the adaptation of our electro-mechanical device for inspecting not only bottlenecks, but other bottle parts as well.

Acknowledgements

We gratefully acknowledge partial funding by CONICYT – Colegio Doctoral Franco-Chileno, grant no. 21050185.

References

Bay, H., *et al.*, 2008. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110 (3), 346–359.

Canivet, M., Zhang, R.D., and Jourlin, M., 1994. Finish inspection by vision for glass production. *In*: B.M. Dawson, S.S. Wilson and F.Y. Wu, eds. *Machine Vision Applications in Industrial Inspection II*, Vol. 2183, San Jose, CA, USA Proceedings of SPIE, 164–169.

Carrasco, M. and Mery, D., 2006. Automated Visual Inspection using trifocal analysis in an uncalibrated sequence of images. *Materials Evaluation*, 64 (9), 900–906.

Carrasco, M., Pizarro, L., and Mery, D., 2008. Image Acquisition and Automated Inspection of Wine Bottlenecks by Tracking in Multiple Views. *In*: *Proc. of 8th Int. Conf. on Signal Processing, Computational Geometry and Artificial Vision (ISCGAV'08)*, Rhodes Island, Greece, 82–89.

Castleman, K., 1996. *Digital Image Processing.* New Jersey: Prentice-Hall.

Chen, Z., *et al.*, 2000. A robust algorithhm to estimate the fundamental matrix. *Pattern Recogn Letters*, 21, 851–861.

Drury, C.G., Saran, M., and Schultz, J., Effect of Fatigue / Vigilance/ Environment on Inspectors Performing Fluorescent Penetrant and/or Magnetic Particle Inspection. , 2004. , Interim Report 03-G-012, University at Buffalo, Federal Aviation Administration William J. Hughes Technical Center.

Duan, F., *et al.*, 2007. A machine vision inspector for beer bottle. *Eng Appl Artif Intell*, 20 (7), 1013–1021.

Firmin, C., *et al.*, 1997. Gaussian neural networks for bottles inspection: a learning procedure. *Int J Neural Syst*, 8 (1), 41–46.

Flusser, J. and Suk, T., 1993. Pattern recognition by affine moment invariants. *Pattern Recogn*, 26 (1), 167–174.

Flusser, J., Suk, T., and Saic, S., 1996. Recognition of images degraded by linear motion blur without restoration. *Computing. Supplement*, 11, 37–51.

Hamad, D., *et al.*, 1998. Neural Networks inspection system for glass bottles production: A comparative study. *Int J Pattern Recogn Artif Intell*, 12 (4), 505–516.

Haralick, R., Shanmugam, K., and Dinstein, I., 1973. Textural Features for Image Classification. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-3 (6), 610–621.

Hartley, R. and Zisserman, A., 2000. *Multiple View Geometry in Computer Vision.* Cambridge, UK: Cambridge University Press.

Hu, M.K., 1962. Visual Pattern Recognition by Moment Invariants. *IRE Trans Info Theory*, IT (8), 179–187.

Hui-Fuang, N., 2006. Automatic thresholding for defect detection. *Pattern Recogn Letters*, 27 (14), 1644–1649.

Katayama, K., *et al.*, 2008. Optical Inspection of Glass Bottles using Multiples Cameras. [online] USA Pantent No. 7.329.855 B2.

Kopparapu, S.K., 2006. Lighting design for machine vision application. *Image Vis Comput*, 24 (7), 720–726.

Lowe, D.G., 1999. Object recognition from local scaleinvariant features. *In*: *International Conference on Computer Vision*, Vol. 2, Corfu, Greece, 1150–1157.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis*, 60 (2), 91–110.

Ma, H.M., Su, G.D., and Ni, J.Y.W.Z., 2002. A glass bottle defect detection system without touching. *In*: *Int. Conf. on Machine Learning and Cybernetics*, Vol. 2, 628–632.

Martin, J., 2005. BASIC Stamp Syntax and Reference Manual.

Mery, D., 2003. Crossing line profile: a new approach to detecting defects in aluminium castings. *LNCS*, 2749, 725–732.

Mery, D. and Carrasco, M., 2006. Advances on Automated Multiple View Inspection. *LNCS*, 4319, 513–522.

Mery, D. and Medina, O., 2004. Automated visual inspection of glass bottles using adapted median filtering. *LNCS*, 3212, 818–825.

Mery, D. and Filbert, D., 2002. Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence. *IEEE Trans. Robotics and Automation*, 18 (6), 890–901.

Mikolajczyk, K. and Schmid, C., 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 (10), 1615–1630.

Mindru, F., *et al.*, 2004. Moment invariants for recognition under changing viewpoint and illumination. *Comput Vis Image Underst*, 94 (1-3), 3–27.

Newman, T.S. and Jain, A.K., 1995. A survey of Automated Visual Inspection. *Computer Vision and Image Understanding*, 61 (2), 231–262.

Parker, J., 2000. Defect in glass and their origin. *In*: *First Balkan Conference on Glass Science and Technology*, 9-10 Oct., Vollos, Greace.

Pizarro, L., *et al.*, 2008. Robust automated multiple view inspection. *Pattern Analysis and Applications*, 11 (1), 21–32.

Shafait, F., Imran, S., and Klette-Matzat, S., 2004. Fault detection and localization in empty water bottles through machine vision. *In*: *Emerging Technology Conference*, 30–34.

Sidibe, D., Montesinos, P., and Janaqi, S., 2007. Fast and Robust Image Matching using Contextual Information and Relaxation. *In*: *2nd International Conference on Computer Vision Theory and Applications, VISAPP*, March., Barcelona, Spain.

Sonka, M., Hlavac, V., and Boyle, R., 1999. *Image Processing, Analysis, and Machine Vision.* 2 Pacific Grove, CA: PWS Publishing.

Wang, J. and Asundi, A., 2000. A computer vision system for wineglass defect inspection via Gabor-filter-based texture features. *Inform Sci*, 127, 157–171.

Wang, Y.N., H.-J., L., and Duan, F., 2005. A bottle finish inspect method based on fuzzy support vector machines and wavelet transform. *In*: *International Conference on Machine Learning and Cybernetics*, Vol. 8, 4588–4592.

Yan, T.S. and Cui, D.W., 2006. The method of intelligent inspection of product quality based on computer vision. *In*: *Proc. of the 7th Int. Conf. on Computer-Aided Industrial Design and Conceptual Design (CAIDCD '06)*, Hangzhou, 1–6.

Yepeng, Z., Yuezhen, T., and Zhiyong, F., 2007. Application of digital image process technology to the mouth of beer bottle defect inspection. *In*: *Proc. of the 8th Int. Conf. on Electronic Measurement and Instruments*, Vol. 2, 905–908.

Yi, S., Haralick, R., and Shapiro, L., 1995. Optimal sensor and light source positioning for machine vision. *Comput Vis Image Underst*, 61 (1), 122–137.