A Machine Vision Apparatus and Method for Can-End Inspection

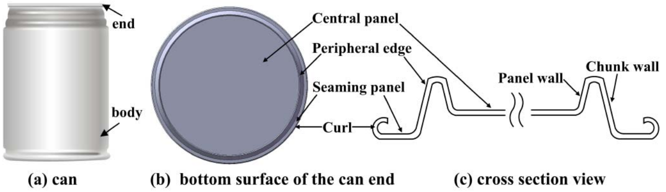
Tiejian Chen, Yaonan Wang, Changyan Xiao, and Q. M. Jonathan Wu, *Senior Member, IEEE*

***Abstract*—As a critical component of metal can containers, can end and its manufacture quality are closely relevant to product safety in food and beverage industry. To satisfy the requirements of quality control, a machine vision apparatus for real-time can-end inspection is presented in this paper. With a brief description of the apparatus system design and imaging system, our emphasis is put on the postprocessing image analysis. To detect defects and deformations across the imaged can-end surface, an entropy-rate clustering algorithm combined with prior shape constraint is proposed to locate the can-end object and divide it into multiple measuring regions. Then, a superpixel grouping and selection scheme is adopted to find defective areas inside the flat central panel. For the other three annular measuring regions, a multiscale ridge detection algorithm is introduced to seek defects and deformations along their projection profiles. According to in-line experiments and test, our apparatus can find out a majority of the can-end defects with a detection accuracy as high as 99.48% for various circular can ends.**

***Index Terms*—Entropy-rate clustering, machine vision, ridge detection, superpixel, surface defect detection.**

I. Introduction

IN FOOD and beverage industry, metal cans are widely used as a main-stream packaging container due to their distinct merits, such as lightweight, full recyclability, easiness to store, and cost-effectiveness. In general, a typical two-piece can [see Fig. 1(a)] includes two parts: a can body and a can end. The former is composed of a cylindrical sidewall and an integrally formed bottom wall. It has an opening at the top with an annular and outwardly extending peripheral flange. The latter consists of four functional parts, as shown in Fig. 1(b), that is: 1) central panel; 2) peripheral edge; 3) seaming panel; and 4) curl from inside to outside [1]. According to design specifications [2], the central panel is used to seal the opening of can body after filling with desired food or beverage, and the other three peripheral parts are devised to form an interlock structure with the flange of can body for product preservation. To achieve this aim, the peripheral edge is designed to consist of two slope surfaces, i.e., the panel and chunk wall, as shown in Fig. 1(c), while sealing compound is applied to the countersink seaming panel for can closure, and the curl is a curly structure used to hook the flange. The complex structure and series of manufacture processes [3] often make the can end much more vulnerable to various defects or deformations compared with the can body. These potential defects might cause serious safety problems in product storage and transportation. Hence, it is mandatory to inspect the quality of can ends before they are used for packaging.



(a) can

(b) bottom surface of the can end

(c) cross section view

Fig. 1. Structure of a typical can end. (a) Whole metal can. (b) and (c) Bottom and cross-sectional views of a can end, respectively.

To satisfy the requirement of can-end quality control, the early work mainly focused on in-line inspection of manufacture process status. Some sensing modules were developed, which utilized metal proximity sensors [4] or strain gauges [5] to acquire the operation parameters of the tooling system during can-end forming, and then, some data processing units were used to detect the abnormality and malfunction of operations. However, this approach can only detect specific kind of defect.

As a promising and nondestructive measurement technique, machine vision has been widely used in a variety of industry inspection applications [6]. It also provides a new and flexible solution to can-end quality control [7]. Although there are a few commercial vision systems available for can-end inspection [8], the detailed technical implementation and performance are seldom reported in the literature. Currently, such system is only used for inspecting sealing compound in the seaming panel [3]. In general, the development of machine vision method for can-end inspection is comparatively rare. A related work can be referred to [9], but they mainly handled the coating quality control with little concern on general surface defects. Recently, Feng *et al.* [10] briefly described a binary-channel machine vision system for can cover defect detection; however, their experimental validation and detected defect types appear limited.

Manuscript received September 29, 2015; revised January 6, 2016; accepted February 14, 2016. Date of publication May 23, 2016; date of current version August 9, 2016. This work was supported by the National Natural Science Foundation of China under Grant 61433016, Grant 61573134, and Grant 61571184. The Associate Editor coordinating the review process was Dr. Amitava Chatterjee.

T. Chen, Y. Wang, and C. Xiao are with the National Engineering Laboratory for Robot Visual Perception and Control Technology, College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: chentiejian@hnu.edu.cn; yaonan@hnu.edu.cn).

Q. M. J. Wu is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2016.2566442

Similar work can be referred to the applications of machine vision technique in metallic surface defect detection studies. In [11], the defects on rail head are detected with local normalization algorithm, and wavelet-based method is proposed for casting defect detection in [12]. Image classification is also widely adopted in defect detection [13]. For example, the defects on hot-rolled flat steel products are located by choosing defective image blocks with the kernel classifier [14]. Other approaches include the robust k-means clustering algorithm for deformations detection on E-shaped magnetic cores [15], the active contour model [16] for Liquid Crystal Display defect detection, and so on. However, these algorithms are mostly proposed to solve specific problems. Besides, most test items have flat surface, which are basically different from the can end. Hence, these methods cannot be directly applied to can-end defect detection.

In this paper, we will present a machine vision apparatus for circular can-end inspection, which is expected to detect all the surface defects on their bottom surface with real-time speed. The emphasis is put on image-processing algorithms for defect detection across different can-end regions. The main contribution of this paper is fourfold.

1) A novel imaging system is designed to obtain high-quality image on the product line.
2) A multistep inspection method is proposed to locate the can end, extract the measuring regions, and detect the defects in each region.
3) A graph-based superpixel grouping and selection algorithm is proposed for defect detection in the central circle region.
4) A multiscale ridge (valley) detection algorithm is adopted to seek the defects along the projection profiles for each annular measuring region.

The remainder of this paper is organized as follows. Section II introduces the apparatus system design. The proposed can-end inspection method is illustrated in Section III. Section IV gives the experiment results, and Section V concludes this paper.

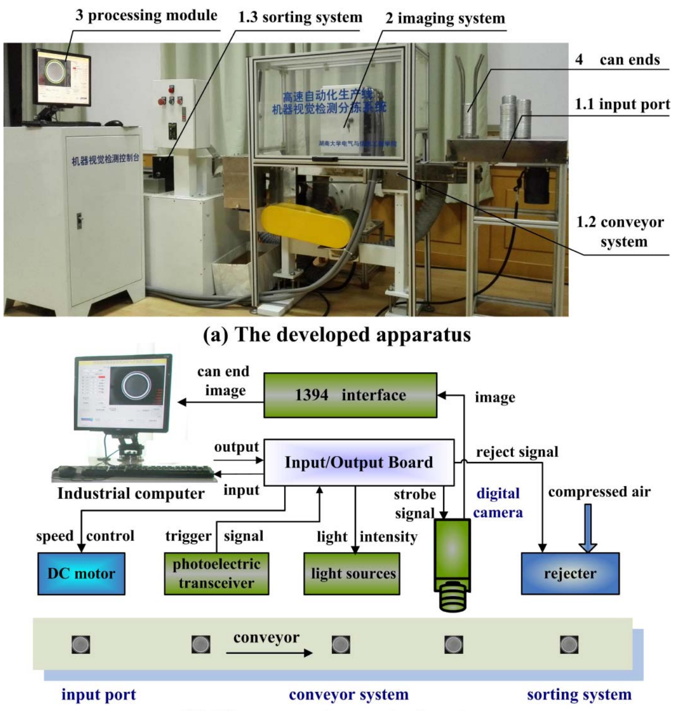
II. System Design of the Apparatus

In this section, system architecture of the can-end inspection apparatus is first introduced, and particular considerations are given to the illumination scheme design. Then, the image properties and the challenges for defect detection are investigated.

A. System Architecture

As shown in Fig. 2(a), the apparatus mainly consists of three parts: 1) electromechanical device; 2) imaging system; and 3) processing module. Among them, the electromechanical device is used to achieve automatic motion control and sorting of can ends. The imaging system is designed to acquire high-quality images for the fast moving test items. The processing module is adopted to process the obtained image and control the operation of the machine.

Furthermore, the electromechanical device is roughly divided into input port (1.1), conveyor system (1.2), and sorting system (1.3) according to their functions. The input port (1.1) is designed to input and separate can ends with the mechanical splitter controlled by a DC motor. The conveyor system (1.2) adopts a novel pneumatic approach for motion control. It connects to a vacuum pump, which generates a low air pressure area on the conveyor surface. Can ends are absorbed steadily on the conveyor by the pressure difference, and they are transmitted along the conveyor, which is driven by an AC motor. The sorting system (1.3) is used to classify can ends according to the inspection result. The defective can ends are separated off the conveyor by a rejecter, which uses compressed air pulse as a driving force. The defect-free products are exported orderly for the subsequent process. Other auxiliary components include ac/dc power converters and electromagnetic interference filters to improve power quality and reduce mechanical vibrations.

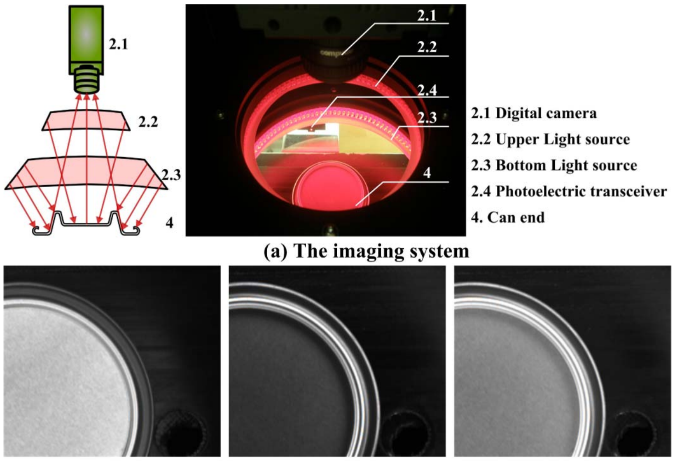


(a) The developed apparatus

(b) The apparatus control system

Fig. 2. System architecture of the developed inspection apparatus. (a) Developed apparatus. (b) Apparatus control system.

The imaging system (2) is mounted at the imaging station above the conveyor [see Figs. 2(b) and 3(a)]. It consists of a digital camera (2.1), light sources (2.2 and 2.3), and a photoelectrical transceiver (2.4). Here, can ends with the diameter of 99 mm are used for demonstration, and a camera resolution of 780 × 582 is sufficient to meet the inspection requirement. The processing module (3) mainly consists of an industrial computer with Core i7 processor and 8-GB RAM, and it collaborates with the different parts to work together. When the can end is transmitted to the imaging station, it triggers the photoelectric transceiver (2.4), and a trigger signal is sent to input/output board of the industrial computer, which then turns ON the light sources and commands the digital camera (2.1) to acquire an image of the test item. The obtained image is transmitted to the industrial computer through 1394 interface, where it is processed by the analysis and inspection software.



(a) The imaging system

(b) Upper light source (c) Lower light source (d) Two light sources

Fig. 3. (a) Imaging system and comparison of different illumination schemes. (b)–(d) Images obtained under a single upper light source, a single lower light source, and both light sources, respectively.

B. Imaging System and Illumination Scheme Design

In the obtained image, it is expected that the gray levels are uniform across the whole can-end area and the defects are distinct from the background. However, the complex 3-D can-end structure makes it very difficult to achieve a homogeneous illumination on the whole metallic surface. This inevitably degrades the image quality, because the gray levels are proportional to surface light intensity $L(X, Y)$ and exposure time t_e

$$I(x, y) \propto L(X, Y) * t_e. \qquad (1)$$

To circumvent this problem, a specific illumination scheme, consisting of two concentrically placed cone shape area light sources, is shown in Fig. 3(a). An upper light source is designed to mainly illuminate the central panel, while the larger lower one illuminates the peripheral regions. As shown in Fig. 3(b)–(d), with the upper light source, the gray level of central panel is much higher than other regions, and the boundaries between the peripheral regions become indistinguishable. This is contrary to the situation with the lower light source. When both light sources are used, their light fields overlay, which significantly improve the illumination uniformity and thus results in better image quality. In practical usage, the light intensities of both light sources are separately adjusted to find out the optimal blending ratio. In addition, considering the fast speed of conveyor, a short t_e is used to avoid motion blur.

C. Defects and Challenges

As shown in Fig. 3(d), a normal can end usually takes relatively uniform gray levels in each structure region under the specifically designed illumination. Meanwhile, the interregion differences appear obvious, which might bring difficulties to the subsequent discrimination of defects in images.

The difficulty of can-end inspection mainly arises from the diversity of surface defects, such as contaminants, scratches, and deformation, as shown in Fig. 4, which distribute across different structure regions and usually take varying shape or appearance. For easiness of analysis, the typical defects are divided into the following categories according to their causes.

1) *Foreign Contaminants [Fig. 4(a), (f), (k), (l), and (p)]:* They come from the clip-out, foreign matter, or other objects, which are pressed into the can end during manufacturing.
2) *Scratch [Fig. 4(b), (c), (h), and (m)]:* The protective surface coating is damaged by malfunctions of the tooling system, which makes the product directly contact with the steel.
3) *Compound Splash [Fig. 4(d)]:* Some sealing compound is wrongly splashed into the central panel.
4) *Dent [Fig. 4(g) and (q)]:* The defective shell press makes a hollow area on the edge area.
5) *Missing Compound [Fig. 4(n) and (o)]:* There is no sealing compound at some location of the seaming panel.
6) *Shape Distortion [Fig. 4(r)]:* The curler fails to curve some curl segments to the right position.
7) *Shape Deformation [Fig. 4(s)]:* External stress makes part of the curl area deviate severely from normal shape.
8) *Broken Edge [Fig. 4(j) and (t)]:* Part of the edge is damaged.

These defects will pollute the product [defects 1)–3)] or cause poor sealing even broken packages [defects 4)–8)].

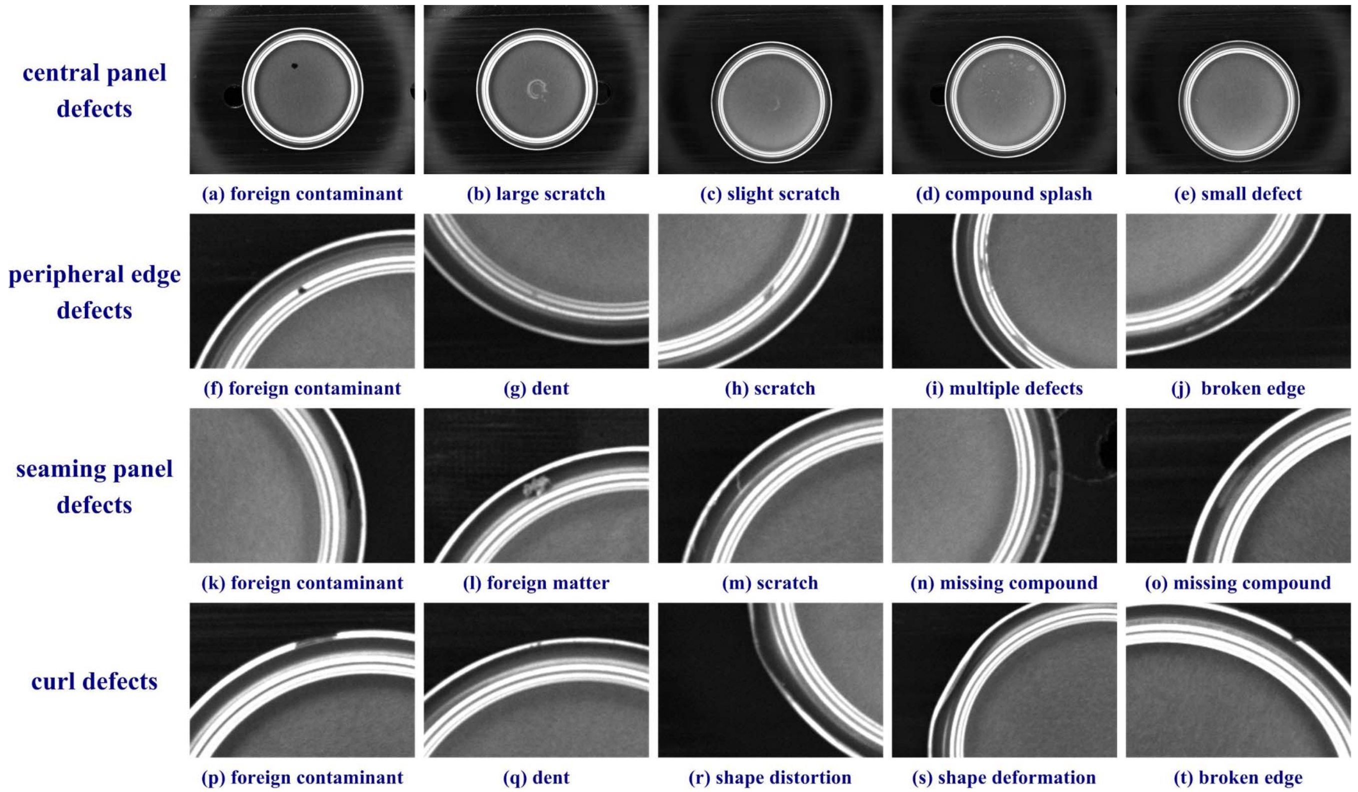
Reflected in the obtained images, contaminants, dents, and broken edge defects always present as dark areas. Contaminants can take various irregular shapes, while dents are usually square and the broken edge defects are ring area. Scratches, compound splash, and missing compound defects usually have higher gray level than the background. The scratches often take curve or line shape, while compound splash defects are approximately round. Moreover, shape deformation and distortion frequently cause local geometric structures different from their neighboring regions. In conclusion, properties, such as gray level, size, and shape, of these defects vary largely with the type of defect. Particularly, the background around each defect also varies among different regions, and it is even inhomogeneous inside the same region. As an example, in the central panel, the gray level of the marginal area is much lower than the central area.

III. Can-End Inspection Method

In this section, an image analysis pipeline for can-end inspection is presented. As shown in Fig. 5, the can-end object is first located and divided into multiple measuring regions. Then, different algorithms are developed and applied to detect defects across each region.

A. Can-End Location and Measuring Region Extraction

Due to random deviations of conveyor motion, can ends do not present at a fixed position in the obtained images, as revealed in Fig. 4(a)–(e). In addition, the defect properties might change with locations and the interregion gray-level distribution also varies largely as mentioned earlier. Therefore, it is necessary to locate the object and divide it into different measuring regions beforehand. Here, this is achieved



(a) foreign contaminant (b) large scratch (c) slight scratch (d) compound splash (e) small defect

(f) foreign contaminant (g) dent (h) scratch (i) multiple defects (j) broken edge

(k) foreign contaminant (l) foreign matter (m) scratch (n) missing compound (o) missing compound

(p) foreign contaminant (q) dent (r) shape distortion (s) shape deformation (t) broken edge

Fig. 4. Typical defects across different regions of the can end. Defects locating in (a)–(e) central panel, (f)–(j) peripheral edge, (k)–(o), seaming panel, and (p)–(t) curl regions.

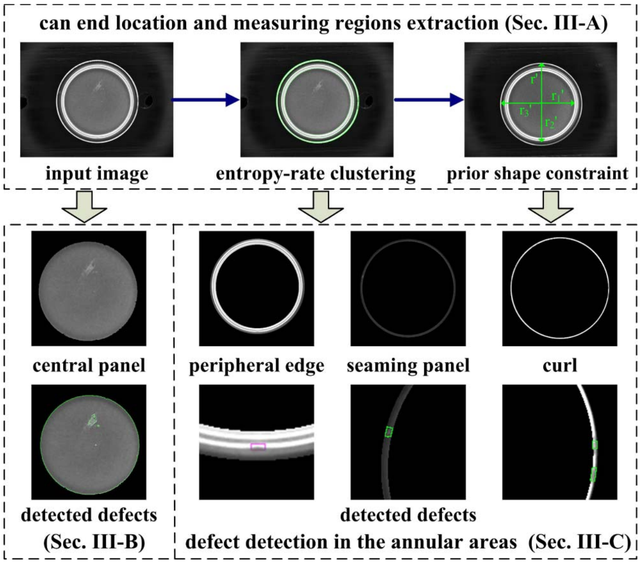


Fig. 5. Proposed multistep can-end inspection method.

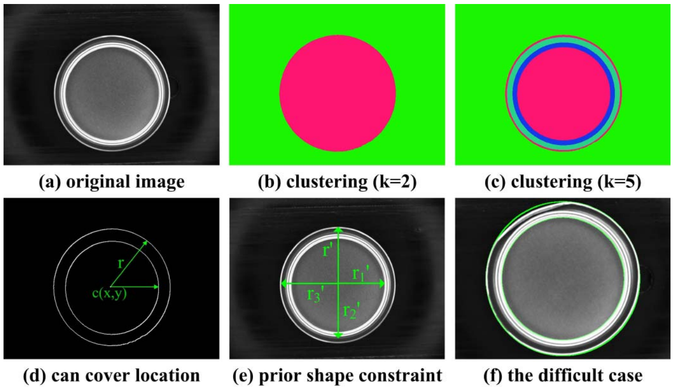


Fig. 6. Can-end location and measuring region extraction process. (a) Original image. (b) Clustering ($k = 2$). (c) Clustering ($k = 5$). (d) Can cover location. (e) Prior shape constraint. (f) Difficult case.

by combining an entropy-rate clustering algorithm with the prior shape constraint of can end, as shown in Fig. 6.

The entropy-rate clustering algorithm [17] is originally a superpixel generation method, which is able to oversegment the image into arbitrary k superpixels. The clustering process is based on the image graph representation $G_1 = (V, E, W)$, and the superpixels are generated by choosing an edge subset $A \subseteq E$, which partitions the graph into k disjoint subsets $S_1, S_2, \ldots, S_k$. This is achieved by optimizing an objective function F with respect to A [18], which comprises the entropy rate $H(A)$ of a random walk on $G_2 = (V, A)$ and a balancing term $B(A)$, that is

$$\max_{A \subseteq E} F(A) = H(A) + \lambda B(A) \quad \text{s.t.} \quad N_A \geq k \tag{2}$$

where N_A is the current cluster number determined by the edge subset A, and $\lambda \geq 0$ is the weight of the balancing term. The entropy rate favors the formation of the compact and homogeneous clusters. It can be written as a set function

$$H(A) = -\sum_i \mu_i \sum_j p_{ij}(A) \log(p_{ij}(A)) \tag{3}$$

with the transition probabilities $p_{ij}:2^E \to V$

$$p_{ij} = \begin{cases} \frac{w_{ij}}{w_i} & \text{if } i \neq j \text{ and } e_{ij} \in A \\ 0 & \text{if } i \neq j \text{ and } e_{ij} \notin A \\ 1 - \frac{\sum_{j:e_{ij} \in A} w_{ij}}{w_i} & \text{if } i = j \end{cases} \tag{4}$$

and the stationary distribution μ

$$\mu = (\mu_1, \mu_2, \ldots, \mu_{|V|}) = \left(\frac{w_1}{w_T}, \frac{w_2}{w_T}, \ldots, \frac{w_{|V|}}{w_T}\right) \tag{5}$$

where w_{ij} is the edge weight to measure the similarity between vertex pair (i, j), $w_i = \sum_{k:e_{ik} \in E} w_{ik}$, and $w_T = \sum_1^{|V|} w_i$. The balancing term $B(A)$ favors clusters with similar sizes

$$B(A) = H(Z_A) - N_A = -\sum_{i=1}^{N_A} p_{Z_A}(i) \log(p_{Z_A}(i)) - N_A \tag{6}$$

where $p_{Z_A}(i) = (|S_i|/|V|)$, $|S_i|$ and $|V|$ are the membership of the cluster i and the total vertex count separately. The objective function is submodular and can be optimized by a greedy heuristic algorithm. The algorithm starts with an empty set, that is $A = \varnothing$ (a fully disconnected graph), and sequentially adds edges to the set. At each iteration, it adds an edge that yields the largest gain until $N_A = k$.

When the entropy clustering method is used to segment the can-end image, an interesting phenomenon is that it is able to achieve accurate segmentation of the can end when k is very small, despite the varying background and various defects. This is mainly because both the can end and the background are relatively homogeneous, while their gray-level distributions are entirely different. A typical segmentation result is shown in Fig. 6(a)–(c). With $k = 2$, the can end is separated accurately from the background. When k is increased, the central panel, peripheral edge, seaming panel, and curl are gradually extracted.

In most cases, the segmentation result is accurate and sufficient to indicate the multiple measuring regions [see Fig. 6(c)]. However, sometimes, shape deformation and large defects might make the extracted regions deviate from ground truth. To ensure a reliable result, the segmentation result is postprocessed by taking advantage of the prior shape constraint. Can end and all its measuring regions are concentric circles or rings, so they can be represented by the center and inner and outer radii. For object location, the boundaries of the can end and central panel, which are obtained from the segmentation result with $k = 5$, are used to calculate the can-end center $c(x, y)$ and radius r by circle fitting algorithm, as shown in Fig. 6(d). To extract the multiple measuring regions, the inner and outer radii of each region are measured in a standard can-end image acquired beforehand. Then, in the current image, each measuring region is extracted by taking $c(x, y)$ as center, and the according radii are scaled by $s = (r/r')$, where r' is the can-end radius in the reference image. As shown in Fig. 6(f), the proposed algorithm can still accurately extract the measuring regions even the can end is severely deformed.

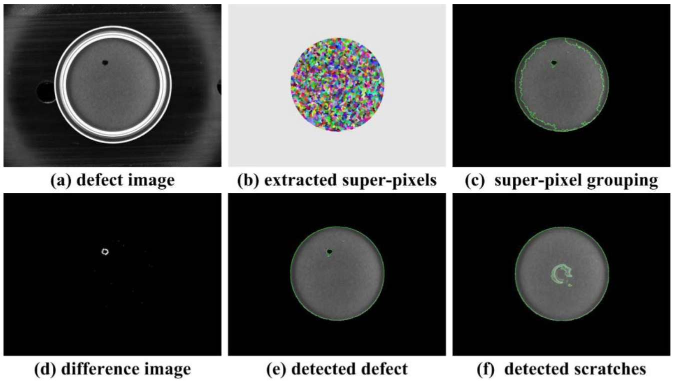


Fig. 7. Central panel defect detection process. (a) Defect image. (b) Extracted superpixels. (c) Superpixel grouping. (d) Difference image. (e) Detected defect. (f) Detected scratches.

Based on shape similarity, the extracted measuring regions generally fall into two categories, circular region (central panel) and annular region (peripheral edge, seaming panel, and curl). Sections III-B–III-D illustrate the proposed defect detection algorithms for each type of region, respectively.

B. Defect Detection in Central Panel

In the central panel region, the defects mainly include contaminants [Fig. 4(a)], scratches [Fig. 4(b) and (c)], compound splashes [Fig. 4(d)], and so on. Although their gray level, size, and shape are different from each other, a common property is that they often cover a small closed area with homogeneous gray level distinct from the central panel background. Considering that the superpixels are defined as inner homogeneous, closed, and boundary inherent small areas [19], so each defect can be seen as a collection of similar superpixels, which can be selected by detecting the comparatively large gray-level variations from their neighborhood. Hence, a superpixel grouping and selection algorithm is proposed for defect detection in the central panel, as shown in Fig. 7.

In superpixel generation process, the entropy-rate clustering algorithm is adopted to generate N_i superpixels ($N_i > 6000$), and the central panel superpixels are extracted by taking the region information as a mask [see Fig. 7(b)]. When N_i is large, very small defects can be detected, but the computation load is high. In superpixel grouping process, the graph segmentation method introduced in [20] is reformulated to group similar superpixels. First, a weighted region adjacency graph $G_h = (V_h, E_h, W_h)$ is constructed by taking each superpixel S_i as a node, and the weight of edge connecting two neighboring superpixels S_i and S_j is set to difference between their average gray value L_A, that is

$$W_h(S_i, S_j) = |L_A(S_i) - L_A(S_j)|. \tag{7}$$

Then, two adjacent components $C_1, C_2 \subseteq V_h$ in the graph are merged if their interdifference, i.e., the minimum weight connecting C_1 and C_2, is smaller than the minimum internal difference M_{Int}, which can be formulated as

$$M_{\text{Int}}(C_1, C_2) = \min(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2)) \tag{8}$$

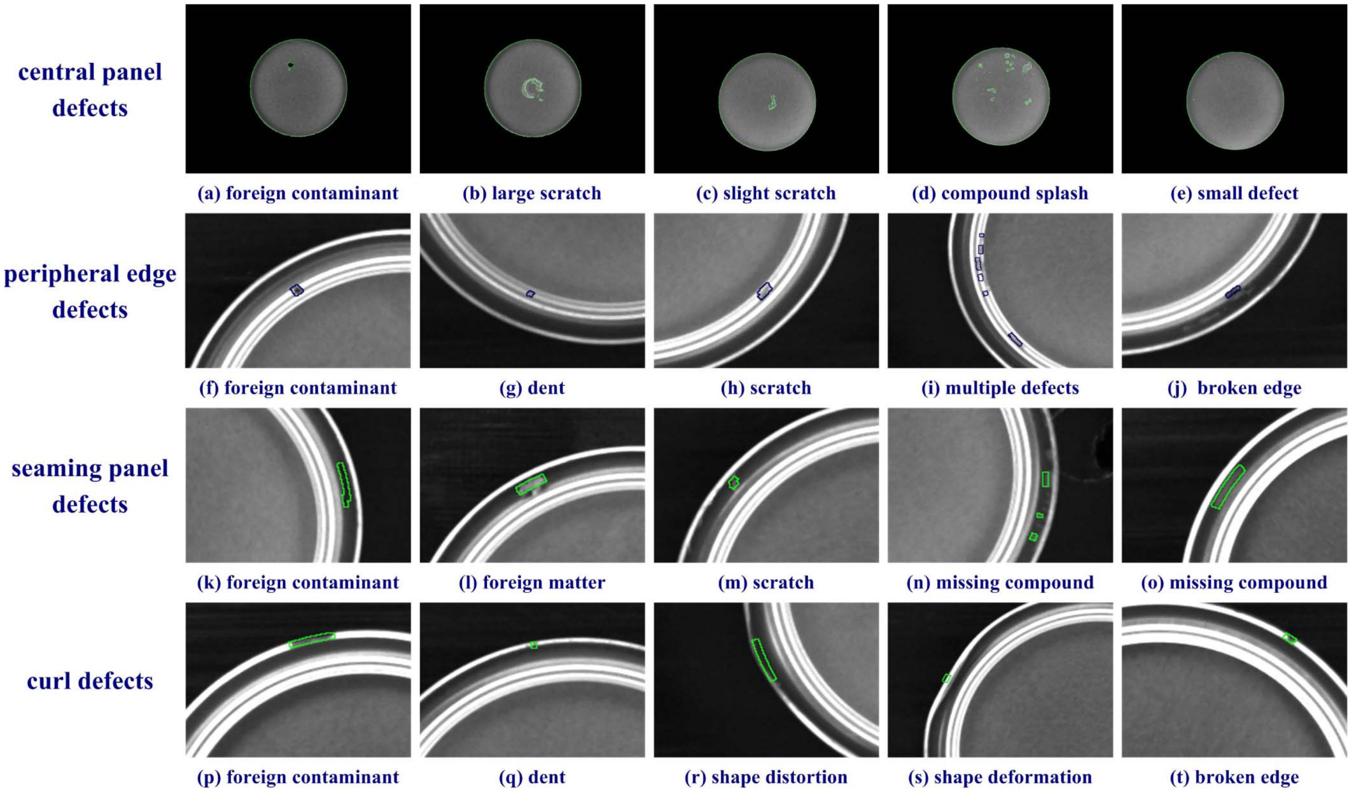


Fig. 8. Defect detection results in (a)–(e) central panel, (f)–(j) peripheral edge, (k)–(o) seaming panel, and (p)–(t) curl, separately.

where $\text{Int}(C)$ is the maximum weight in the minimum spanning tree of a cluster C. $\tau(C) = (k_2/|C|)$ is a bias term that increases the internal difference for small clusters, where $|C|$ is the superpixel counts contained in the cluster C. With this approach, the central panel superpixels are grouped into p inner homogeneous regions $R_1, R_2, \ldots, R_p$ controlled by the parameter k_2, as shown in Fig. 7(c). A larger k_2 reduces p and the postprocessing complexity; however, it might miss some small defects. Considering that the defect size is usually small, the largest region, which also has very small inner difference, is taken as the defect-free background.

The following step is to select the defective regions from the merging result. A specific index is defined to assess the local gray-level variations for each region. First, the central panel region is smoothed by a binomial filter with coefficients

$$M_{ij} = \frac{1}{2^{m+n-2}} * \frac{(m-1)!}{i! * (m-1-i)!} * \frac{(n-1)!}{j! * (n-1-j)!} \tag{9}$$

where m and n are the height and the width of the filter, respectively. More details are smoothed with larger m and n, and $m = n = 13$ is used here. The difference between the original and smoothed image is calculated, as shown in Fig. 7(d). For each merged region R_i, the defect index D_i is set to sum of the difference above certain threshold T_a in the region

$$D_i = \sum_{q \in R_i, |I - I \otimes M|_q > T_a} |I - I \otimes M|_q \tag{10}$$

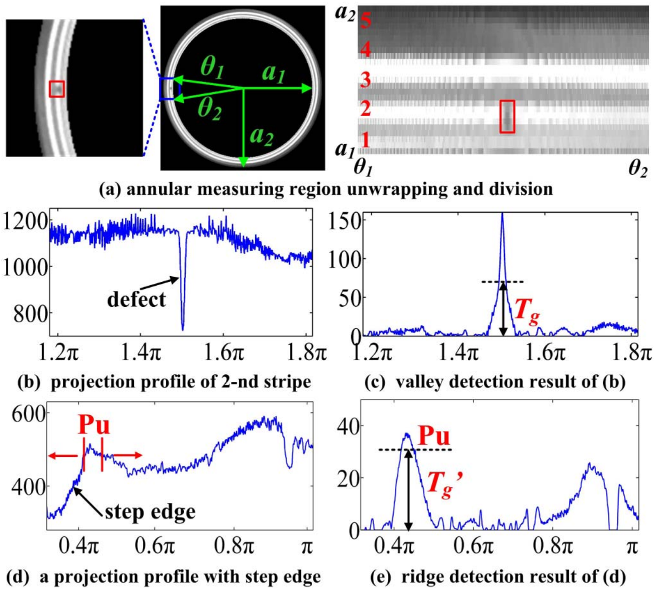
where T_a is used to suppress the background noise. Then, for each nonbackground region R_i, it is labeled as defect if $(D_i/|R_i|)$ is above a small threshold T_b [see Fig. 7(e)], where $|R_i|$ is the superpixel counts in R_i.

With the proposed algorithm, the detection results of the typical central panel detects are shown in Fig. 8(a)–(e). As observed, it is effective to detect and segment various defects with a different gray level, shape, and size.

C. Defect Detection in Annular Measuring Regions

In the three peripheral annular regions, the defects are diverse, as shown in Fig. 4(f)–(t). Moreover, the complex structures make the background gray-level distribution vary remarkably with location, as revealed in Fig. 9(a). These factors dramatically increase the difficulty of defect detection. Fortunately, the pixels around a circle centered at $c(x, y)$ are expected to have homogeneous gray level due to the geometric symmetry of can-end structure. Hence, the various surface defects in these regions can be generally characterized as local gray-level abnormalities around the circumference. Based on this, a multiscale ridge (valley) detection algorithm is proposed to detect the defects by finding abrupt gray-level changes along projection profiles of these narrow annular regions.

In Fig. 9, the whole defect detection algorithm for the annular measuring regions is shown. First, each annular region defined by its inner radius a_1 and outer radius a_2 is unwrapped to form a subimage [see Fig. 9(a)] in polar coordinates. The subimage is then divided into several isolated or partially overlapped small rectangular stripes [marked with 1–5 in Fig. 9(a)], which have a small constant width w_a ($2 \leq w_a \leq 7$) to accommodate only slight gray-level variation along the



(a) annular measuring region unwrapping and division

(b) projection profile of 2-nd stripe

(c) valley detection result of (b)

(d) a projection profile with step edge

(e) ridge detection result of (d)

Fig. 9. Defect detection process for the annular measuring regions. (a) Demonstrated peripheral edge is unwrapped and then divided into five small rectangular stripes. (b) and (c) Valley (ridge) detection on the projection profile of each stripe to find the defects. (d) and (e) Projection profile containing step edge (obtained from the seaming panel) and its ridge detection result. Pu denotes the wrongly detected defect area caused by a step edge. T_g and T_g' are the thresholds used for peripheral edge and seaming panel, respectively.

radial direction. For each separate rectangular strip, its projection profile is obtained by accumulating gray levels of the w_a pixels in the radius direction. As shown in Fig. 9(b), the defect appears as an obvious local abrupt gray-level variation on the projection profile, which makes it possible to locate the defect with postprocessing signal analysis. However, the obtained 1-D signals are very noisy, and their mean gray-level or background often drifts with location on the same stripe and varies remarkably across different stripes. These factors remain big challenges to the conventional algorithms, such as threshold binarization, image derivative, and peak detection.

To solve this problem, a multiscale ridge or valley detection algorithm is proposed to find the abrupt gray-level variation on the projection profiles. The ridge (valley) was defined as a curve, whose points are local maxima (minimum) in $N-1$ dimensions for an N variable function [21]. As observed in Fig. 9(b), the defects on the projection profile can be perfectly modeled with the ridge or valley definition. To be more specific, a common property of the defects is that they are local minimum or maximum and generally have a convex Gaussian shape. Motivated by the well-established ridge detection theory and its various applications [22], [23], we propose to detect the defects based on a 1-D ridge (valley) strength measure of the projection profile. To be simplified, a γ-normalized second-order Gaussian filter

$$h(x,\sigma)=\sigma^{2\gamma}\frac{\partial}{\partial x^2}G(x,\sigma)=\frac{1}{\sqrt{2\pi}}\left(\frac{x^2}{\sigma^3}-\frac{1}{\sigma}\right)e^{-\frac{x^2}{2\sigma^2}} \tag{11}$$

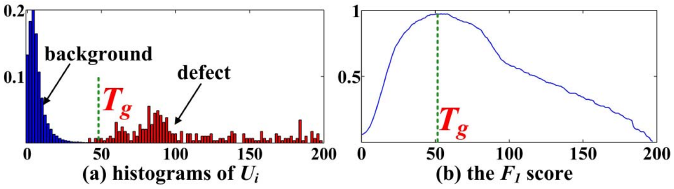
with σ as the scale parameter is adopted, where the normalization coefficient $\gamma = 1$ is used to ensure scale-invariant. Obviously, $h(x,\sigma)$ gets a positive response for the local minimum, so it is suitable for valley detection. As for ridge detection, its negative filter is used instead, that is

$$h_r(x,\sigma) = -h(x,\sigma). \tag{12}$$

To account for the defect width variation, the multiscale framework introduced by Lindeberg [24] is followed. In the ridge (valley) detection process, the projection profile P_i of each stripe is first convolved with either $h(x,\sigma)$ or $h_r(x,\sigma)$ under different scales, and then, the maximum response under all scales is taken as the final result, that is

$$U_i = \max_{\sigma_{\min}\le\sigma\le\sigma_{\max}} P_i \otimes K \quad K \in \{h(x,\sigma), h_r(x,\sigma)\}. \tag{13}$$

Here, the minimum and maximum scales $\sigma_{\min}$ and $\sigma_{\max}$ are set to half of the minimum distinguishable and maximum defects size, respectively. As observed in Fig. 9(c), U_i usually takes a large value at the defect location, while it is comparatively small for the extremely noisy background and not affected by the slow or even large background drifting. Based on this, a simple threshold on the ridge or valley response is sufficient to sift out the real defect. As shown in Fig. 9(c), a threshold T_g is adopted to discriminate desired defects from noisy interferences, and the part with $U_i > T_g$ is taken as a defect segment.



(a) histograms of U_i

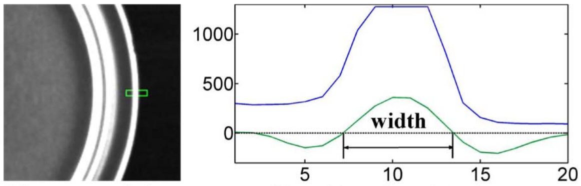
(b) the F_1 score

Fig. 10. Threshold selection process for T_g, with x-axis standing for U_i. (a) Histograms of U_i in the background area (blue) and the defect area (red). (b) F_1-score with different thresholds.

Owing to the distinct gray-level distribution in different measuring regions, a unique and fixed T_g is used for each region. The thresholds are mainly chosen based on a prior statistical analysis of the U_i distributions, and a F_1-score maximization criterion is adopted to select the optimal threshold. Taking the peripheral edge as an example, the histograms of U_i in the background and defect area are separately obtained through statistical approach beforehand. As observed in Fig. 10(a), there is a large margin between the defect and background distribution. Then, the F_1-score (see Section IV-A4) is calculated with every possible threshold. Finally, the threshold maximizing the F_1-score is used as the optimal threshold [see Fig. 10(b)].

In extremely rare cases, the step edge also cause a strong response to the ridge (valley) detector, which might be wrongly determined as a defect, as shown in Fig. 9(d) and (e). To avoid such error, the asymmetric structure of step edge is further exploited. To be specific, for each defect segment, if the difference between average P_i around U_i peak and the left or right local background is below a global threshold T_d, it is taken as step edge other than defect.

For the peripheral edge and curl, normal gray level is very high and the defects are dark regions, as shown



(a) measure window **(b) curl boundary detection**

Fig. 11. Curl width measurement process. (a) Measure window. (b) Curl boundary detection.

in Fig. 4(f)–(j) and (p)–(t), so valley detection is adopted and the detection results are shown in Fig. 8(f)–(j) and (p)–(t), respectively. As observed, the main defects in peripheral edge, including contaminants, dents, scratches, and broken edge, are accurately detected by the proposed algorithm. In the curl region, two special defects are shape distortion and deformation, as shown in Fig. 4(r) and (s). They also cause abrupt variations in the projection profiles, so the proposed algorithm is also able to detect these defects. In the seaming panel, except contaminants, all other defects, including foreign matter, compound missing, and scratch, are with a higher gray level than the background, as shown in Fig. 8(k)–(o). So both ridge and valley detector are used, and the defect detection results are shown in Fig. 8(k)–(o). As shown, all the typical defects are correctly detected.

D. Curl Width Measurement

Another defect is the double ends, which arise from a misoperation that multiple can ends are wrongly punched together in the manufacturing process. This defect is usually accompanied with a clear increase of the curl width. Therefore, the curl width measurement is important for detecting such defect.

Considering the curl width seldom has large change with location, it can be sampled at multiple positions, and then, their mean value is taken as the final measure result. To achieve this, several narrow rectangle windows ($3 \leq w_r \leq 7$) are placed across the curl area with their length directions along the can-end radius, as shown in Fig. 11(a), and the curl width is obtained by measuring the distance between the boundaries in each window. Inspired by the well-studied Canny edge detector, the second-order derivative of Gaussian kernel in (12) is directly convolved with the projection profile, and the distance between the two zero crossing points is taken as the curl width. If the measure result is above the maximum allowable curl width, it is considered as defective.

IV. System Test and Implementation

In this section, the defect detection algorithms and apparatus performance are tested with in-line obtained images, and a series of indexes is defined for quantitative evaluation. For comparison, several conventional methods are also implemented and adopted in the experiments.

In our experiments, 954 can-end images acquired with our apparatus, including 454 typical defect images and 500 randomly selected normal images, are used for testing.

A. Quantitative Evaluation

Within the apparatus system design, the influence of imaging and sorting modulus on quality control is minimized. Therefore, the apparatus inspection performance is mainly determined by the image analysis algorithms. To further understand the effect of each step in the image processing pipeline, the inspection performance is assessed from different perspectives (i.e., object location, defect detection, and segmentation) using different indexes.

TABLE I

Quantitative Indexes for Can-End Location

Index[a]	Ave. E_c	Max. E_c	σ_{E_c}	Ave. E_r	σ_{E_r}
Value (pixel)	0.3	0.9	0.14	0.4	0.22

[a] Ave. and Max. are abbreviations for average, maximum, respectively. σ indicates the standard deviation.

1) Can-End Location: The location of randomly distributed can-end objects from the in-line images is the basis for further defect detection. Its performance is evaluated in terms of center deviation E_c and radius mismatching E_r. E_c is defined as the distance between the located center $c(x, y)$ and its true position $c_t(x, y)$, i.e., $E_c = ||c(x, y) - c_t(x, y)||_2$. E_r is set to difference between the calculated can-end radius r and its true length r_t, i.e., $E_r = r - r_t$. In the experiments, the true can-end location $c_t(x, y)$ and radius r_t of each image are obtained by manual measurement, and the statistics of E_c and E_r are shown in Table I. As observed, both the average E_c and E_r are very small; besides, the maximum E_c is <1 pixel, this is sufficient to ensure an accurate extraction of the measuring regions.

2) Defect Detection: To adapt this requirement, the terms defective and defect-free are used to classify the defect detection results, whereas qualified and unqualified are adopted for the ground truth. Among the inspection results, true positive (TP), true negative, false positive (FP), and false negative (FN) correspond to the count of defective unqualified, defect-free qualified, defective qualified, defect-free unqualified products, respectively. Considering that the qualified products are usually with much larger numbers than the unqualified ones, precision (P_r), recall (R_c), and accuracy (A_{cc}) are used to investigate the defect detection performance

$$P_r = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad R_c = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

$$A_{\text{cc}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{15}$$

where A_{cc} is the proportion of correct inspection results. P_r and R_c correspond to the percentages of the correctly detected unqualified products in total defective and unqualified products, respectively. Obviously, it is desired that all these indexes are 100%.

The defect detection performance is verified using the obtained experiment data, and the quantitative indexes are listed on columns 2–4 of Table II with each row corresponding to different measuring regions. As seen, most defects in the central panel are easily detected with the proposed algorithm, though some exceptions might happen to tiny and ambiguous defects or the very noisy interferences are occasionally discriminated as defects. The errors are mainly due to the

TABLE II

QUANTITATIVE INDEXES FOR DEFECT DETECTION AND SEGMENTATION

Measuring regions	Defect detection			Defect segmentation	
	$P_r(\%)$	$R_c(\%)$	$A_{cc}(\%)$	$D_{P_r}(\%)$	$D_{R_c}(\%)$
central panel	98.13	99.36	99.58	90.45	87.22
peripheral edge	100	100	100	86.56	93.98
seaming panel	94.23	99.04	99.27	71.06	89.50
curl	100	98.93	99.90	89.77	85.59
curl width	100	100	100	–	–
total	99.60	99.39	99.48	86.43	90.50

largely varying gray-level distribution of the background. For the peripheral edge, a perfect measure is achieved owing to the excellent noise suppression and local structure detection abilities of multiscale ridge (valley) detection algorithms. In the seaming panel, there are quite a few FP errors caused by shape deformation defects in Fig. 4(s). The main reason is that part of the curl area is wrongly taken as a seaming panel during measuring region extraction, which causes abrupt gray level changes in projection profiles. Moreover, the small defects near the region boundaries are occasionally missed. In the curl area, some small defects are not detected with tradeoff thresholds. The main difficulty is the extremely noisy projection profile caused by largely varying gray levels in this area. In addition, the proposed algorithm is very sensitive to shape deformations or distortions, and such defects can be easily detected even they are small. In the final inspection results, the FP errors are accumulated if the product is qualified, while the FN errors increase when there are defects in only one region. Otherwise, the errors are eliminated.

With the proposed method, all the critical and major defects, including contaminants, scratches, dents, missing compounds, broken edges, and so on, are easily detected. The main errors are the small, ambiguous defects or the noisy background.

3) Defect Segmentation: To give a quantitative evaluation of the segmentation performance, we still follow the receive operating characteristic theory to adopt the precision and recall indexes. However, different from the previous defect detection, TP is set to the pixel counts correctly labeled as defective, while FN and FP are the pixel numbers wrongly labeled as background and defects, respectively. Accordingly, we use D_{P_r} and D_{R_c} to represent the precision and recall indexes for segmentation here. The performance statistics for each measuring region are presented in columns 5 and 6 of Table II. For the seaming panel, both ridge and valley detection algorithms are used to locate the defects, which causes many FPs and a low D_{P_r} due to the unexpected responses of step edges. As for other measuring regions, acceptable defect segmentation accuracies are achieved with the proposed algorithms.

4) Parameter Sensitivity and Algorithm Robustness Analysis: As mentioned earlier, thresholds used in the proposed method are chosen by maximizing their according F_1-score, that is

$$F_1 = \frac{2 \times P_r \times R_c}{P_r + R_c}. \tag{16}$$

Other parameters are mainly chosen based on the experiments, and computation load is also a major consideration. In general, each parameter can be chosen from a large range without affecting the performances obviously.

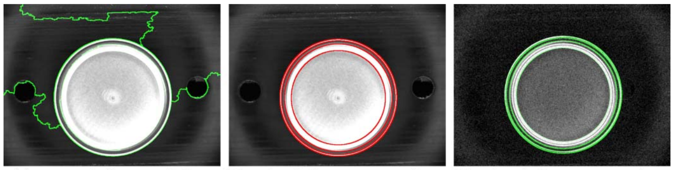

(a) segmentation result for high brightness image **(b) extracted measure regions for high brightness image** **(c) extracted measure regions for noisy image**

Fig. 12. Robustness analysis of the can-end location and measuring region extraction algorithm. (a) Segmentation result for high brightness image. (b) Extracted measure regions for high brightness image. (c) Extracted measure regions for noisy image. Images (a) and (b) are obtained under extremely high light intensity. Image (c) is corrupted by Gaussian noise with $m_n = 0$ and $\sigma_n = 29$.

Another important concern is the robustness of the proposed algorithms under different circumstances. Can-end location is the prerequisite of defect detection, so its robustness is the most important. To assess this performance, the images obtained with extreme illumination conditions and the Gaussian noise corrupted images are used for experiments. For these images, the segmentation results might deviate severely from expectation, as shown in Fig. 12(a). The background is partitioned into multiple regions, even worse, the peripheral edge is merged with central panel. To avoid errors, the background label is obtained by combining the superpixel labels of image borders. In addition, the prior knowledge of the radius ranges is used to remove false circle fitting result. As shown in Fig. 12(b) and (c), our algorithm can still correctly extract the measuring regions for images obtained under extreme illumination conditions and noise corrupted images when $\sigma_n \leq 29$. As for the defect detection algorithms, their tolerance to noise is limited, but they are insensitive to the light intensity variations.

5) Speed Analysis: The inspection speed is mainly determined by computation complexity of the proposed algorithms. Among them, the most computation intensive part is entropy-rate clustering, which is used in both can-end location and superpixel generation. Due to its iterative solution, only one clustering is conducted. When the superpixel count is equal to N_i, the result is saved for central panel defect detection. Then, the clustering proceeds until the can end is segmented. The algorithm implementation is optimized in many ways, for example, the graph weight is calculated by accessing a lookup table. Besides, the pixel size is not large, the clustering process takes 0.5 s averagely, and the whole quality measure process is accomplished in 0.7 s with our apparatus.

B. Comparison With Other Methods

Although there are a few methods addressing the whole can-end inspection issue, parts of our pipeline can be replaced with other methods. Here, some conventional algorithms are implemented for comparison but only on separate steps.

1) Edge Detection for Can-End Location: The boundary between the test item and the background is usually very strong, which makes it possible to locate the object with an edge detection algorithm. For comparison,

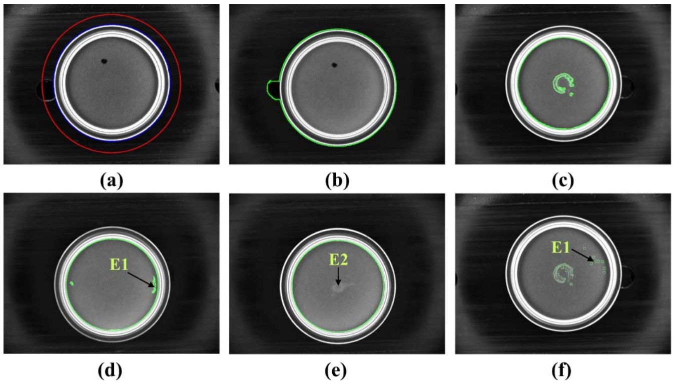


Fig. 13. Experiment results with other methods. (a) Can-end location result with method 2. Red line: initialized circle contour with radius $r_i = 250$. Blue line: located boundary. (b) Can-end location result using method 3 with the parameters $k_2 = 200\,000$ and $S_{\min} = 8000$. (c)–(e) Central panel defect detection results using method 3 with the parameters $k_2 = 81\,500$ and $S_{\min} = 50$. (f) Central panel defect detection results using method 5. $E1$ and $E2$ denote the falsely detected defects and the undetected defects, respectively.

we adopt a conventional Canny edge detector to extract the boundary. Subsequently, a circle-fitting model is applied to find its center and radius. With this method, the computation load is low, and acceptable accuracy is achieved for most cases. However, when the curl is defective, as shown in Fig. 4(p)–(t), the location errors appear large.

2) Active Contour Model-Based Location: As a popular boundary extraction algorithm, the active contour model can also be applied to can-end location. Here, the fast reinitialized free level set method proposed in [25] is adopted, and the location result is shown in Fig. 13(a). This method is able to achieve a subpixel location accuracy. However, its performance is very sensitive to the initialized contour. Moreover, the convergence time might change between 5 and 20 s even using an optimal circle initialization, and this is mainly because the can end distributed randomly within a large range.

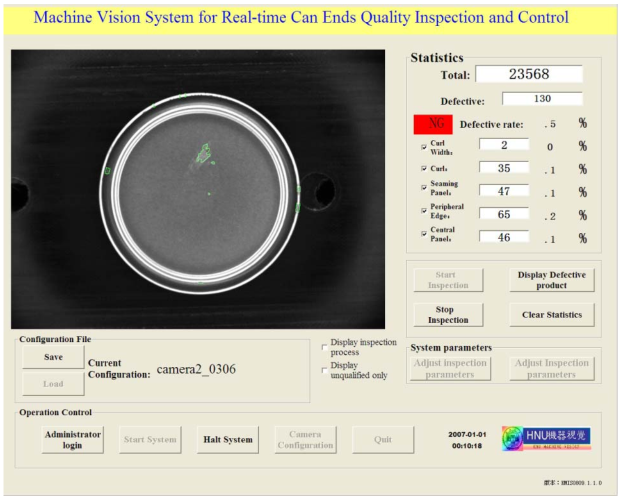
3) Graph-Based Segmentation Method: Theoretically, the graph segmentation model presented in (8) can be used for not only can-end location but also defect detection in the central panel. For can-end location, with large control parameter k_2 and $S_{\min}$ (the allowed minimum size), the can end is segmented from the background. However, as shown in Fig. 13(b), this method is easily influenced by background variations. As for defect detection, the main difficulty is the choice of optimal control parameter, which varies in a large range for different test items. With a tradeoff choice, some background is taken as defects [Fig. 13(d)], while some defects are undetected [Fig. 13(e)]. Besides, the inhomogeneous background, especially the darker area near boundaries, also causes detection errors.

4) Background Subtraction-Based Defect Detection: Similar to (10), difference between the original image and its smoothed regions can also be utilized for defect detection. However, the main difficulty is the selection of a binarizing threshold. As shown in Fig. 7(d), this method only detects the boundaries of large defects. Another drawback is that many backgrounds are wrongly taken as defects.

TABLE III

PERFORMANCE STATISTICS OF OTHER METHODS

Method	Can-end location				Defect detection	
	Ave. E_c	σ_{E_c}	Ave. E_r	σ_{E_r}	$P_r(\%)$	$R_c(\%)$
1	0.8	0.32	0.7	0.21	–	–
2	0.1	0.1	0.2	0.1	–	–
3	1.4	0.45	1.2	0.37	84.23	92.48
4	–	–	–	–	98.44	94.65
5	–	–	–	–	95.72	81.72


Machine Vision System for Real-time Can Ends Quality Inspection and Control

Statistics

Total: 23568

Defective: 130

NG Defective rate: .5 %

☑ Curl Width:	2	0	%
☑ Curl:	35	.1	%
☑ Seaming Panel:	47	.1	%
☑ Peripheral Edge:	65	.2	%
☑ Central Panel:	46	.1	%

Configuration File

Current Configuration: camera2_0306

□ Display inspection process

□ Display unqualified only

Operation Control

2007-01-01
00:10:18

版本：HMIS0809.1.1.0

Fig. 14. Software user interface.

5) Superpixel Classification for Defect Detection: The defects can also be detected by directly classifying the superpixels. For comparison, a support vector machine is used to classify a 9-bin histogram feature of the superpixel with an intersection kernel. As shown in Fig. 13(f), this method is able to detect a majority of middle-sized defects. The main problem is that the defect-free superpixels are dominant, which provides a strict requirement on the classifier design. Even a very small classifying error might result in serious detection errors.

The same experiment data are used to assess the performances of the above methods. The previously defined indexes E_c and E_r are adopted to investigate the can-end location accuracy of methods 1–3, and the defect detection performances of methods 3–5 are evaluated by P_r and R_c. The according performance statistics are presented in Table III. Among the location methods, active contour model can achieve better accuracy than our method. However, its long and largely varying convergence time makes it unsuitable for industrial applications. Compared with the traditional defect detection approaches, the proposed method can achieve better defect detection and segmentation accuracy simultaneously.

C. System Implementation

The inspection software is implemented with Halcon machine vision library [26], which provides many deeply optimized image-processing functions. This greatly reduces the time consumed by an inspection process. The designed user interface is shown in Fig. 14, the left part presents the obtained image of current test item, and the right part presents the statistics of the measure results. Parameter sets can be adjusted by a pop up interface when clicking the system parameters button at the lower right corner.

V. Conclusion

In this paper, we have presented a machine vision apparatus for real-time can-end inspection. A multistep image analysis pipeline is proposed to detect various defects across different parts of can-end surface according to the quality control requirements. First, the can end is located by combining entropy-rate clustering algorithm with prior shape constraint. Then, the separate algorithms are developed to detect the defects in different measuring regions. In central panel, a superpixel clustering and selection algorithm is proposed for defect detection. In the other three annular measuring regions, a multiscale ridge (valley) detection algorithm is adopted to seek the defects along their projection profiles.

After more than one year's test in production line, our apparatus can achieve a very high accuracy for can ends with the diameter between 55 and 153 mm. Its accuracy and robustness are sufficient to satisfy the harsh industrial requirements. Moreover, the developed algorithms can also be extended to many other industrial inspection applications.

Acknowledgment

The authors would like to thank the editors and anonymous reviews for their help in improving this paper.

References

[1] K. Yamazaki, J. Han, T. Otsuka, T. Hasegawa, and S. Nishiyama, "Tooling system design for forming aluminum beverage can end shells," *J. Mech. Design*, vol. 133, no. 11, p. 114502, 2011.

[2] R. G. Forrest and T. Turner, "Can end for a container," U.S. Patent 7 000 797 B2, Feb. 21, 2006.

[3] Ball Corp. (2014). *How Ball Makes Beverage Ends*. [Online]. Available: http://www.ball.com/na/solutions/markets-capabilities/learning-center /manufacturing

[4] M. L. Castor, C. L. Bartels, and D. R. Miller, "Can end tab sensing apparatus," U.S. Patent 4 932 823, Jun. 12, 1990.

[5] P. W. Gold and B. A. Moen, "Monitor and control assembly for use with a can end press," U.S. Patent 4 939 665, Jul. 3, 1990.

[6] D.-M. Tsai, S.-C. Wu, and W.-Y. Chiu, "Defect detection in solar modules using ICA basis images," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 122–131, Feb. 2013.

[7] Ball Corp. (2014). *Innovation is in our DNA*. [Online]. Available: http://www.ball.com/na/vision/innovation/packaging/technology

[8] Applied Vision Corp. (2015). *Shell Inspection System*. [Online]. Available: http://appliedvision.com/pdfs/products/genius/ShellProdSR1 .pdf

[9] P. Marino, V. Pastoriza, M. Santamaria, and E. Martinez, "Can end inspection using neuro-fuzzy modeling," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Dec. 2004, pp. 926–930.

[10] B. Feng, S.-X. Guo, F.-L. Zhang, C.-J. Zhu, and L. Wang, "Binary-channel can covers defects detection system based on machine vision," in *Proc. Int. Conf. Anti-Counterfeiting, Secur. Identificat. (ASID)*, Aug. 2012, pp. 1–4.

[11] Q. Li and S. Ren, "A real-time visual inspection system for discrete surface defects of rail heads," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 8, pp. 2189–2199, Aug. 2012.

[12] X. Li, S. K. Tso, X.-P. Guan, and Q. Huang, "Improving automatic detection of defects in castings by applying wavelet technique," *IEEE Trans. Ind. Electron.*, vol. 53, no. 6, pp. 1927–1934, Dec. 2006.

[13] H. Feng, Z. Jiang, F. Xie, P. Yang, J. Shi, and L. Chen, "Automatic fastener classification and defect detection in vision-based railway inspection systems," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 4, pp. 877–888, Apr. 2014.

[14] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta, "Automatic defect detection on hot-rolled flat steel products," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 3, pp. 612–621, Mar. 2013.

[15] H. Gao, C. Ding, C. Song, and J. Mei, "Automated inspection of E-shaped magnetic core elements using K-tSL-center clustering and active shape models," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1782–1789, Aug. 2013.

[16] Y. Gan and Q. Zhao, "An effective defect inspection method for LCD using active contour model," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 9, pp. 2438–2445, Sep. 2013.

[17] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2097–2104.

[18] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 99–112, Jan. 2014.

[19] S. Liu, L. Zhang, Z. Zhang, C. Wang, and B. Xiao, "Automatic cloud detection for all-sky images using superpixel segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 354–358, Feb. 2015.

[20] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[21] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, "Ridges for image analysis," *J. Math. Imag. Vis.*, vol. 4, no. 4, pp. 353–373, 1994.

[22] C. Xiao, M. Staring, Y. Wang, D. P. Shamonin, and B. C. Stoel, "Multiscale bi-Gaussian filter for adjacent curvilinear structures detection with application to vasculature images," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 174–188, Jan. 2013.

[23] T. Chen, Y. Wang, and C. Xiao, "An apparatus and method for real-time stacked sheets counting with line-scan cameras," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1876–1884, Jul. 2015.

[24] T. Lindeberg, "Scale-space for discrete signals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 3, pp. 234–254, Mar. 1990.

[25] K. Zhang, L. Zhang, H. Song, and D. Zhang, "Reinitialization-free level set evolution via reaction diffusion," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 258–271, Jan. 2013.

[26] MVTec Software GmbH. (2009). *Halcon/HDevelop 12.02 Reference Manual*. [Online]. Available: http://www.halcon.com/ download/documentation

Tiejian Chen received the B.Eng. and M.S. degrees in control engineering from Hunan University, Changsha, China, in 2008 and 2011, respectively, where he is currently pursuing the Ph.D. degree.

His current research interests include imaging instrument and image processing.

Yaonan Wang received the B.S. degree in computer engineering from East China Science and Technology University, Shanghai, China, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1994, respectively.

He was a Senior Humboldt Fellow in Germany from 1998 to 2000. From 2001 to 2004, he was a Visiting Professor with the University of Bremen, Bremen, Germany. He has been a Professor with Hunan University since 1995. His current research interests include intelligent control, image processing, and computer vision system for industrial applications. He is corresponding author of the paper.

Changyan Xiao received the B.Eng. and M.S. degrees in mechanical and electronic engineering from the National University of Defense Technology, Changsha, China, in 1994 and 1997, respectively, and the Ph.D. degree in biomedical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2005.

He was a Visiting Post-Doctoral Researcher with the Division of Image Processing, Leiden University Medical Center, Leiden, The Netherlands, from 2008 to 2009. Since 2005, he has sequentially been an Associate Professor and a Full Professor with the College of Electrical and Information Engineering, Hunan University, Changsha. His current research interests include medical imaging, machine vision and embedded instrument.

Q. M. Jonathan Wu (M'92–SM'09) received the Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990.

He was with the National Research Council of Canada, Ottawa, ON, Canada, from 1995 to 2005, where he became a Senior Research Officer and a Group Leader. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. He has authored over 300 peer-reviewed papers in computer vision, image processing, intelligent systems, robotics, and integrated microsystems. His current research interests include 3-D computer vision, active video object tracking and extraction, interactive multimedia, sensor analysis and fusion, and visual sensor networks.

Dr. Wu holds the Tier 1 Canada Research Chair of Automotive Sensors and Information Systems. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Cognitive Computation*, and the *International Journal of Robotics and Automation*. He has served on the technical program committees and international advisory committees for many prestigious conferences.