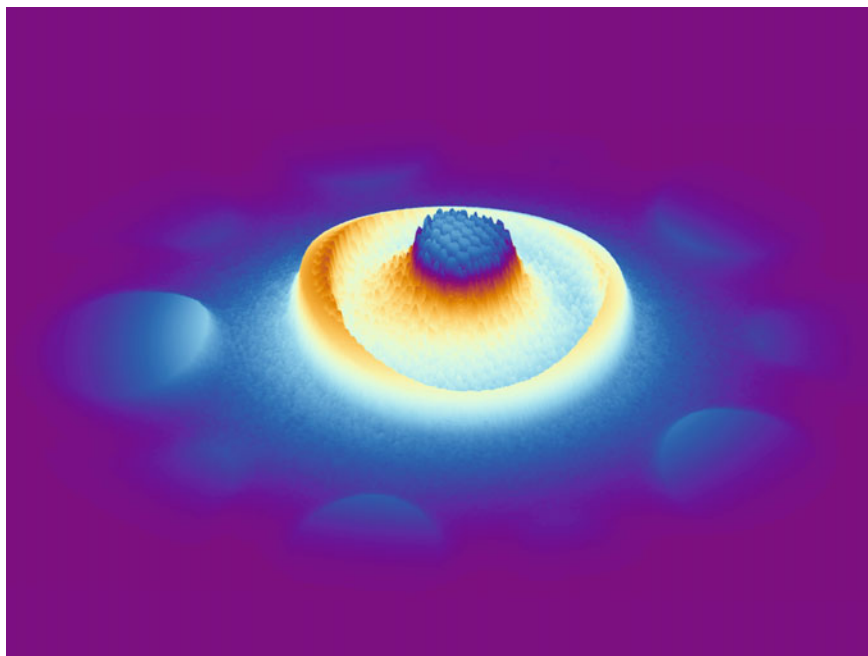


Chapter 9

Applications in X-ray Testing



Abstract In this chapter, relevant applications on X-ray testing are described. We cover X-ray testing in (i) castings, (ii) welds, (iii) baggage, (iv) natural products, and (v) others (like cargos and electronic circuits). For each application, the state of the art is presented. Approaches in each application are summarized showing how they use computer vision techniques. A detailed approach is shown in each application and some examples using Python are given in order to illustrate the performance of the methods.



Cover Image: 3D representation of the X-ray image of a wheel (X-ray image C0023_0001 colored with 'sinmap' colormap).

9.1 Introduction

In this chapter, we review some relevant applications in X-ray testing such as (i) castings, (ii) welds, (iii) baggage, (iv) natural products, and (v) others (like cargos and electronic circuits). For the first four application applications, in which the authors have been undertaking research over the last decades, we will present a description, the state of the art, a detailed approach and an example in Python. For the last application, different techniques are mentioned.

9.2 Castings

Light-alloy castings produced for the automotive industry, such as wheel rims, steering knuckles, and steering gear boxes are considered important components for overall roadworthiness. Non-homogeneous regions can be formed within the work piece in the production process. These are manifested, for example, by bubble-shaped voids, fractures, inclusions, or slag formation. To ensure the safety of construction, it is necessary to check every part thoroughly using X-ray testing. In casting inspection, automated X-ray systems have not only raised quality, through repeated objective inspections and improved processes, but have also increased productivity and consistency by reducing labor costs. Some examples are illustrated in Fig. 9.1.

9.2.1 State of the Art

Different methods for the automated detection of casting discontinuities using computer vision have been described in the literature over more than thirty years [22, 42]. In the past, the published approaches to detecting were divided into three groups [100]:

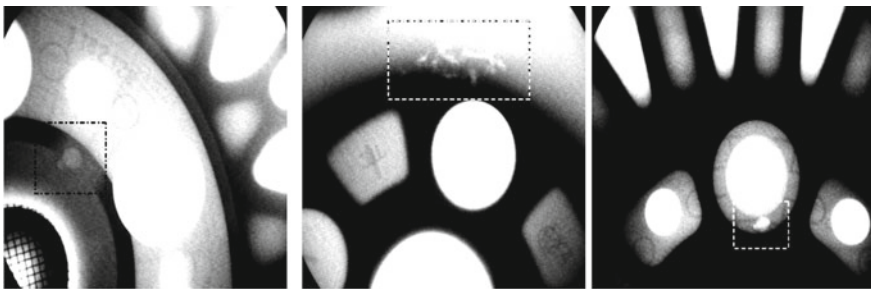


Fig. 9.1 Real defects in X-ray images of wheels

- Reference methods: In reference methods, it is necessary to take still images at selected programmed inspection positions. A test image is then compared with the reference image. If a significant difference is identified, the test piece is classified as defective.
- Methods without apriori knowledge of the structure: These approaches using pattern recognition, expert systems, artificial neural networks, general filters or multiple-views analyzes to make them independent of the position and structure of the test piece.
- Computed tomography: These approaches use computed tomography to make a reconstruction of the cast piece and thereby detect discontinuities.

Nowadays, computed tomography and multiple views for the inspection of castings are rarely used. It is clear that the methods that achieve the best performance are based on deep learning using single views. Deep learning has been successfully used in image and video recognition (see, for example, [20, 77, 155]), and it has been established as the state of the art in many areas of computer vision. The key idea of deep learning, as we show in Chap. 7, is to replace *handcrafted* features with features that are *learned* efficiently using a hierarchical feature extraction approach.

Selected approaches are summarized in Table 9.1. In this table, we follow the 3X-strategy outlined in Sect. 1.8, in which we distinguish (i) the X-ray energy used to generate the X-ray images (monochromatic, dual-, or multi-energy), (ii) the number of views used by the algorithms (single-view, multi-views, or computed tomography) and complexity of the algorithms (simple, medium, and complex—here, deep learning methods—). In this area, the automated systems are very effective, because the inspection task is fast and obtains a high performance.

9.2.2 An Application

In this section, we present a method for the automated detection of flaws based on *tracking principle* in an X-ray image sequence, i.e., first, it identifies potential defects in each image of the sequence, and second, it matches and tracks these from image to image. The key idea is to consider as false alarms those potential defects which cannot be tracked in the sequence [107]. The method for automated flaw detection presented here has basically two steps (see Fig. 9.2): *identification* and *tracking of potential flaws*. These will be described in this section.

Identification of Potential Flaws

A digital X-ray image sequence of the object test is acquired (see, for example, series C0001 of GD \times ray+). In order to ensure the tracking of flaws in the X-ray images, similar projections of the specimen must be achieved along the sequence. For this reason, the sequence consists of X-ray images taken by the rotation of the casting at small intervals (e.g., 5⁰). Since many images are captured, the time of the data acquisition is reduced by taking the images without frame averaging. The position of

Table 9.1 State of art in inspection of castings

Authors	Year	Ref	\mathbb{X}_1^* energies	\mathbb{X}_2^* views	\mathbb{X}_3^* algorithms
			1 2 3	1 2 3	1 2 3
Bandara et al.	2020	[13]	☒ ☒ □	□ □ ☒	☒ □ □
Carrasco and Mery	2011	[24]	☒ □ □	☒ ☒ □	☒ ☒ □
Cogranne and Reira	2014	[27]	☒ □ □	☒ □ □	☒ ☒ □
Du et al.	2019	[34]	☒ □ □	☒ □ □	□ □ ☒
Ferguson et al.	2017	[40]	☒ □ □	☒ □ □	□ □ ☒
Ferguson et al.	2017	[41]	☒ □ □	☒ □ □	□ □ ☒
Jin et al.	2020	[66]	☒ □ □	☒ □ □	☒ ☒ □
Kamalakannan and Rajamanickam	2017	[68]	☒ □ □	☒ □ □	☒ ☒ □
Li et al.	2006	[81]	☒ □ □	☒ □ □	☒ ☒ □
Li et al.	2015	[80]	☒ □ □	☒ □ □	☒ ☒ □
Li et al.	2019	[79]	☒ □ □	□ □ ☒	☒ □ □
Lin et al.	2018	[86]	☒ □ □	☒ □ □	□ □ □
Mery and Filbert	2002	[107]	☒ □ □	☒ ☒ □	☒ ☒ □
Mery et al.	2013	[114]	☒ □ □	☒ ☒ □	☒ ☒ □
Mery	2015	[103]	☒ □ □	☒ ☒ □	☒ ☒ □
Mery and Arteta	2017	[105]	☒ □ □	☒ □ □	☒ ☒ ☒
Mery	2020	[104]	☒ □ □	☒ □ □	□ □ ☒
Pieringer and Mery	2010	[136]	☒ □ □	☒ ☒ □	☒ ☒ □
Pizarro et al.	2008	[137]	☒ □ □	☒ ☒ □	☒ ☒ □
Ramirez and Allende	2013	[138]	☒ □ □	☒ □ □	☒ ☒ □
Ren et al.	2019	[139]	☒ □ □	☒ □ □	□ □ ☒
Tang et al.	2019	[163]	☒ □ □	☒ □ □	□ □ ☒
Tang et al.	2009	[162]	☒ □ □	☒ □ □	☒ ☒ □
Yahaghi et al.	2020	[179]	☒ □ □	☒ □ □	☒ ☒ □
Yong et al.	2016	[182]	☒ □ □	☒ □ □	☒ ☒ □
Zhao et al.	2014	[189]	☒ □ □	☒ □ □	☒ ☒ □
Zhao et al.	2015	[190]	☒ □ □	☒ □ □	☒ ☒ □
Zhang et al.	2018	[187]	☒ □ □	☒ □ □	☒ ☒ □
		*1	Mono	Mono	Simple
		2	Dual	Multi	Medium
		3	Multi	CT	Complex

□ not used, ☒ used

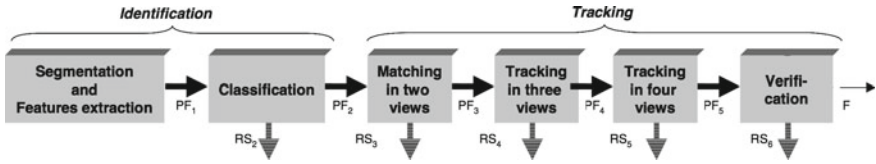


Fig. 9.2 Automated flaw detection in aluminum castings based on the tracking of potential defects in an X-ray image sequence: PF = potential flaws, RS = potential flaws classified as regular structures, F = detected flaws [107]

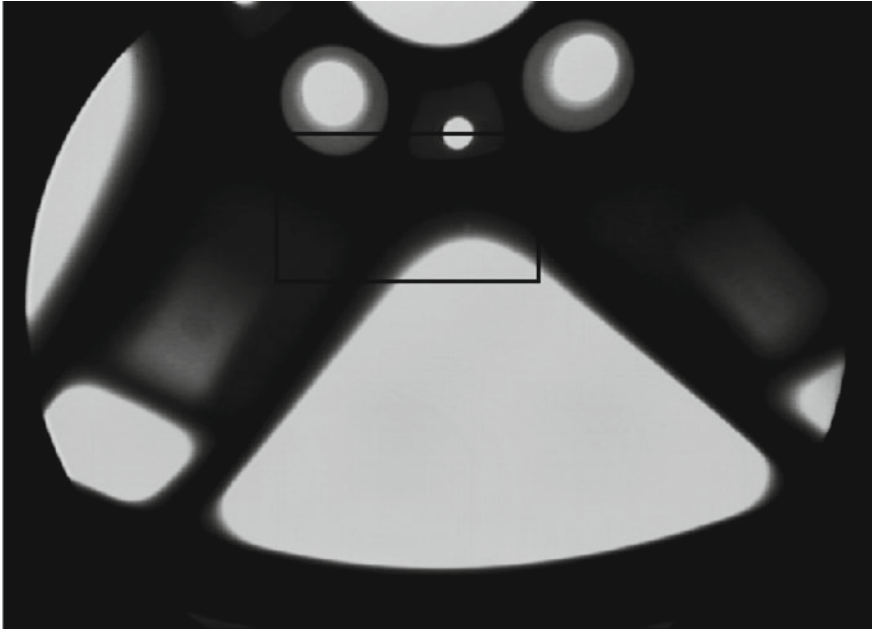


Fig. 9.3 X-ray image C0001_0030 of an aluminum wheel (see zoom in Fig. 9.4)

the casting, provided on-line by the manipulator is registered at each X-ray image to calculate the perspective projection matrix \mathbf{P}_p (for details see Sect. 3.3.4 and Example 3.5). An X-ray image sequence is shown in Fig. 9.5.

The detection of potential flaws identifies regions in X-ray images that may correspond to real defects. This process takes place in each X-ray image of the sequence without considering information about the correspondence between them. Two general characteristics of the defects are used for identification purposes: (i) a flaw can be considered as a connected subset of the image, and (ii) the gray level difference between a flaw and its neighborhood is significant. However, as the signal-to-noise ratio in our X-ray images is low, the flaws signal is slightly greater than the background noise, as illustrated in Fig. 9.4. In our experiments, the mean gray level of the flaw signal (without background) was between 2.4 and 28.8 gray values with a

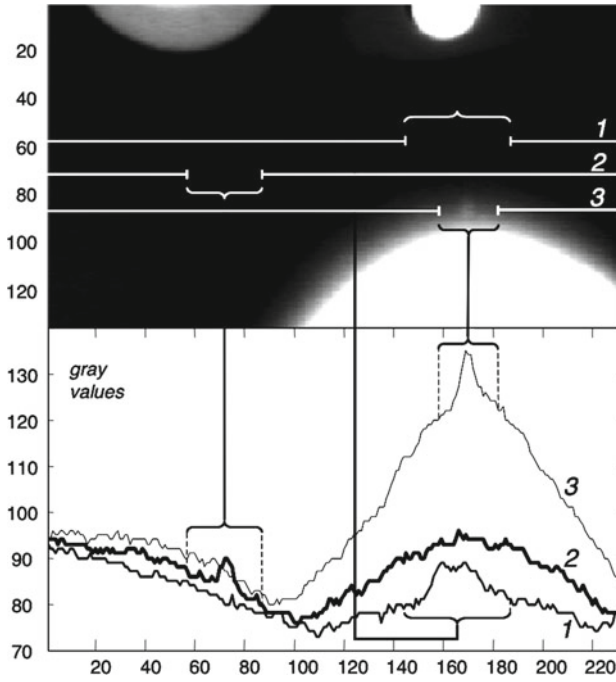


Fig. 9.4 Zoom of Fig. 9.3 and gray level profile along three rows crossing defects

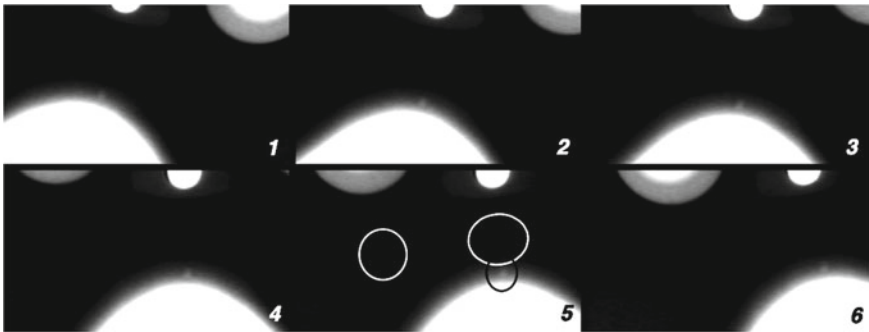


Fig. 9.5 X-ray image sequence with three flaws (image 5 is shown in Fig. 9.4)

standard deviation of 6.1. Analyzing a homogeneous background in different areas of interest of normal parts, we found that the noise signal was within ± 13 gray values with a standard deviation of 2.5. For this reason, the identification of real defects with poor contrast can also involve the detection of false alarms.

According to the mentioned characteristics of the real flaws, our method of identification has the following two steps (see Fig. 9.6):

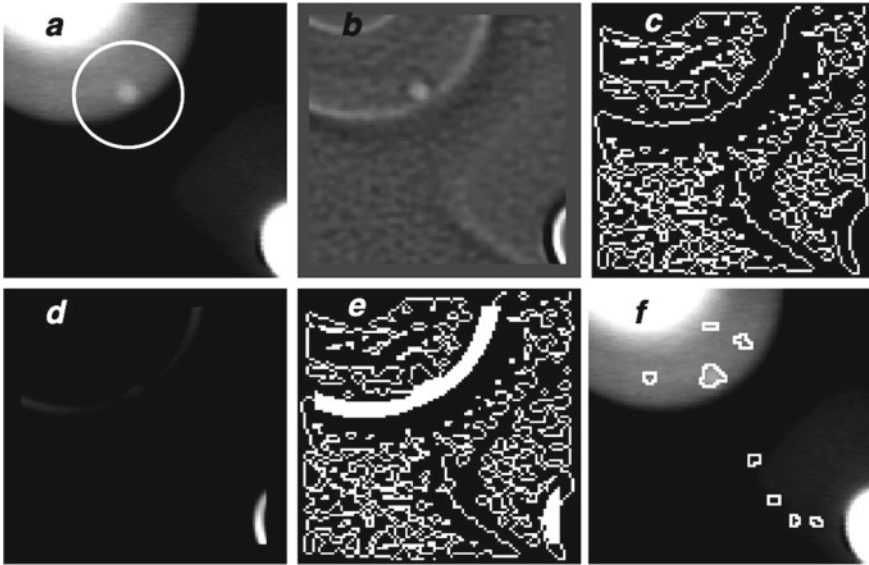


Fig. 9.6 Identification of potential flaws: **a** X-ray image with a small flaw at an edge of a regular structure, **b** Laplacian-filtered image with $\sigma = 1.25$ pixels (kernel size = 11×11), **c** zero-crossing image, **d** gradient image, **e** edge detection after adding high gradient pixels, and **f** potential flaws

Edge Detection: A Laplacian-of-Gaussian (LoG) kernel and a zero-crossing algorithm [37] are used to detect the edges of the X-ray images. The LoG-operator involves a Gaussian low-pass filter which is a good choice for the pre-smoothing of our noisy images. The resulting binary edge image should produce at real flaws closed and connected contours which demarcate *regions*. However, a flaw may not be perfectly enclosed if it is located at an edge of a regular structure as shown in Fig. 9.6c. In order to complete the remaining edges of these flaws, a thickening of the edges of the regular structure is performed as follows: (a) the gradient image¹ of the original image is computed (see Fig. 9.6d); (b) by thresholding the gradient image at a high gray level a new binary image is obtained; and (c) the resulting image is added to the zero-crossing image (see Fig. 9.6e).

Segmentation and Classification of Potential Flaws: Afterwards, each closed region is segmented and classified as a potential flaw if (a) its mean gray level is 2.5% greater than the mean gray level of its surroundings (to ensure the detection of the flaws with a poor contrast); and (b) its area is greater than 15 pixels (very small flaws are permitted). A statistical study of the classification of potential flaws using more than 70 features can be found in [108].

¹The gradient image is computed by taking the square root of the sum of the squares of the gradient in a horizontal and vertical direction. These are calculated by the convolution of the X-ray image with the first derivative (in the corresponding direction) of the Gaussian low-pass filter used in the LoG-filter.

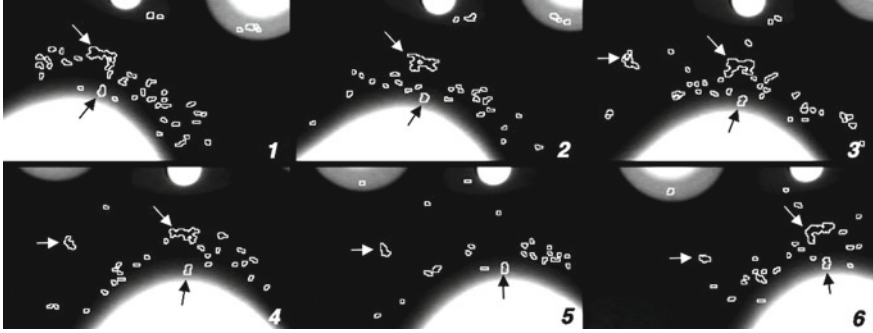


Fig. 9.7 Identification of potential flaws (the arrows indicate real flaws)

This is a very simple detector of potential flaws (see implementation in Example 5.6). However, the advantages are as follows: (a) it is a single detector (it is the same detector for each image), and (b) it is able to identify potential defects independent of the placement and the structure of the specimen.

Using this method, some real defects cannot be identified in all X-ray images in which they appear if the contrast is very poor or the flaw is not enclosed by edges. For example, in Fig. 9.7 one can observe that the biggest real flaw was identified in images 1, 2, 3, 4, and 6, but not in image 5 where only two of the three real flaws were identified (compare with Fig. 9.4). Additionally, if a flaw is overlapped by edges of the structure of the casting, not all edges of the flaw can be detected. In this case, the flaw will not be enclosed and therefore not be segmented. Furthermore, a small flaw that moves in front (or behind) a thick cross section of the casting, in which the X-rays are highly absorbed, may cause an occlusion. In our experiments, this detector identified the real flaws in four or more (not necessarily consecutive) images of the sequence.

Multiple-View Detection

In the previous step, n_1 potential regions were segmented and described in the entire image sequence \mathbb{I} . Each segmented region is labeled with a unique number $r \in \mathbf{T}_1 = \{1, \dots, n_1\}$. In view i , there are m_i segmented regions that are arranged in a subset $\mathbf{t}_i = \{r_{i,1}, r_{i,2}, \dots, r_{i,m_i}\}$, i.e., $\mathbf{T}_1 = \mathbf{t}_1 \cup \mathbf{t}_2 \cup \dots \cup \mathbf{t}_m$.

The matching and tracking algorithms combine all regions to generate consistent tracks of the object's parts of interest across the image sequence. The algorithm has the following steps:

Matching in Two Views: All regions in view i that have corresponding regions in the next p views are searched, i.e., regions $r_1 \in \mathbf{t}_i$ that have corresponding regions $r_2 \in \mathbf{t}_j$ for $i = 1, \dots, m - 1$ and $j = i + 1, \dots, \min(i + p, m)$. In our experiments, we use $p = 3$ to reduce the computational cost. The matched regions (r_1, r_2) are those that meet *similarity* and *location* constraints. The similarity constraint means that corresponding descriptors \mathbf{y}_{r_1} and \mathbf{y}_{r_2} must be similar enough such that

$$\|\mathbf{y}_{r_1} - \mathbf{y}_{r_2}\| < \varepsilon_1. \quad (9.1)$$

The location constraint means that the corresponding locations of the regions must meet the epipolar constraint. In this case, the Sampson distance between \mathbf{x}_{r_1} and \mathbf{x}_{r_2} is used, i.e., the first-order geometric error of the epipolar constraint must be small enough such that:

$$|\mathbf{x}_{r_2}^\top \mathbf{F}_{ij} \mathbf{x}_{r_1}| \left(\frac{1}{\sqrt{a_1^2 + a_2^2}} + \frac{1}{\sqrt{b_1^2 + b_2^2}} \right) < \varepsilon_2, \quad (9.2)$$

with $\mathbf{F}_{ij} \mathbf{x}_{r_1} = [a_1 \ a_2 \ a_3]^\top$ and $\mathbf{F}_{ij}^\top \mathbf{x}_{r_2} = [b_1 \ b_2 \ b_3]^\top$. In this case, \mathbf{F}_{ij} is the fundamental matrix between views i and j calculated from projection matrices \mathbf{P}_i and \mathbf{P}_j [56] (see Sect. 3.5.1). In addition, the location constraint used is as follows:

$$\|\mathbf{x}_{r_1} - \mathbf{x}_{r_2}\| < \rho(j - i), \quad (9.3)$$

because the translation of corresponding points in these sequences is smaller than ρ pixels in consecutive frames.

If we have 3D information about the space where our test object should be, it is worth to evaluating whether the 3D point reconstructed from the centers of mass of the regions must belong to the space occupied by the casting. From \mathbf{m}_p^a and \mathbf{m}_q^b the corresponding 3D point $\hat{\mathbf{M}}$ is estimated using the linear approach of Hartley in [56]. For two views this approach is faster than the least squares technique. It is necessary to examine if $\hat{\mathbf{M}}$ resides in the volume of the casting, the dimensions of which are usually known a priori (e.g., a wheel is assumed to be a cylinder)².

Finally, a new matrix \mathbf{T}_2 sized $n_2 \times 2$ is obtained with all matched duplets (r_1, r_2) , one per row. If a region is found to have no matches, it is eliminated. Multiple matching, i.e., a region that is matched with more than one region, is allowed. Using this method, problems like non-segmented regions or occluded regions in the sequence can be solved by tracking if a region is not segmented in consecutive views.

Matching in 3 Views: Based on the matched regions stored in matrix \mathbf{T}_2 , we look for triplets (r_1, r_2, r_3) , with $r_1 \in \mathbf{t}_i, r_2 \in \mathbf{t}_j, r_3 \in \mathbf{t}_k$ for views i, j , and k . We know that a row a in matrix \mathbf{T}_2 has a matched duplet $[T_2(a, 1) \ T_2(a, 2)] = [r_1 \ r_2]$. We then look for rows b in \mathbf{T}_2 in which the first element is equal to r_2 , i.e., $[T_2(b, 1) \ T_2(b, 2)] = [r_2 \ r_3]$. Thus, a matched triplet (r_1, r_2, r_3) is found if the regions r_1, r_2 , and r_3 meet the trifocal constrain:

$$\|\hat{\mathbf{x}}_{r_3} - \mathbf{x}_{r_3}\| < \varepsilon_3, \quad (9.4)$$

²It is possible to use a CAD model of the casting to evaluate this criterion more precisely. Using this model we could discriminate a small hole of the regular structure that is identified as a potential flaw. Additionally, the CAD model can be used to inspect the casting geometry, as shown in [129].

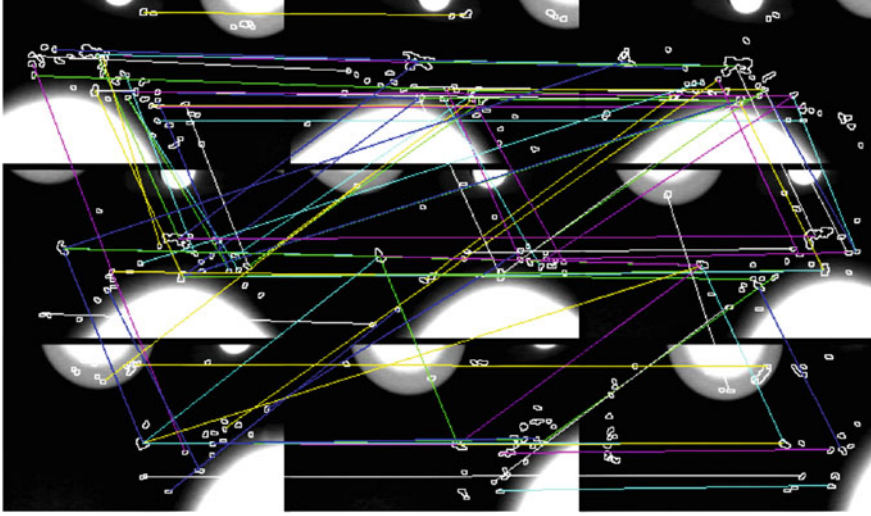


Fig. 9.8 Matching of potential flaws in two views

This means that \mathbf{x}_{r_3} must be similar enough to the re-projected point $\hat{\mathbf{x}}_{r_3}$, computed from the points in views i and j (\mathbf{x}_{r_1} and \mathbf{x}_{r_2}), and the trifocal tensors T_i^{jk} of views i, j, k calculated from projection matrices $\mathbf{P}_i, \mathbf{P}_j$, and \mathbf{P}_k [56] (see (3.76)). A new matrix \mathbf{T}_3 sized $n_3 \times 3$ is built with all matched triplets (r_1, r_2, r_3) , one per row. Regions in which the three views do not match are eliminated.

The results of our example are shown in Fig. 9.8.

Matching in More Views: For $v = 4, \dots, q \leq m$ views, we can build the matrix recursively \mathbf{T}_v , sized $n_v \times v$, with all possible v -tuplets (r_1, r_2, \dots, r_v) that fulfill $[T_{v-1}(a, 1) \dots T_{v-1}(a, v-1)] = [r_1 r_2 \dots r_{v-1}]$ and $[T_{v-1}(b, 1) \dots T_{v-1}(b, v-1)] = [r_2 \dots r_{l-1} r_v]$, for $j, k = 1, \dots, n_{v-1}$. No more geometric constraints are required because it is redundant. The final result is stored in matrix \mathbf{T}_q . For example, for $q = 4$ we store in matrix \mathbf{T}_4 the matched quadruplets (r_1, r_2, r_3, r_4) with $r_1 \in \mathbf{t}_i, r_2 \in \mathbf{t}_j, r_3 \in \mathbf{t}_k, r_4 \in \mathbf{t}_l$ for views i, j, k and l .

Figure 9.10 shows the tracked regions of our example that fulfill this criterion. Only two false trajectories are observed (see arrows).

As our detector cannot guarantee the identification of all real flaws in more than four views, a tracking in five views could lead to the elimination of those real flaws that were identified in only four views. However, if a potential flaw is identified in more than four views, more than one quadruplet can be detected. For this reason, these corresponding quadruplets are joined in a trajectory that contains more than four potential flaws (see trajectory with arrows in Fig. 9.10).

The matching condition for building matrix $\mathbf{T}_i, i = 3, \dots, q$, is efficiently evaluated (avoiding an exhaustive search) by using a k -d tree structure [21] to search

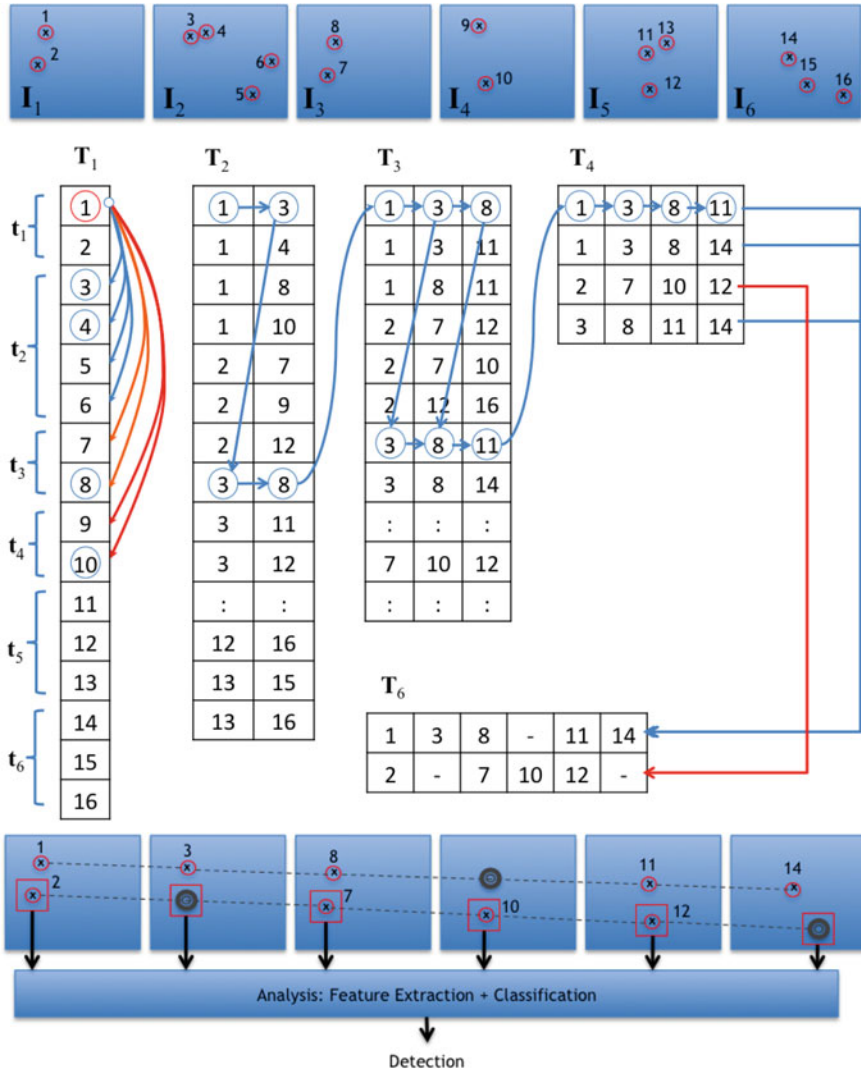


Fig. 9.9 Tracking example with $m = 6$ views. In each view there are 2, 4, 2, 2, 3, and 3 segmented regions, i.e., there are $n_1 = 16$ regions in total. For each region we seek corresponding regions in the next $p = 3$ views (see matching arrows in T_1 : region 1 with regions (3, 4, 5, 6) in view 2, regions (7, 8) in view 3, and (9, 10) in view 4). We observe that after tracking in 2, 3, and 4 views there are only two tracks in T_6 that could be tracked in 5 and 4 views respectively. The regions that were not segmented can be recovered by reprojection (see gray circles in views 2, 4, and 6). Finally, each set of tracked regions are analyzed in order to take the final decision

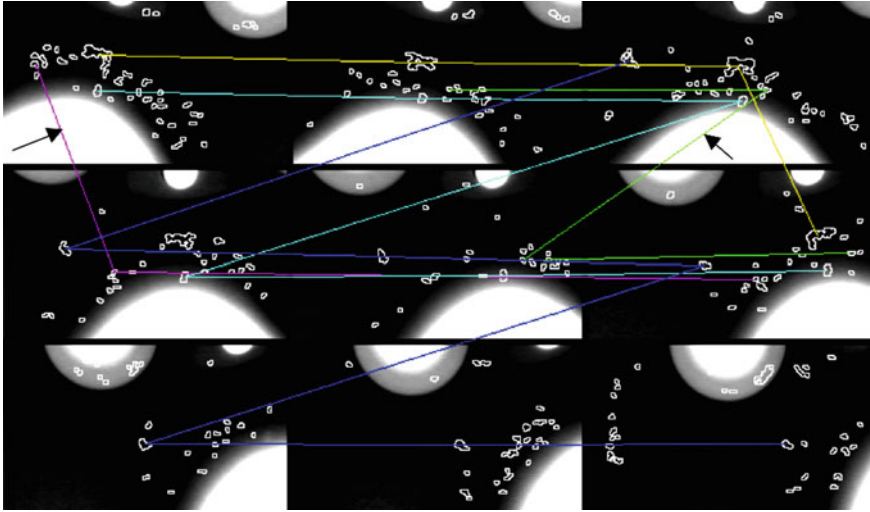


Fig. 9.10 Tracking in more views (the arrows indicate false detections)

the nearest neighbors for zero Euclidean distance between the first and last $i - 2$ columns in \mathbf{T}_{i-1} .

Merging Tracks: Matrix \mathbf{T}_q defines tracks of regions in q views. It can be observed that some of these tracks correspond to the same region. For this reason, it is possible to merge tracks that have $q - 1$ common elements. In addition, if a new track has more than one region per view, we can select the region that shows the minimal reprojection error after computing the corresponding 3D location. In this case, a 3D reconstruction of $\hat{\mathbf{X}}$ is estimated from tracked points [56]. Finally, matrix \mathbf{T}_m is obtained with all merged tracks in the m views. See an example of the whole tracking algorithm in Fig. 9.9.

Analysis: The 3D reconstructed point $\hat{\mathbf{X}}$ from each set of tracked points of \mathbf{T}_m can be reprojected in views where the segmentation may have failed to obtain the complete track in all views. The reprojected points of $\hat{\mathbf{X}}$ should correspond to the centroids of the non-segmented regions. It is then possible to calculate the size of the projected region as an average of the sizes of the identified regions in the track. In each view, a small window centered in the computed centroids is defined. These corresponding small windows, referred to as *tracked part*, will be denoted as $\mathbb{W} = \{\mathbf{W}_1, \dots, \mathbf{W}_m\}$. In each view a small window is defined with the estimated size in the computed centers of gravities (see Fig. 9.11). Afterwards, the corresponding windows are averaged. Thus, the attempt is made to increase the signal-to-noise ratio by the factor \sqrt{n} , where n is the number of averaged windows. As flaws must appear as contrasted zones relating to their environment, we can verify if the contrast of each averaged window is greater than 2.5%. With this verification it is possible to eliminate all remaining false detections. Figure 9.11 shows the detection in our sequence using

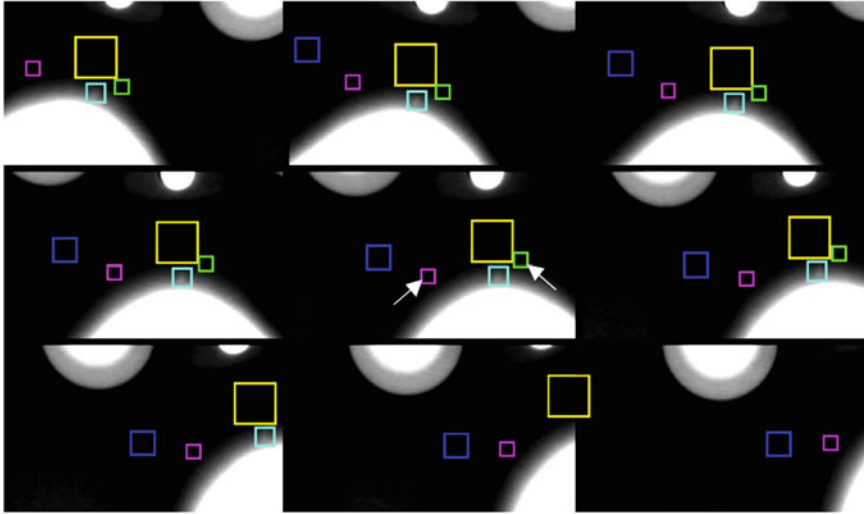


Fig. 9.11 Reconstruction and verification: the false detections (indicated by the arrows) are eliminated after the verification in all images of the sequence

this method. Our objective is then achieved: the real defects were separated from the false ones.

Experimental Results

In this section, results of automatic inspection of cast aluminum wheels using the outlined approach are presented. These results have been achieved recently on synthetic flaws and real data. The parameters of our method have been manually tuned, giving $\sigma = 1.25$ pixels (for LoG-operator), $\varepsilon_2 = 0.75$ mm, $\varepsilon_s = 0.7$, and $\varepsilon_3 = 0.9$ mm. These parameters were not changed during these experiments. A wheel was considered to be a cylinder with the following dimensions: 470 mm diameter and 200 mm height. The focal length (distance between X-ray source and entrance screen of the image intensifier) was 884 mm. The bottom of a wheel was 510 mm from the X-ray source. Thus, a pattern of 1 mm in the middle of the wheel is projected in the X-ray projection coordinate system as a pattern of 1.73 mm, and in the image coordinate system as a pattern of 2.96 pixels. The sequences of X-ray images were taken by rotation of the casting at 5° .

The detection performance will be evaluated by computing the number of True Positives (TP) and False Positives (FP). They are respectively defined as the number of flaws that are correctly classified and the number of misclassified regular structures. The TP and FP will be normalized by the number of existing flaws (E) and the number of identified potential flaws (I). Thus, we define the following percentages: $TPP = TP / E \times 100$ and $FPP = FP / I \times 100$. Ideally, $TPP = 100\%$ and $FPP = 0\%$.

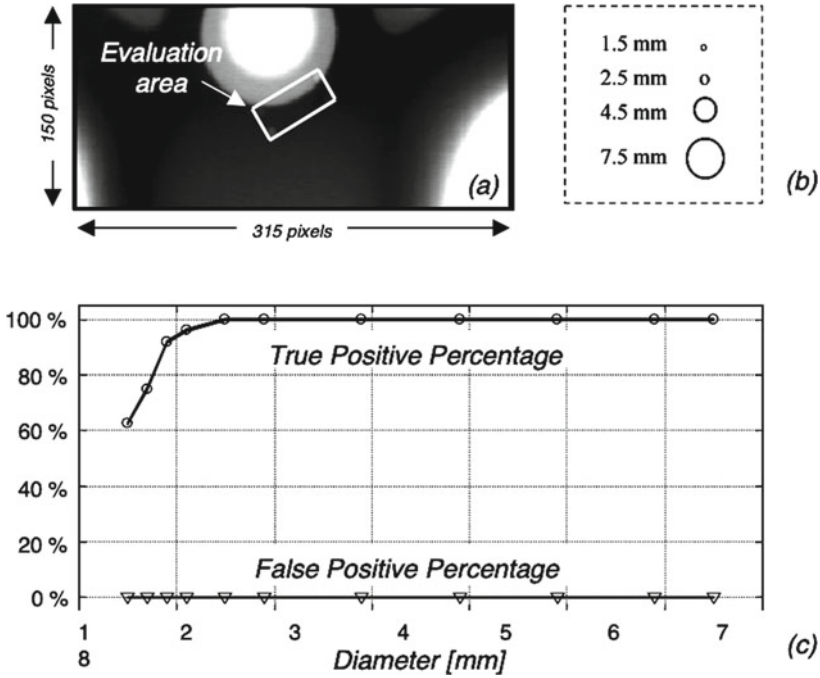


Fig. 9.12 Detection on synthetic flaws: **a** X-ray image and evaluated area, **b** flaw sizes, and **c** TPP and FPP

Synthetic Flaws: To evaluate the performance of our method in critical cases, real data in which synthetic flaws have been added were examined (see Sect. 8.4.3). A simple 3D modeled flaw (a spherical bubble) was projected and superimposed on real X-ray images of an aluminum wheel according to the law of X-ray absorption [98]. In our experiment, a flaw is simulated in 10 X-ray images of a real casting, in an area that included an edge of the structure (see Fig. 9.12a). In this area the synthetic flaw was located in 24 different positions in a regular grid manner. At each position TPP and FPP were tabulated. This test was repeated for different sizes of the flaws ($\varnothing = 1.5 \sim 7.5$ mm) which are illustrated in Fig. 9.12b. The results are shown in Fig. 9.12c. It was observed that the FPP was always zero. The TPP was 100% for $\varnothing \geq 2.5$ mm, and greater than 95% for $\varnothing \geq 2.1$ mm. However, the identification of the flaw may fail (and therefore also its detection) if it is very small and is located at the edge of the structure of the casting. In this case one may choose a smaller value of the parameter σ in the LoG operator of the edge detection, which will unfortunately increment the FPP. Other non-critical experiments, where the area of the simulation does not include an edge of the structure, have led to perfect results (TPP = 100%, FPP = 0%) for $\varnothing \geq 1.5$ mm (≥ 4.4 pixels). Usually, the minimum detectable defect size according to inspection specifications is in the order of $\varnothing = 2$ mm. In X-ray

Table 9.2 Detection of flaws on real data

Seq.	X-ray	Flaws in the	Flaws in the	Identification			Detection	
	Images	Sequence	Images (E)	TP	FP	Total (I)	TP	FP
1	10	2	12	12	249	261	2	0
2	9	1	9	8	238	246	1	0
3	9	3	23	19	253	272	3	0
4	8	1	8	4	413	417	1	0
5	6	1	6	6	554	560	1	0
6	8	1	8	8	196	204	1	0
7	6	3	18	14	445	459	3	0
8	6	0	0	0	178	178	0	0
9	9	0	0	0	256	256	0	0
10	8	0	0	0	150	150	0	0
11	8	0	0	0	345	345	0	0
12	6	0	0	0	355	355	0	0
13	6	0	0	0	365	365	0	0
14	9	0	0	0	313	313	0	0
Total	108	12	84	71	4310	4381	12	0
Percentage				85%	98%		100%	0%

testing, smaller flaws can be detected by decreasing the distance of the object test to the X-ray source.

Real Data: Fourteen X-ray image sequences of aluminum wheels with twelve known flaws were inspected. Three of these defects were existing blow holes (with $\varnothing = 2.0 \sim 7.5$ mm). They were initially detected by a visual (human) inspection. The remaining nine flaws were produced by drilling small holes ($\varnothing = 2.0 \sim 4.0$ mm) in positions of the casting which were known to be difficult to detect. Casting flaws are present only in the first seven sequences. The results are summarized in Table 9.2, Figs. 9.13, and 9.14. In the identification of potential flaws, it was observed that the FPP was 98% (4310/4381). Nevertheless, the TPP in this experiment was good, and it was possible to identify 85% (71/84) of all projected flaws in the sequences (13 of the existing 84 flaws were not identified because the contrast was poor or they were located at edges of regular structures). It was observed that in the next steps, the FPP was reduced to nil. The detection of the real flaws was successful in all cases. The first six images of sequence 3 and its results were already illustrated in Figs. 9.5, 9.7, 9.8, 9.9, 9.10 and 9.11. The results on the other sequences with flaws are shown in Fig. 9.13.

Comparison with Other Methods: In this section, we present a comparison of our proposed algorithm with other methods that can be used to detect defects in aluminum

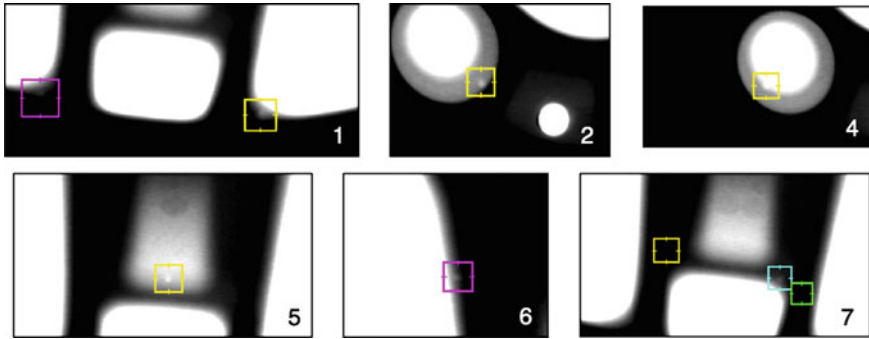


Fig. 9.13 Detected flaws in sequences 1, 2, 4, 5, 6, and 7 (sequence 3 is shown in Fig. 9.11)

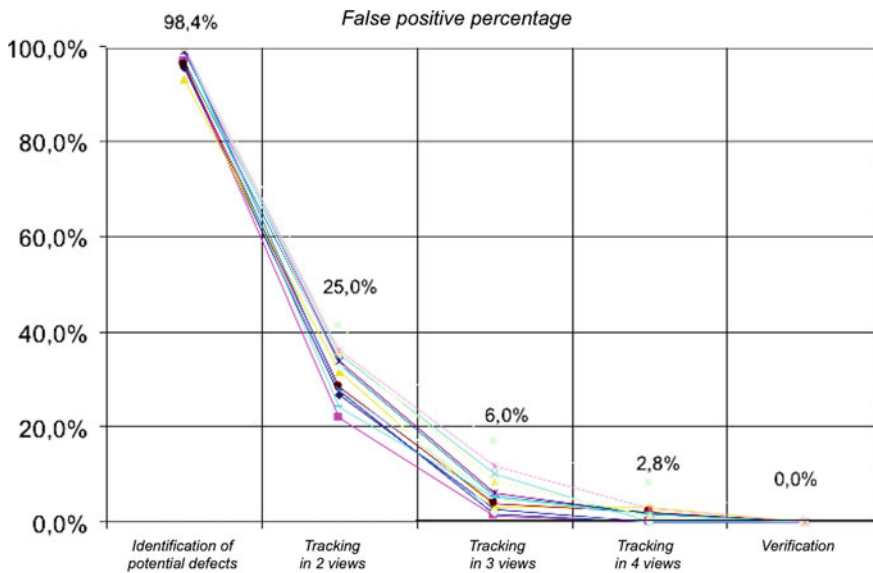


Fig. 9.14 False positive percentage on real data in the fourteen real sequences (the number of identified potential flaws corresponds to 100%). The mean of each step is given over the fourteen curves

castings. In this comparison, we evaluate the same real fourteen sequences used in the previous section. The results are summarized in Table 9.3.

Firstly, we compared the first step of our method (*identification of potential flaws*). The objective of this step is the use of a single filter, instead of a set of filters adapted to the regular structure of the specimen. We evaluated the well-known Canny filter (see, for example, [37]). As this filter detects sparse edge pixels that not necessarily produce at real flaws closed and connected contours, the TPP of this detector was unacceptable, only 4% of the real flaws were identified ('Canny I' in Table 9.3). In

Table 9.3 Comparison with other methods

Method	Identification		Detection	
	TPP	FPP	TPP	FPP
Canny I	4%	97%	0%	–
Canny II	40%	99%	17%	40%
Median I	55%	85%	33%	36%
Median II	88%	98%	92%	45%
Tracking in 3	85%	98%	100%	25%
Tracking in 5	85%	98%	83%	0%
PXV-5000	–	–	100%	0%
Proposed	85%	98%	100%	0%

order to increase the number of closed regions a dilation of the edges using a 3×3 mask was performed. Although the TPP is improved to 40% (‘Canny II’ in Table 9.3), many flaws were not detected in any of the images of the sequence. For this reason, only 17% of the real flaws were detected after the tracking and verification.

Another detection of potential flaws can be performed using a region-based segmentation. Median filtering is normally used to generate an error-free image, since defect structures are essentially eliminated, while design features of the test piece are normally preserved [109]. Once the error-free reference image is computed, an error difference image between original and error-free images is calculated. Casting defects are then identified when a sufficiently large gray level in the error difference image occurs. The best results were obtained using a median filter with a 11×11 mask. We evaluated two thresholds: $\theta = 6$ and $\theta = 2$ —by 256 gray levels—(see ‘Median I’ and ‘Median II’ in Table 9.3). In the first case the TPP was only 55%. By decreasing the threshold value we increased the TPP to 88%, that is slightly better than our detector (85%). However, systematic false alarms were detected at the corners of the regular structures. Since these false alarms satisfy the multifocal conditions, they can be tracked in the sequence. For this reason, this detector can only be used if the median filter is adapted to the regular structures of the specimen using a priori information. Normally, a set of median filters is used for each X-ray image [42, 58, 59].

In order to evaluate the second step of our method (*tracking of potential flaws*), we tested the method by tracking the potential flaws in 3 and in 5 views, instead of 4 views (see ‘Tracking in 3’, ‘Tracking in 5’, and ‘Proposed’ in Table 9.3). By considering only three views we obtained so many false alarms that the verification step detected 4 false alarms (25%). In the other case, by tracking the potential flaws in five views, real flaws that were segmented in only four views of the sequences were not tracked. For this reason, only 83% of the real flaws were detected.

Finally, we inspected the test castings using a classic image processing method. In our experiments, we used the industrial software PXV-5000 [110]. The results were excellent: 100% of the real flaws were detected without false alarms. As a result of

its peak detection performance, the classic image processing methods have become the most widely established in industrial applications. However, these methods suffer from the complicated configuration of the filtering, which is tailored to the test piece. In our experiments, the configuration process has taken two weeks. Nevertheless, as our method requires only a few number of parameters, the configuration could be carried out in hours.

Conclusions

A new method for automated flaw detection in aluminum castings using multiple-view geometry has been developed. Our method is very efficient because it is based on a two-step analysis: identification and tracking. The idea was to try to imitate the way a human inspector inspects X-ray images: first relevant details (potential defects) are detected, followed by tracking them in the X-ray image sequence. In this way, the false detections can be eliminated without discriminating the real flaws.

The great advantage of our first step is the use of a single filter to identify potential defects, which is independent of the structure of the specimen. Nevertheless, its disadvantages are as follows: (a) the false positive percentage is enormous; (b) the true positive percentage could be poor if the flaws to be detected are very small and located at the edge of a structure; and (c) the identification of regions is time-consuming. Contrarily, the second step is highly efficient in both discrimination of false detections and tracking of real defects, and is not time-consuming, due to the use of the multiple-view tensors.

To inspect a whole wheel our method requires approximately 100 views of 256×256 pixels, that can be processed in one minute. The required computing time is acceptable for practical applications because a typical inspection process takes about one minute, independently of whether it is performed manually or automatically.

We have shown that these preliminary results are promising. However, given that the performance of the method has been verified on only a few X-ray image sequences, an evaluation on a broader data base is necessary.

It is possible to combine our second step with existing defect detection technologies, which use a priori information of the regular structures of the casting to detect flaws in single images (see, for example [110]). This method could also be used in the automated flaw detection of other objects. In the adaptation of our method, one must determine the number of views in which a flaw must be tracked. If the false positive percentage by identifying potential flaws is low (or high), one may track a flaw in fewer (or more) views of the sequence. However, one must guarantee that the real flaws will be identified as potential flaws in these views.

9.2.3 An Example

In this section, an implementation that can be used for defect detection of castings in single views is presented. It consists of features that are extracted from positive class (the defects) and negative class (the background).

An example of using detection in multiple views can be found in Sect. 9.4.3.



Python Example 9.1: In this example, we show how to implement a classifier that is able to detect casting defects in single X-ray images. For this end we use series C0002 that contains small images with and without defects. In addition, for this series we have the ground truth for all defects. The strategy of this example is the strategy that we proposed in Algorithm 1, that means we extract many features the proposed algorithm searches the combination of features and a classifier that maximizes the accuracy.

Listing 9.1 : Defect detection in castings

```

import numpy as np
import numpy as np
from pyxvis.io.data import load_features,save_features
from pyxvis.learning.evaluation import best_features_classifier
from pyxvis.features.selection import clean_norm,clean_norm_transform
from pyxvis.features.extraction import extract_features_labels

dataname = 'c32' # prefix of npy files of training and testing data
fxnew = 1 # the features are (0) loaded or (1) extracted and saved
if fxnew:
    # features to extract
    fx = ['basicint','gabor-ri','lbp-ri','haralick-2','fourier','hog','clp']
    # feature extraction in training images
    path = '../images/castings/'
    X,d = extract_features_labels(fx,path+'train','jpg')
    # feature extraction in testing images
    Xt,dt = extract_features_labels(fx,path+'test','jpg')
    # backup of extracted features
    save_features(X,d,Xt,dt,dataname)
else:
    X,d,Xt,dt = load_features(dataname)

X,sclean,a,b = clean_norm(X)
Xt = clean_norm_transform(Xt,sclean,a,b)
# Classifiers to evaluate
ss_cl = ['maha','bayes-kde','svm-lin','svm-rbf','qda','lda','knn3','knn7','nn']
# Number of features to select
ff = [3,5,10,12,15]
# Feature selectors to evaluate
ss_fs = ['fisher','qda','svm-lin','svm-rbf']

clbest,ssbest = best_features_classifier(ss_fs,ff,ss_cl,X,d,Xt,dt,
                                     'Accuracy in Castings')
print(' Selected Features: '+str((np.sort(sclean[ssbest])))

```

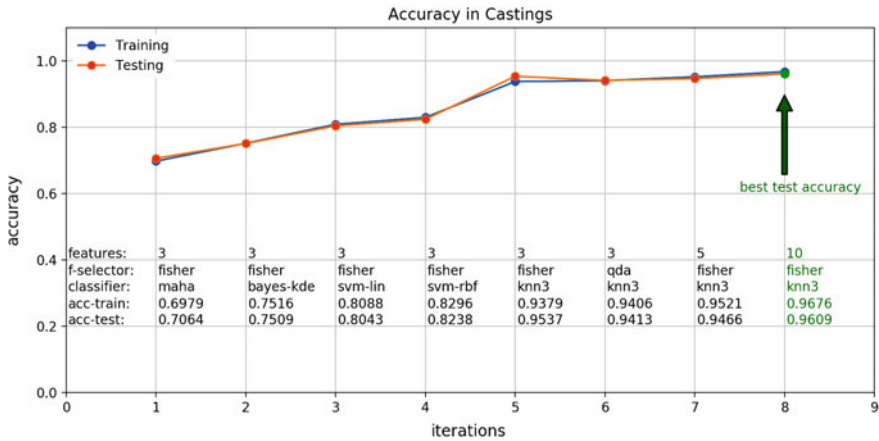


Fig. 9.15 Accuracy on training and testing dataset for castings defect detection. In this example we use the strategy proposed in Algorithm 1. [→ Example 9.1 🗺️]

The output of this code is the estimated accuracy:

```

-----
Best iteration: 8 (maximum of testing accuracy)
Feature Selector: fisher with 10 features
                  : (Fisher, )
Classifier: knn3
            : (KNeighborsClassifier, n_neighbors=3) CrossVal with 5 folds
Training-Acc: 0.9676
Testing-Acc: 0.9609
Selected Features: [ 1  2  3 16 20 24 25 26 38 72]
-----
    
```

The accuracy of the selected classifier (knn with 3 neighbors) is 96.76% with 10 features (Fig. 9.15). In this code we used `best_features_classifier` of `pyxvis` Library. The reader can use additional series of `GDxray+`, that contain annotated defects in aluminum wheels. □

9.3 Welds

In welding process, a mandatory inspection using X-ray testing is required in order to detect defects like porosity, inclusion, lack of fusion, lack of penetration, and cracks. Industrial X-ray images of welds is widely used for detecting those defects in the petroleum, chemical, nuclear, naval, aeronautics and civil construction industries, among others. An example is illustrated in Fig. 9.20.

9.3.1 *State of the Art*

Over the last 35 years, substantial research has been performed on automated detection and classification of welding defects in continuous welds using X-ray imaging [152, 153]. Typically, the approaches follow a classical computer vision schema: (i) image acquisition—an X-ray digital image is taken and stored in the computer, (ii) pre-processing—the digital image is improved in order to enhance the details, (iii) segmentation—potential welding defects are found and isolated, (iv) feature extraction/selection—significant features of the potential welding defects and their surroundings are quantified, and (v) classification—the extracted features are interpreted automatically using a priori knowledge of the welding defects in order to separate potential defects into detected welding defects or false alarms. In the last few years, some methods based on deep learning have been developed with promising results. Selected approaches are summarized in Table 9.4. In this table, we follow the 3X-strategy outlined in Sect. 1.8, in which we distinguish (i) the X-ray energy used to generate the X-ray images (monochromatic, dual-, or multi-energy), (ii) the number of views used by the algorithms (single-view, multi-views, or computed tomography) and complexity of the algorithms (simple, medium, and complex—here, deep learning methods–). As we can see there is much research on weld inspection. Achieved performance of the developed algorithms is still not high enough, thus it is not suitable for fully automated inspection.

9.3.2 *An Application*

In computer vision, many object detection and classification problems have been solved without classic segmentation using *sliding-windows*. Sliding-window approaches have established themselves as state of the art in computer vision problems where an object must be separated from the background (see, for example, successful applications in face detection [171] and human detection [30]). In sliding-window methodology, a detection window (see black square in Fig. 9.16) is slidded over an input image in both horizontal and vertical directions, and for each localization of the detection window a classifier decides to which class the corresponding portion of the image belongs to according to its features. In this section, an approach to detect defects based on sliding-windows in welds is presented [102].

Overview

We developed an X-ray computer vision approach to detect welding defects using this methodology yielding promising results. We will differentiate between the ‘detection of defects’ and the ‘classification of defects’ [82]. In the detection problem, the classes that exist are only two: ‘defects’ and ‘no-defects’, whereas the recognition of the type of the defects (e.g., porosity, slag, crack, lack of penetration, etc.) is known as classification of flaw types. This section describes our approach on detection only

Table 9.4 State of art in inspection of welds

Authors	Year	Ref	\mathbb{X}_1^* energies			\mathbb{X}_2^* views			\mathbb{X}_3^* algorithms		
			1	2	3	1	2	3	1	2	3
Ajmi et al.	2018	[4]	☒	□	□	☒	□	□	☒	☒	□
Anand et al.	2009	[11]	☒	□	□	☒	□	□	☒	☒	□
Baniukiewicz	2014	[14]	☒	□	□	☒	□	□	☒	☒	□
Gao and Yu	2014	[47]	☒	□	□	☒	□	□	☒	☒	□
Hassan et al.	2012	[57]	☒	□	□	☒	□	□	☒	☒	☒
Hou et al.	2018	[61]	☒	□	□	☒	□	□	□	□	☒
Hou et al.	2019	[62]	☒	□	□	☒	□	□	□	□	☒
Kaftandjian et al.	2003	[67]	☒	□	□	☒	□	□	☒	☒	□
Kumar et al.	2014	[74]	☒	□	□	☒	□	□	☒	☒	□
Kumar et al.	2014	[75]	☒	□	□	☒	□	□	☒	☒	□
Liao	2008	[83]	☒	□	□	☒	□	□	☒	☒	□
Liao	2009	[84]	☒	□	□	☒	□	□	☒	☒	□
Lindgren	2014	[87]	☒	□	□	☒	☒	□	☒	☒	□
Liu et al.	2017	[88]	☒	□	□	☒	□	□	□	□	☒
Mery and Berti	2003	[106]	☒	□	□	☒	□	□	☒	☒	□
Mery	2011	[102]	☒	□	□	☒	□	□	☒	☒	□
Mu et al.	2011	[120]	☒	□	□	☒	□	□	☒	☒	□
Muniategui et al.	2019	[121]	☒	□	□	☒	□	□	☒	☒	□
Muravyov and Pogadaeva	2020	[122]	☒	□	□	☒	□	□	☒	☒	□
Pan et al.	2020	[135]	☒	□	□	☒	□	□	☒	☒	☒
Shao et al.	2014	[149]	☒	□	□	☒	☒	□	☒	☒	□
Shi et al.	2007	[150]	☒	□	□	☒	□	□	☒	☒	□
da Silva et al.	2009	[154]	☒	□	□	☒	□	□	☒	☒	□
Suyama et al.	2019	[161]	☒	□	□	☒	□	□	□	□	☒
Tong et al.	2012	[164]	☒	□	□	☒	□	□	☒	☒	□
Vilar et al.	2009	[170]	☒	□	□	☒	□	□	☒	☒	□
Wang et al.	2008	[174]	☒	□	□	☒	□	□	☒	☒	□
Wang et al.	2019	[173]	☒	□	□	☒	□	□	☒	☒	□
Yiron et al.	2015	[181]	☒	□	□	☒	☒	□	☒	☒	□
Zapata et al.	2008	[185]	☒	□	□	☒	□	□	☒	☒	□
		* 1	Mono			Mono			Simple		
		2	Dual			Multi			Medium		
		3	Multi			CT			Complex		

□ not used, ☒ used

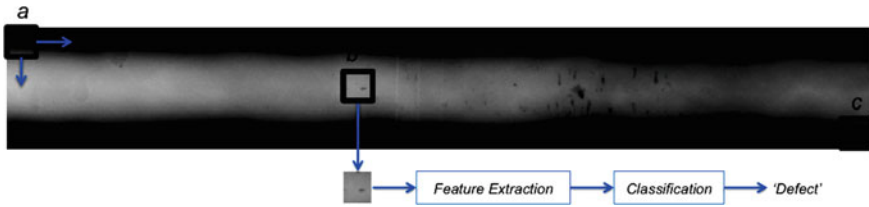


Fig. 9.16 Sliding window approach: A detection window (see black square) is slidded over the X-ray image starting at place 'a' and ending at 'c'. For each position, e.g., at 'b', features are extracted only from the sub-image defined by the square, and a classifier determines the class of this portion of the image

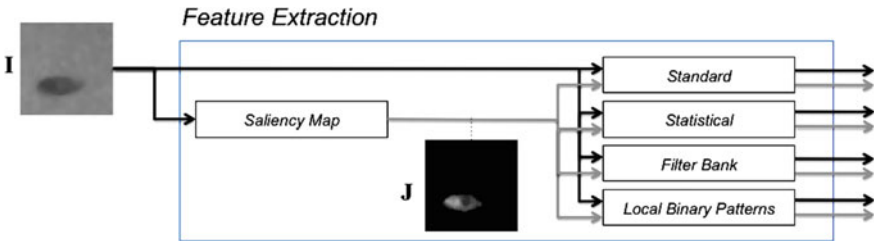


Fig. 9.17 Feature extraction: from each detection window several features are extracted (see black path). Additionally, the same features are extracted from a saliency map of the sub-window (see gray path)

and the corresponding validation experiments. The classification of defects can be developed by the reader using a similar methodology.

The key idea of this example is to use a computer vision methodology, as shown in Figs. 9.16 and 9.17, to automatically detect welding defects. In the following, feature extraction, feature selection, classification, and validation will be explained in further detail.

Feature Extraction, Selection, and Classification

Features provide information about the intensity of a sub-image. In our approach, p features per *intensity* channel were extracted. The used intensity channels in our work are only two: the grayscale X-ray image (**I**) and a saliency map (**J**) computed from **I**, i.e., $p \times 2$ features for two intensity channels. In order to reduce the computational time, we restricted the feature extraction for these only two channels, however, other channels, like Harris transform [55] or other saliency maps, can be used.

The saliency map **J** is obtained using a center-surround saliency mechanism based on a biologically inspired attention system [118]³. In order to achieve faster processing, this theory proposes that the human visual system uses only a portion of the image, called *focus of attention*, to deal with complex scenes. In our approach, we

³The saliency function is implemented in [saliency](#) of pyxvis Library.

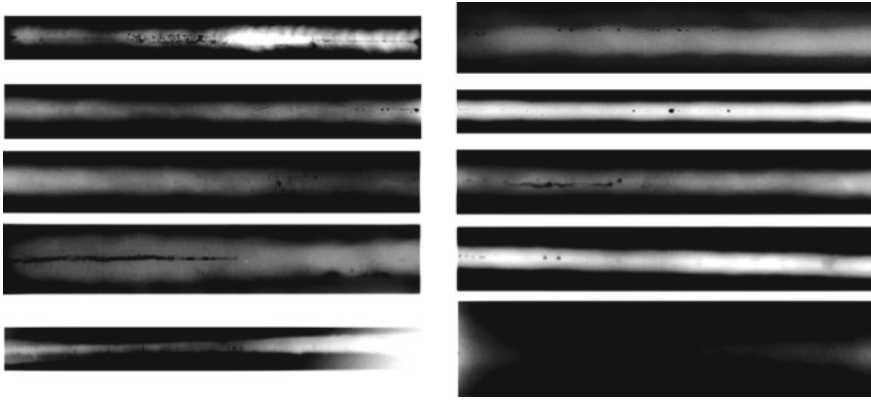


Fig. 9.18 X-ray images used in our experiments (series W0001 of GDXray+)

use the *off-center* saliency map that measures the different dark areas surrounded by a bright background, as shown in Fig. 9.17.

In a training phase, using a priori knowledge of the welding defects, the detection windows are manually labeled as one of two classes: ‘defects’ and ‘no-defect’. The first class corresponds to those regions where the potential welding defects are indeed welding defects. Alternatively, the second class corresponds to false alarms. For this end, we use series W0001 and W0002 of GDXray+. In the first series, we have the X-ray images, whereas in the second one we have the corresponding binary images representing the ground truth. Thus, the ideal segmentation of image W0001_00i.png is binary image W0002_00i.png, for $i = 01 \dots 10$. Intensity features of each channel are extracted for both classes. Features extracted from each area of an X-ray image region are divided into four groups: basic intensity features (see Sect. 5.3.1), statistical features (see Sect. 5.3.5), Fourier and DCT features (see Sect. 5.3.7), Gabor features (see Sect. 5.3.6), and Local Binary Patterns (see Sect. 5.4.1). Afterwards, the extracted features are selected using feature selection approaches (see Sect. 5.6, and several classifiers (see Sect. 6.2) were evaluated using cross-validation (see Sect. 6.3.2). indexGabor features

Experiments

We experimented with 10 representative X-ray images (see Fig. 9.18). The average size of the image was 1.35 mega-pixels. For each X-ray image, 250 detection windows with defects and 250 without defects were selected, yielding $2 \times 250 \times 10 = 5000$ detection windows. Each detection window was labeled with ‘1’ for class *defects* and ‘0’ for *no-defects*. The size of the detection windows were 24×24 pixels. For each detection window 586 features were extracted. This means that 586 features were extracted from 5000 samples (2500 with defects and 2500 without defects) .

After the feature extraction, 75% of the samples from each class were randomly chosen to perform the feature selection. The best performance was achieved using

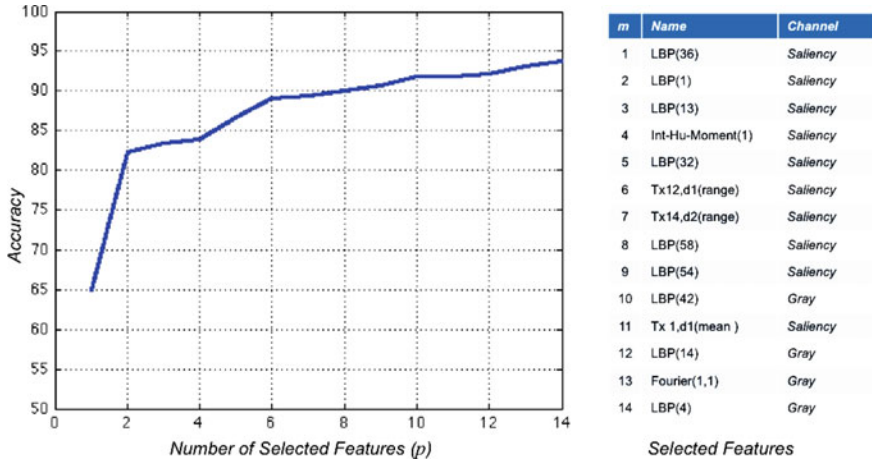


Fig. 9.19 Classification performance using the first *p* features

Sequential Forward Selection. The best 14 features are shown in Fig. 9.19 in ascending order.

The performance of the classification using the SVM classifier and the first *p* selected features was validated using an average of ten cross-validation with 10 folds. The results are shown in Fig 9.19. We observe that by using 14 features, the performance was almost 94% with a 95% confidence interval between 93.0 and 94.5%.

In order to test this methodology on X-ray images, the technique was implemented using a sliding window sized 24×24 pixels that was shifted by 4 pixels. Thus, in each position a sub-window of 24×24 pixels was defined and the corresponding features were extracted. The sub-window was marked if the trained classifier detected it as a discontinuity. Using a size of 24×24 pixel and a shift of 4 pixels, an image pixel could be marked from 0 to 36 times. Finally, if a pixel of the image was marked more than 24 times, then the pixel was considered as a discontinuity. The aforementioned parameters were set using an exhaustive search. The described steps are shown in Fig. 9.20 for one X-ray image. The results on other X-ray images are shown in Fig. 9.21. From these, one can see the effectiveness of the proposed technique.

Conclusions

In this section, we presented a new approach to detecting weld defects without segmentation based on sliding-windows and novel features. The promising results outlined in our work show that we achieved a very high classification rate in the detection of welding defects using a large number of features combined with efficient feature selection and classification algorithms. The key idea of the proposed method was to select, from a large universe of features, namely 572 features, only those features that were relevant for the separation of the two classes. We tested our method

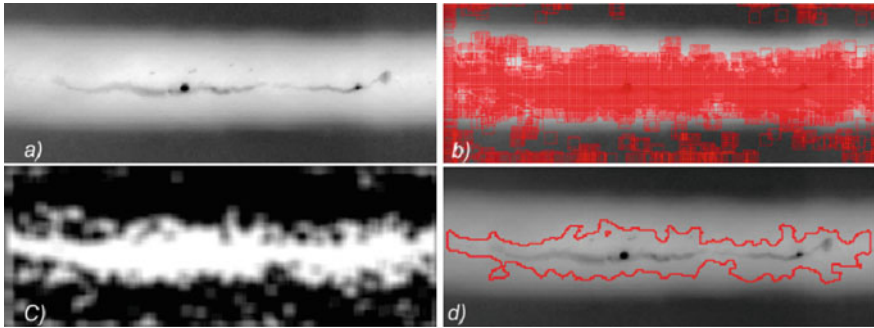



Fig. 9.20 Weld inspection using a sliding-window: **a** X-ray image, **b** detected windows, **c** activation map, **d** detection [102]

on 10 representative X-ray images yielding a performance of 94% in accuracy using only 14 features and support vector machines. It is important to note that local binary pattern features extracted from the saliency map play an important role in the performance of the classifier. The method was implemented and tested on real X-ray images showing high effectiveness.

9.3.3 An Example

In this section, we present a Python code that can be used to detect defects in welds according to sliding-windows approach explained above.

 **Python Example 9.2:** In this example, we show how to implement—for a simple perspective—the strategy explained in the previous section using CNN. We will use one part of image `W0001_0001.png` as training, and another part as testing. Using a sliding windows strategy, we will extract patches of 32×32 pixels on the right side of the image for training and on the left side for testing. These patches are stored in file `welds32x32.mat`. In this example, there are around 10,000 patches for training and other 10,000 for testing. The reader can modify this dataset including more X-ray images of `GDXray+` in order to achieve better results. The reader will note that this example has pedagogical purposes only. In order to develop a real application, more training images must be taken into account.

Listing 9.2 : Detection of weld defects using CNN

```

from pyxvis.learning.cnn import CNN
# execution type
type_exec = 0 # training & testing
# patches' file for training and testing
patches_file = '../data/weld32x32.mat'

```

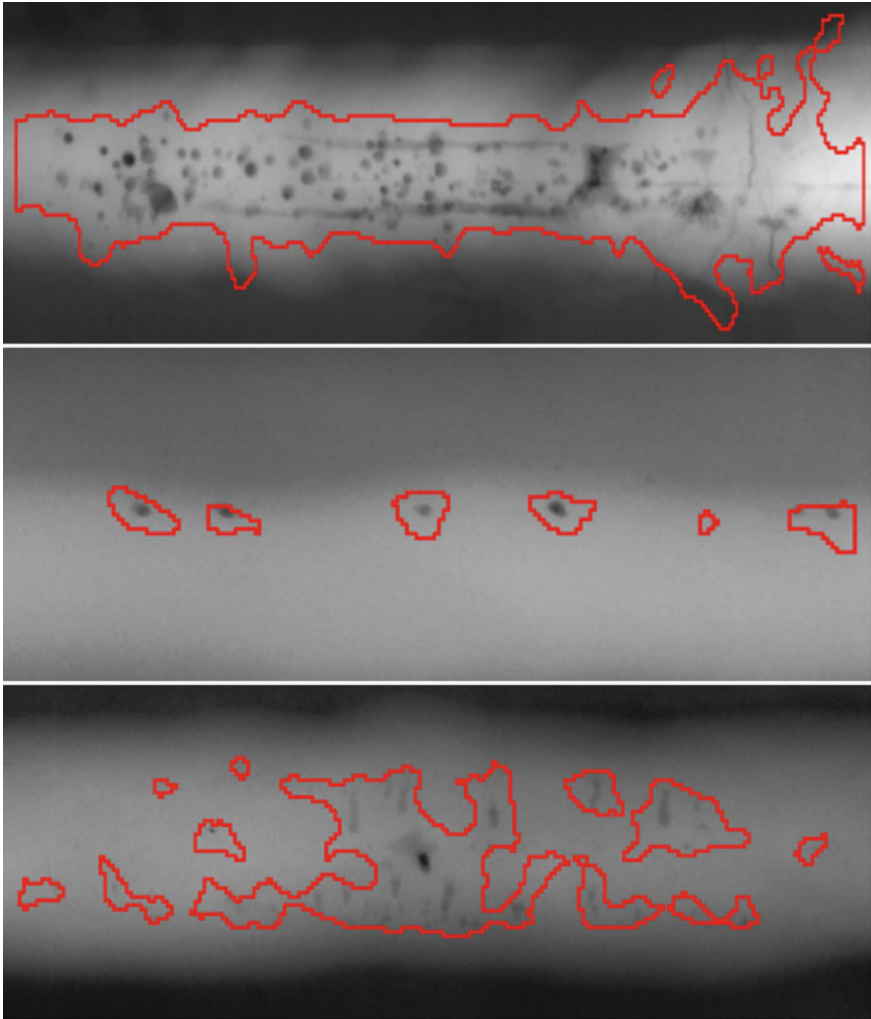


Fig. 9.21 Detection of defects on X-ray images

```
# architecture
p = [9,7,5,3] # Conv2D mask size
d = [32,64,128,256] # Conv2D channels
f = [64,32] # fully connected

# training and testing
CNN(patches_file,type_exec,p,d,f)
```

The output of this code is shown in Fig. 9.22. We can see the final detection on testing image. In this example, we use CNN of pyxvis Library to train the convolutional neural network as explained in Sect. 7.3. The patches are extracted only in the region of

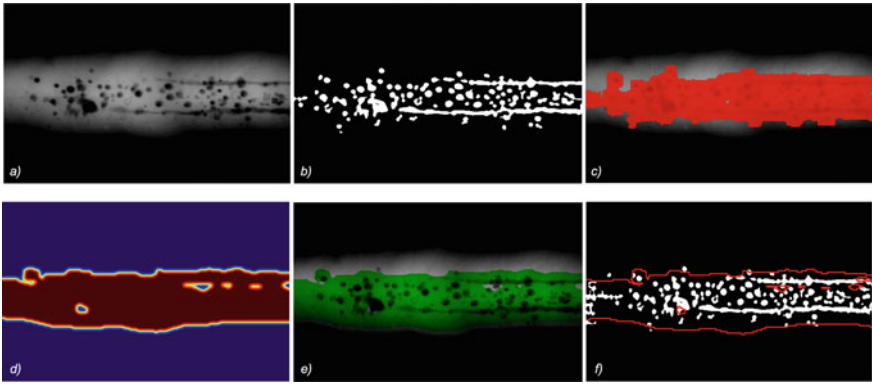


Fig. 9.22 Detection of defects on X-ray images using sliding windows. **a** Testing image. **b** Ground truth (binary image). **c** Ground truth (sliding windows patches of target class). **d** Activation map of the detection. **e** Detection (after thresholding) and testing image **f** Boundary of the detection and binary ground truth. [→ Example 9.2 🗨️]

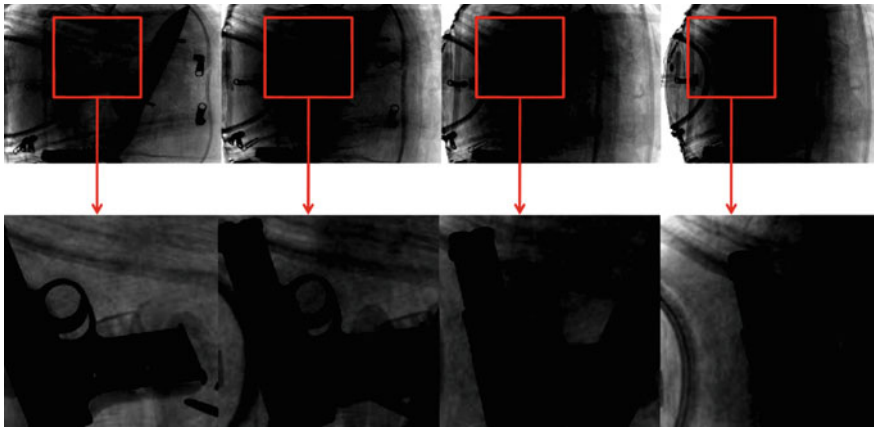


Fig. 9.23 Detection of a handgun based on the trigger identification in multiple views [112]

interest defined by the segmentation of the weld using `seg_bimodal` of `pyxvis` Library. The reader can observe the effectiveness of this strategy. However, it is clear that better results can be achieved by considering more features, classifiers, and training images. □

9.4 Baggage

Since the September 11 attacks, automated (or semi-automated) 3D recognition using X-ray images have become a very important element in baggage screening. The inspection process, however, is complex, basically because threatening items are very difficult to detect when placed in close-packed bags, superimposed by other objects, and/or rotated showing an unrecognizable view [186]. In baggage screening, where human security plays an important role and inspection complexity is very high, human inspectors are still used. Nevertheless, during peak hours in airports, human screeners have only a few seconds to decide whether a bag contains or not a prohibited item, and detection performance is only about 80–90% [117].

9.4.1 State of the Art

Before 9/11, the X-ray analysis of luggage mainly focused on capturing the images of their content: the reader can find in [123] an interesting analysis carried out in 1989 of several aircraft attacks around the world, and the existing technologies to detect terrorist threats based on Thermal-Neutron Activation (TNA), Fast-Neutron Activation (FNA), and dual-energy X-rays (used in medicine since the early 70s). In the 90s, Explosive Detection Systems (EDS) were developed based on X-ray imaging [124], and computed tomography through elastic scatter X-ray (comparing the structure of irradiated material, against stored reference spectra for explosives and drugs) [160]. All these works were concentrated on image acquisition and simple image processing; however, they lacked advanced image analysis to improve detection performance. Nevertheless, the 9/11 attacks increased the security measures taken at airports, which in turn stimulated the interest of the scientific community in the research of areas related to security using advanced computational techniques. Over the last decade, the main contributions were: analysis of human inspection [172], pseudocoloring of X-ray images [1, 25], enhancement and segmentation of X-ray images [156], and detection of threatening items in X-ray images, based on texture features (detecting a 9mm Colt Beretta automatic (machine) pistol) [131], neural networks and fuzzy rules (yielding about 80% of performance) [89], and SVM classifier (detecting guns in real time) [126].

In baggage screening, the use of multiple-view information yields a significant improvement in performance as certain items are difficult to recognize using only one viewpoint. As reported in a study that measures the human performance in baggage screening [17], (human) multiple-view X-ray inspection leads to a higher detection performance of prohibited items under difficult conditions, however, there are no significant differences between the detection performance (single versus multiple view) for difficult-easy multiple-view conditions, i.e., two *difficult* or two *easy* views are redundant. We observed that for intricate conditions, multiple-view X-ray inspection is required.

Recently, some algorithms based on multiple X-ray views were reported in the literature. For example: synthesis of new X-ray images obtained from Kinetic Depth Effect X-ray (KDEX) images based on SIFT features in order to increase detection performance [2]; an approach for object detection in multi-view dual-energy X-ray with promising preliminary results [45]; X-ray active vision that is able to adequate the viewpoint of the target object in order to obtain better X-ray images to analyze [142]; and tracking across multiple X-ray views in order to verify the diagnoses performed using a single view [101, 103, 112, 114].

Finally, methods based on deep learning have been proposed in the last years and they have established themselves as state of the art in baggage inspection. In single views, we can mention [115] using mono-energy and [7–9, 12, 116] using dual-energy. In addition, there are some contributions based on GAN's (see Sect. 7.6) to generate synthetic X-ray images that can be used as data augmentation in the training stage [5, 146, 180]. A review of deep learning method in baggage inspection can be found in [6].

An example is illustrated in Fig. 9.23. A survey on explosives detection can be found in [157, 176]. Selected approaches are summarized in Table 9.5. In baggage screening, where human security plays an important role and inspection complexity is very high, human inspectors are still used. For intricate conditions, multiple-view X-ray inspection using dual-energy is required.

9.4.2 An Application

In this section, we present the use of an automated method based on multiple X-ray views to recognize certain regular objects with highly defined shapes and sizes. The method consists of two steps: ‘monocular analysis’, to obtain possible detections in each view of a sequence, and ‘multiple-view analysis’, to recognize the objects of interest using matchings in all views. The search for matching candidates is efficiently performed using a lookup table that is computed off-line. In order to illustrate the effectiveness of the proposed method, experimental results on recognizing regular objects (clips, springs, and razor blades) in pen cases are shown. In this section, we explain in further detail the proposed method. The strategy consists of two main stages: *off-line* and *on-line*.

Off-Line Stage

The first stage, performed off-line, consists of two main steps: (i) learning a model that is used for the recognition and (ii) estimation of a multiple-view geometric model that is used for data association.

Learning: In this step, we learn a classifier h to recognize parts of the objects that we are attempting to detect. It is assumed that there are $C + 1$ classes (labeled as ‘0’ for non-object class, and ‘1’, ‘2’, ... ‘C’ for C different objects). Images are taken of representative objects of each class from different points of view. In order to model the

Table 9.5 State of art in baggage inspection

Authors	Year	Ref	X_1^* energies 1 2 3	X_2^* views 1 2 3	X_3^* algorithms 1 2 3
Abusaeeda et al.	2011	[2]	☒☒☐	☒☒☐	☒☒☐
Akcaý and Breckon	2017	[7]	☐☒☐	☒☐☐	☐☐☒
Akcaý et al.	2016	[8]	☐☒☐	☒☐☐	☐☐☒
Akcaý et al.	2018	[9]	☐☒☐	☒☐☐	☐☐☒
Akcaý et al.	2018	[5]	☐☒☐	☒☐☐	☐☐☒
Aydin et al.	2018	[12]	☒☐☐	☐☒☐	☐☒☐
Baştan	2015	[15]	☐☒☐	☐☒☐	☐☒☐
Baştan et al.	2011	[16]	☐☒☐	☒☐☐	☐☒☐
Chen et al.	2005	[26]	☒☒☐	☒☐☐	☒☒☐
Ding et al.	2006	[31]	☒☒☐	☒☐☐	☒☒☐
Franzel et al.	2012	[45]	☐☒☐	☐☒☐	☐☒☐
Flitton et al.	2013	[43]	☐☒☐	☐☐☒	☐☒☐
Flitton et al.	2015	[44]	☐☒☐	☐☐☒	☐☒☐
Heitz and Chechik	2010	[60]	☒☒☐	☒☐☐	☒☒☐
Liu et al.	2018	[90]	☐☒☐	☒☐☐	☐☐☒
Lu and Connors	2006	[92]	☒☒☐	☒☐☐	☒☒☐
Mansoor and Rajashankari	2012	[94]	☒☒☐	☒☐☐	☒☒☐
Mery	2015	[103]	☒☐☐	☐☒☐	☐☒☐
Mery et al.	2013	[114]	☒☐☐	☐☒☐	☐☒☐
Mery et al.	2016	[115]	☒☐☐	☒☐☐	☐☒☒
Miao et al.	2019	[116]	☐☒☐	☒☐☐	☐☐☒
Mouton and Breckon	2015	[119]	☐☒☐	☐☐☒	☐☒☐
Nercessian et al.	2008	[127]	☒☐☐	☒☐☐	☒☐☐
Riffo and Mery	2016	[143]	☒☐☐	☒☐☐	☐☒☐
Riffo and Mery	2012	[142]	☒☐☐	☐☒☐	☐☒☐
Riffo et al.	2019	[141]	☒☐☐	☐☐☒	☐☒☐
Riffo et al.	2017	[140]	☒☐☐	☐☒☐	☐☒☐
Saavedra et al.	2020	[145]	☒☐☐	☒☐☐	☐☐☒
Sangwan and Jain	2019	[146]	☒☐☐	☒☐☐	☐☐☒
Sigman and Jain	2020	[151]	☒☐☐	☒☐☐	☐☐☒
Schmidt et al.	2012	[147]	☒☒☐	☒☐☐	☒☒☐
Steitz et al.	2018	[159]	☐☒☐	☐☒☐	☐☐☒
Turcsany et al.	2013	[165]	☐☒☐	☒☐☐	☐☒☐
Uroukov and Speller	2015	[167]	☒☒☐	☒☐☐	☒☒☐
Yuanxi and Liu	2019	[183]	☒☐☐	☒☐☐	☐☐☒
Xu et al.	2018	[178]	☐☒☐	☒☐☐	☐☐☒
Zhang and Zhue	2015	[188]	☒☒☐	☒☐☐	☒☒☐
Zou et al.	2018	[194]	☒☐☐	☒☐☐	☐☐☒
		* 1	Mono	Mono	Simple
		2	Dual	Multi	Medium
		3	Multi	CT	Complex

☐ not used, ☒ used

details of the objects from different poses, several keypoints per image are detected, and for each keypoint a descriptor \mathbf{d} is extracted using, for example, LBP, SIFT, HOG, and SURF, among others (see Sect. 5.4). In this supervised approach, each descriptor \mathbf{d} is manually labeled according to its corresponding class $c \in \{0, 1, \dots, C\}$. Given the training data (\mathbf{d}_t, c_t) , for $t = 1, \dots, N$, where N is the total number of descriptors extracted in all training images, a classifier h is designed which maps \mathbf{d}_t to their classification label c_t , thus, $h(\mathbf{d}_t)$ should be c_t . This classifier will be used in the on-line stage by monocular and multiple-view analysis.

Geometry: Our strategy deals with multiple monocular detections in multiple views. In this problem of data association, the aim is to find the correct correspondence among different views. For this reason, we use multiple-view geometric constraints to reduce the number of matching candidates between monocular detections. For an image sequence with n views $\mathbf{I}_1 \dots \mathbf{I}_n$, the fundamental matrices $\{\mathbf{F}_{ij}\}$ between consecutive frames \mathbf{I}_i and $\mathbf{I}_{j=i+1}$ are computed for $i = 1, \dots, n-1$. In our approach, the fundamental matrix \mathbf{F}_{ij} is calculated from projection matrices \mathbf{P}_i and \mathbf{P}_j that can be estimated using calibration (see Sect. 3.4) or bundle adjustment algorithms (see Sect. 9.4.3).

The geometric constraints are expressed in homogeneous coordinates. Therefore, given a point $\mathbf{m}_i = [x_i \ y_i \ 1]^T$ in image \mathbf{I}_i , a corresponding point $\mathbf{m}_j = [x_j \ y_j \ 1]^T$ in image \mathbf{I}_j must fulfill: (i) epipolar constraint (see Sect. 3.5.1): \mathbf{m}_j must lie near the epipolar line $\ell = \mathbf{F}_{ij}\mathbf{m}_i$, and (ii) location constraint: for small variations of the point of views between \mathbf{I}_i and \mathbf{I}_j , \mathbf{m}_j must lie near \mathbf{m}_i . Thus, a candidate \mathbf{m}_j must fulfill:

$$\frac{|\mathbf{m}_j^T \mathbf{F}_{ij} \mathbf{m}_i|}{\sqrt{\ell_1^2 + \ell_2^2}} < e \text{ and } \|\mathbf{m}_i - \mathbf{m}_j\| < r. \quad (9.5)$$

In order to accelerate the search of candidates, we propose the use of a lookup table as follows: Points in images \mathbf{I}_i and \mathbf{I}_j are arranged in a grid format with rows and columns. For each grid point (x, y) of image \mathbf{I}_i , we look for the grid points of image \mathbf{I}_j that fulfill (9.5), as illustrated in Fig. 9.24. Therefore, the possible corresponding points of (x, y) will be the set $\mathbf{S}_{xy} = \{(x_p, y_p)\}_{p=1}^q$, where $x_p = X(x, y, p)$, $y_p = Y(x, y, p)$ and $q = Q(x, y)$ are stored (off-line) in a lookup table. In the on-line stage, given a point \mathbf{m}_i (in image \mathbf{I}_i), the matching candidates in image \mathbf{I}_j are those that lie near to \mathbf{S}_{xy} , where (x, y) is the nearest grid point to \mathbf{m}_i . This search can be efficiently implemented using k -d tree structures [21].

In a controlled and calibrated environment, we can assume that the fundamental matrices are stable and we do not need to estimate them in each new image sequence, i.e., the lookup tables are constant. Additionally, when the relative motion of the point of view between consecutive frames is the same, the computed fundamental matrices are constant, i.e., $\mathbf{F}_{ij} = \mathbf{F}$, and we need to store only one lookup table.

On-Line Stage

The on-line stage is performed in order to recognize the objects of interest in a test image sequence of n images $\{\mathbf{I}_i\}$, for $i = 1, \dots, n$. The images are acquired by

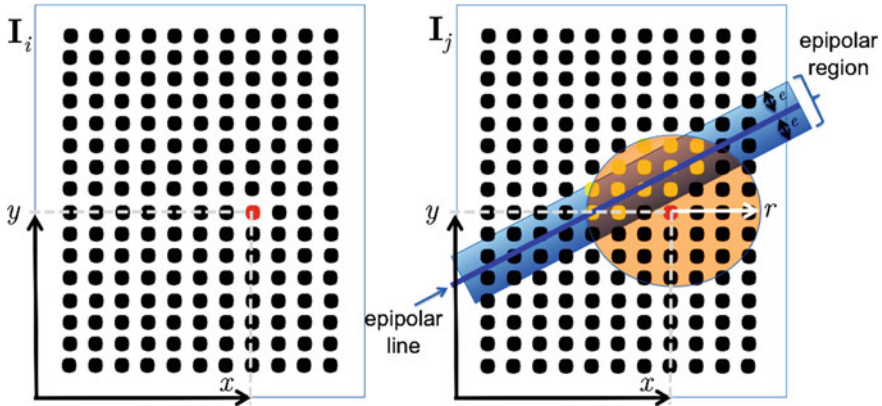


Fig. 9.24 Given the grid point illustrated as the red point at (x, y) , in image I_i , the set of possible corresponding points in image I_j can be those grid points (yellow points) represented by the intersection of the epipolar region (blue rectangle) and neighborhood around (x, y) (orange circle with radius r centered at red point). The use of grid points allows us to use a lookup table in order to search the matching candidates in I_j efficiently

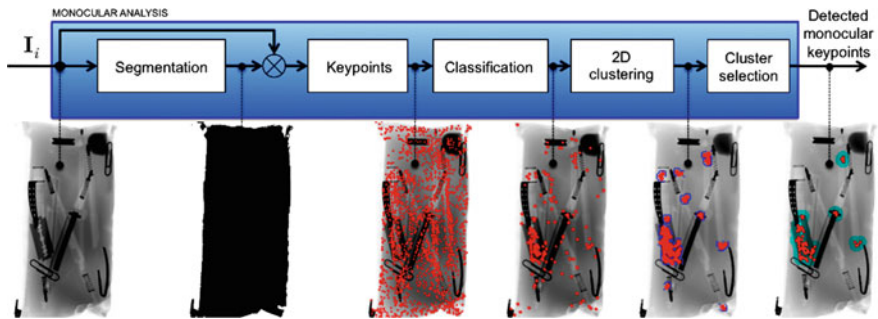


Fig. 9.25 Monocular analysis for each image of the sequence, i.e., for $i = 1, \dots, n$. In this example, the class of interest is ‘razor blade’

rotation of the object being tested at β degrees (in our experiments we used $n = 4$, and $\beta = 10^0$). This stage consisted of two main steps: monocular and multiple-view analysis that will be described in further detail as follows.

Monocular Analysis: This step is performed in each image I_i of the test image sequence, as illustrated in Fig. 9.25 in a real case. The whole object contained in image I_i is segmented from the background using threshold and morphological operations. SIFT-keypoints (or other descriptors)— are only extracted in the segmented portion. The descriptor \mathbf{d} of each keypoint is classified using classifier $h(\mathbf{d})$ trained in the off-line stage, and explained above. All keypoints classified as class c , where c is the class of interest, with $c \in \{1 \dots C\}$ are selected. As we can see in Fig. 9.25 for the classification of ‘razor blade’, there are many keypoints misclassified. For this

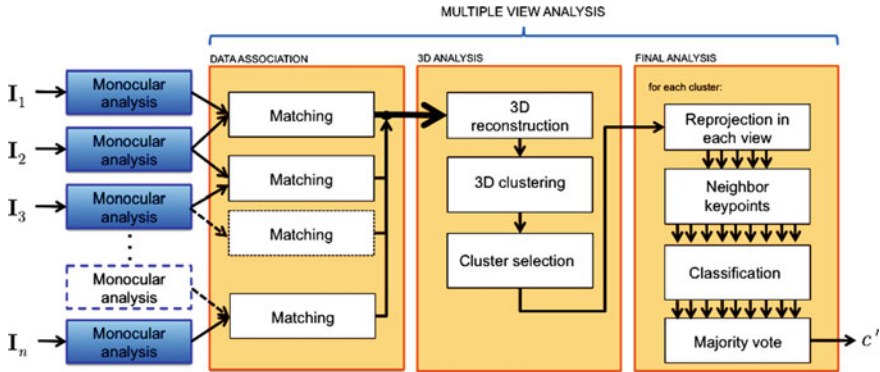


Fig. 9.26 Multiple-view analysis. An explanation of last step (final analysis) is illustrated in Fig. 9.27

reason, neighbor keypoints are clustered in the 2D space using Mean Shift algorithm [28]. Only those clusters that have a large enough number of keypoints are selected. They will be called *detected monocular keypoints*.

Multiple-View Analysis: Multiple view analysis performs the recognition of objects of interest in three steps (see Fig. 9.26): (i) data association, (ii) 3D analysis, and (iii) final analysis. The input is the detected monocular keypoints obtained by the mentioned monocular analysis explained above. The output is c' , the assigned class for each detected object.

• **Data Association:** In this step, we find matchings for all detected monocular keypoints in all consecutive images I_i and $I_{j=i+1}$, for $i = 1, \dots, n - 1$, as follows:

- + For each detected monocular keypoint in image I_i (located at position (x_i, y_i) with descriptor \mathbf{d}_i), we seek in a dense grid of points, the nearest point (x, y) (see red point in Fig. 9.24-left) using a k -d tree structure.
- + We determine S_{xy} , the set of matching candidates in image $I_{j=i+1}$ arranged in a grid manner by reading the lookup table explained above (see yellow points in Fig. 9.24-right).
- + We look for the detected monocular keypoints in image I_j that are located in the neighborhood of S_{xy} , again using a k -d tree structure. They will be called *neighbor keypoints*. When no neighbor keypoint is found, no match is established for (x_i, y_i) .
- + From neighbor keypoints, we select that one (located at position (x_j, y_j) with descriptor \mathbf{d}_j) with minimum distance $\|\mathbf{d}_i - \mathbf{d}_j\|$. In order to ensure the similarity between matching points, the distance should be less than a threshold ε . If this constraint is not satisfied, again no match is established for (x_i, y_i) .

• **3D analysis:** From each pair of matched keypoints (x_i, y_i) in image I_i and (x_j, y_j) in image $I_{j=i+1}$ established in the previous step, a 3D point is reconstructed using the projection matrices \mathbf{P}_i and \mathbf{P}_j of our geometric model (see Sect. 3.6). Similar to

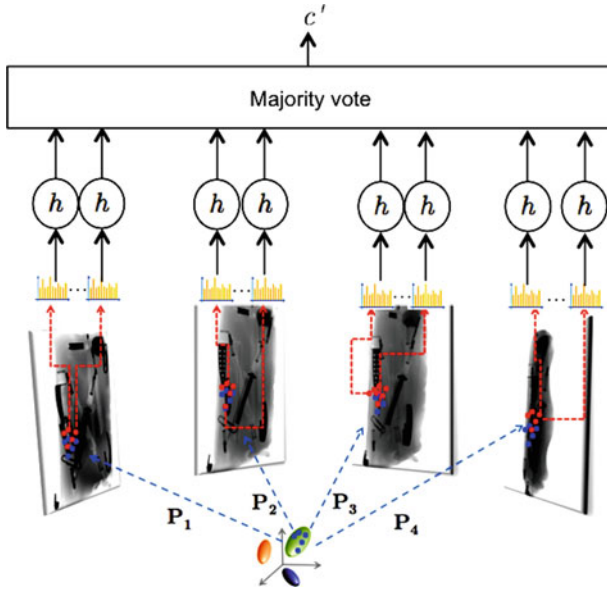


Fig. 9.27 Final analysis: using the geometric model, the reconstructed 3D points in each cluster are reprojected in each view (blue points). The keypoints that are near to the reprojected points are identified (red points). The descriptors of these keypoints (orange histograms) are classified using trained classifier h . The class c' of this cluster is determined by majority vote. In this example of $n = 4$ views, only the green cluster is represented

the monocular detection approach, neighbor 3D points are clustered in the 3D space using Mean Shift algorithm [28], and only those clusters that have a large enough number of 3D points are selected.

- Final analysis: For each selected 3D cluster, all 3D reconstructed points belonging to the cluster are re-projected onto all images of the sequence using the projection matrices of geometric model (see Fig. 9.27). The extracted descriptors of the keypoints located near these re-projected points are classified individually using classifier h . The cluster will be classified as class c' if there is a large number of keypoints individually classified as c' , and this number represents a majority in the cluster.

This majority vote strategy can overcome the problem of false monocular detections when the classification of the minority fails. A cluster can be misclassified if the part that we are trying to recognize is occluded by a part of another class. In this case, there will be keypoints in the cluster assigned to both classes; however, we expect that the majority of keypoints will be assigned to the true class if there are a small number of keypoints misclassified.

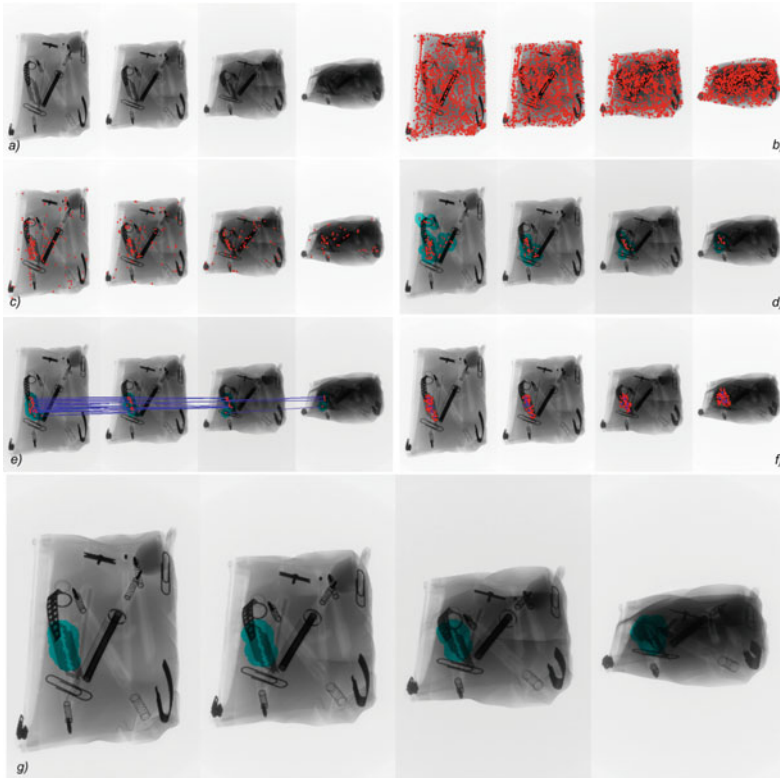


Fig. 9.28 Recognition of a razor blade using our approach. **a** original sequence, **b** keypoints, **c** classified keypoints, **d** detected monocular keypoints, **e** matched keypoints, **f** reprojected 3D points (blue) and neighbor keypoints (red), **g** final detection

Experiments and Results

In our experiments, the task was to recognize three different classes of objects that are present in a pencil case (see, for example, a sequence in Fig. 9.28a). These classes are: ‘clips’, ‘springs’, and ‘razor blades’. We followed the recognition approach explained above.

In the off-line stage we used a structure from a motion algorithm in order to estimate the projection matrices of each view⁴. Additionally, in the learning phase, we used only 16 training images of each class. Due to the small intra-class variation of our classes, this number of training images was deemed sufficient. The training objects were posed at different angles. SIFT descriptors were extracted as explained in [91], and a *k*-Nearest Neighbor (KNN) classifier with *k* = 3 neighbors was ascertained

⁴We use in our experiments a fast implementation of multiple-view geometry algorithms from Balu Toolbox [97].

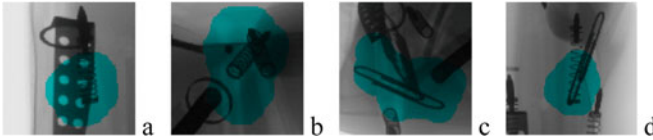


Fig. 9.29 Recognition using our approach in cases with some degree of overlap: **a** one spring, **b** two springs, **c** one clip, **d** one clip. Each figure shows a part of one image of the whole sequence

using the SIFT descriptors of the four classes⁵. Other descriptors (like LBP and HOG) and other classifiers (like SVM or KNN with other values of k) were also tested, although the best performance was achieved with the aforementioned configuration.

In order to illustrate step by step the on-line stage, the recognition of a razor blade is illustrated in Fig. 9.28a–d for monocular analysis and in Fig. 9.28e–g for multiple-view analysis⁶. It is worth mentioning that in monocular detection there are false alarms, however, they can be filtered out after multiple-view analysis. The reason is because false alarms cannot be tracked in the sequence or because the tracked points, when validating the corresponding points in other views of the sequence, do not belong to the class of interest. Other results with some degree of overlap, where the task was the recognition of springs and clips, are illustrated in Fig 9.29.

Testing experiments were carried out by recognizing the three mentioned classes (‘clips’, ‘springs’, and ‘razor blades’) in 45 different sequences of 4 views (15 sequences for each class)⁷. The size of an individual image was 1430×900 pixels. In these experiments there were 30 clips, 75 springs and 15 razor blades to be recognized. A summary of the results using the proposed algorithm is presented in Table 9.6, in which the performance in the recognition of each class is presented in two different parts of our algorithm: after monocular analysis (Mono) and after multiple-view analysis (Multi). These parts are illustrated in Fig. 9.28d and 9.28g respectively for a razor blade. In this table, Ground Truth (GT) is the number of existing objects to be recognized. The number of detected objects by our algorithm is $D = TP + FP$, including False Positives (FP) and true positives (TP). Ideally, $FP = 0$ and $TP = GT$. In our experiments, precision (PR), computed as $PR = TP/D$, is 71.4% and 95.7% in each part; and recall (RE), computed as $RE = TP/GT$, is 90.8% and 92.5% in each step. If we compare single versus multiple view detection, both precision and recall are incremented. Precision, however, is drastically incremented because our approach achieves good discrimination from false alarms.

The amount of time required in our experiments was about 15 minutes for the off-line stage and about 16s for testing each sequence on a iMac OS X 10.7.3, processor 3.06 GHz Intel Core 2 Duo, 4 GB 1067 MHz DDR3 memory. The code of the program—implemented in Matlab—is available on our website.

⁵We used in our experiments fast implementations of SIFT and KNN (based on k -d tree) from VLFeat Toolbox [169].

⁶We used in our experiments a fast implementation of Mean Shift from PMT Toolbox [32].

⁷The images tested in our experiments come from public GDXray database [113].

Table 9.6 Recognition performance

		Mono			Multi	
Class	TP	FP	GT	TP	FP	GT
Clip	114	127	120	26	2	30
Spring	263	30	300	71	3	75
Blade	59	18	60	14	0	15
Total	436	175	480	111	5	120
Precision		71.4%			95.7%	
Recall		90.8%			92.5%	

Conclusions

In this section, we presented a new method that can be used to recognize certain parts of interest in complex objects using multiple X-ray views. The proposed method filters out false positives resulting from monocular detection performed on single views by matching information across multiple views. This step is performed efficiently using a lookup table that is computed off-line. In order to illustrate the effectiveness of the proposed method, experimental results on recognizing regular objects—clips, springs, and razor blades—in pen cases are shown achieving around 93% accuracy in the recognition of 120 objects. We believe that it would be possible to design an automated aid in a target detection task using the proposed algorithm. In our future work, the approach will be tested in more complex scenarios recognizing objects with a larger intra-class variation.

9.4.3 An Example Using Multiple Views

In this example, we show how to detect objects in a non-calibrated image sequence as illustrated in Fig. 9.31. The approach has two parts: *structure estimation* and *parts detection*⁸. The approach follows the same strategy of method explained in Sect. 9.2.2. The results are shown in Fig. 9.32.

Structure Estimation

In case the X-ray imaging system is not calibrated, a geometric model must be estimated. The estimation of the geometric model is based on well-known structure-from-motion (SfM) methodologies. For the sake of completeness, a brief description of this model is presented here. In our work, SfM is estimated from a sequence of m images taken from a rigid object at different viewpoints. The original image sequence is stored in m images $\mathbf{J}_1, \dots, \mathbf{J}_m$.

⁸See implementation in Matlab at <https://github.com/ Domingomery/Xvis> - function Xtrgui.

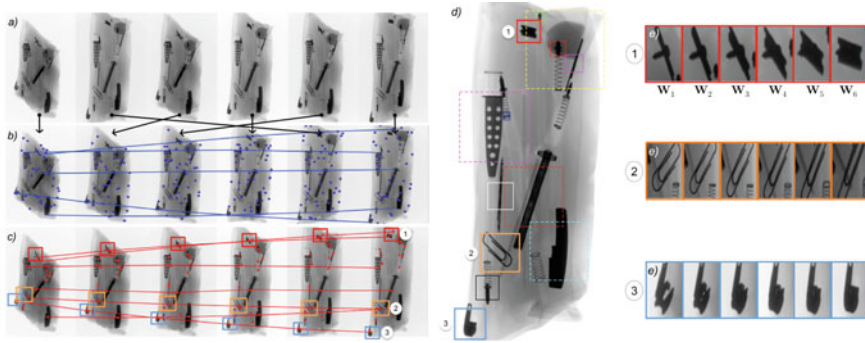


Fig. 9.30 Detection of objects in a pencil case using the proposed method: **a** Unsorted sequence with six X-ray images. The images are sorted according to their similarity (see arrows). **b** Sorted sequence, keypoints (points) and structure from motion (lines across the sequence). **c** Detection in the sequence and tracked regions. **d** Detection of parts of interest in the last image in the sequence (three of them are used in this example to illustrate the next sub-figures). **e** Tracked example regions in each view of the sequence (1: pencil sharpener, 2: clip, and 3: zipper slider body and pull-tab)

Keypoints: For each image, SIFT keypoints are extracted because they are very robust against scale, rotation, viewpoint, noise, and illumination changes [91]. Thus, not only a set of 2D image positions \mathbf{x} , but also descriptors \mathbf{y} , are obtained. Although this method is based on SIFT descriptors, there is no limitation to use other descriptors, e.g., SURF [18].

Image Sorting: If the images are not sorted, a visual vocabulary tree is constructed for fast image indexing. Thus, a new image sequence $\mathbf{I}_1, \dots, \mathbf{I}_m$ is established from $\mathbf{J}_1, \dots, \mathbf{J}_m$ by maximizing the total similarity defined as $\sum \text{sim}(\mathbf{I}_i, \mathbf{I}_{i+1})$, for $i = 1, \dots, m - 1$, where the similarity function ‘sim’ is computed from a normalized scalar product obtained from the visual words of the images [158]. See an example in Fig. 9.30a and 9.30b.

Matching Points: For two consecutive images, \mathbf{I}_i and \mathbf{I}_{i+1} , SIFT keypoints are matched using the algorithm suggested by Lowe [91] that rejects too ambiguous matches. Afterwards, the Fundamental Matrix between views i and $i + 1$, $\mathbf{F}_{i,i+1}$, is estimated using RANSAC [56] to remove outliers. If keypoint k of \mathbf{I}_i is matched with keypoint k' of \mathbf{I}_{i+1} , the match will be represented as $\mathbf{x}_{i,k} \rightarrow \mathbf{x}_{i+1,k'}$.

Structure Tracks: We look for all possible structure tracks—with one keypoint in each image of sequence—that belong to a family of the following matches:

$$\mathbf{x}_{1,k_1} \rightarrow \mathbf{x}_{2,k_2} \rightarrow \mathbf{x}_{3,k_3} \rightarrow \dots \rightarrow \mathbf{x}_{m,k_m}.$$

There are many matches that are eliminated using this approach, however, having a large number of keypoints there are enough tracks to perform the bundle adjustment. We define n as the number of tracks.

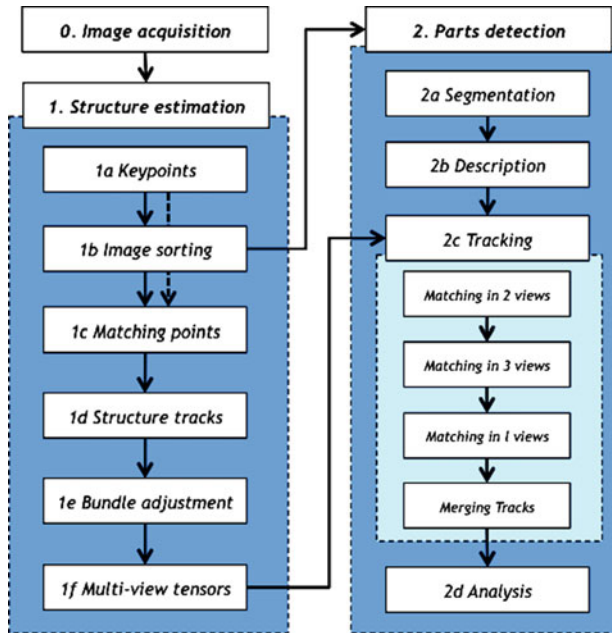


Fig. 9.31 Block diagram of the proposed approach

Bundle Adjustment: The determined tracks define n image point correspondences over m views. They are arranged as $\mathbf{x}_{i,j}$ for $i = 1, \dots, m$ and $j = 1, \dots, n$. Bundle adjustment estimates 3D points $\hat{\mathbf{X}}_j$ and camera matrices \mathbf{P}_i so that $\sum \|\mathbf{x}_{i,j} - \hat{\mathbf{x}}_{i,j}\|$ is minimized, where $\hat{\mathbf{x}}_{i,j}$ is the projection of $\hat{\mathbf{X}}_j$ by \mathbf{P}_i . If $n \geq 4$, we can use the *factorization algorithm* [56] to perform an affine reconstruction because for our purposes the affine ambiguity of 3D space is irrelevant⁹. This method gives a fast and closed-form solution using SVD decomposition. A RANSAC approach is used to remove outliers.

Multiple-View Tensors: Bundle adjustment provides a method for computing bifocal and trifocal tensors from projection matrices \mathbf{P}_i [56], that will be used in the next section.

Parts Detection

In this section, we give details of the algorithm that detects the object parts of interest. The algorithm consists of four steps: segmentation, description, tracking, and analysis as shown in Fig. 9.31.

⁹In this problem, the projective factorization can be used as well [56], however, our simplifying assumption is that only small depth variations occur and an affine model may be used.

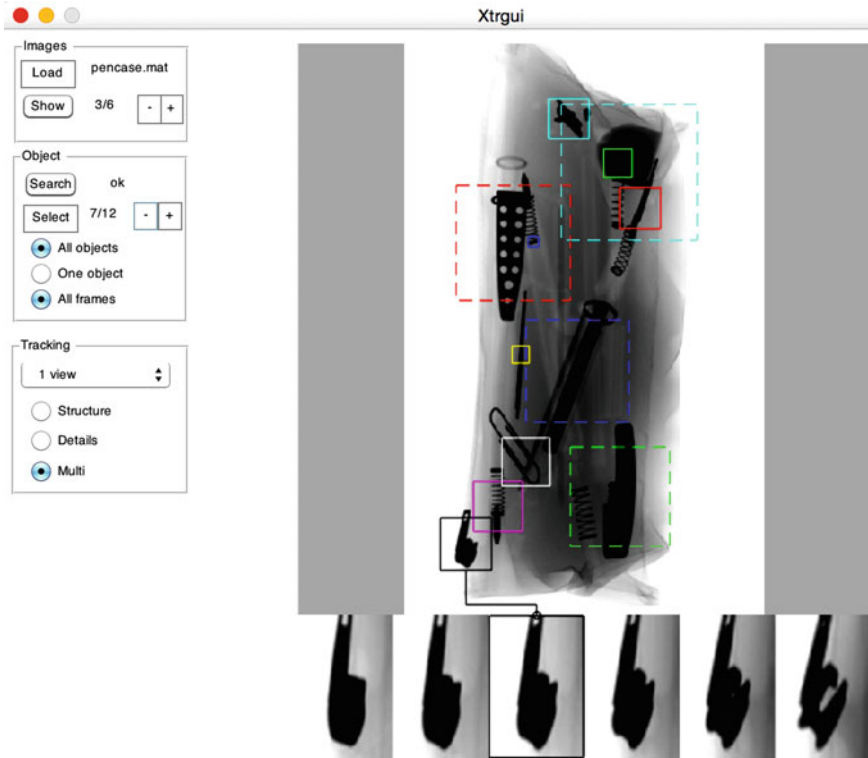


Fig. 9.32 Detection of objects in a pen case using graphic user interface Xtrgui [103]. In this example, we can see the zipper slider body and pull-tab in six different views

Segmentation: Potential regions of interest are segmented in each image I_i of the sequence. It is an *ad-hoc* procedure that depends on the application. For instance, one can be interested in detecting razor blades or pins in a bag, or flaws in a material, etc. This step ensures the detection of the object parts of interest allowing false detections. The discrimination between these two classes takes place by tracking them across multiple views (see steps 2c and 2d). In our experiments we tested three segmentation approaches.

- Spots detector: The X-ray image is filtered using a 2D median filter. The difference between original and filtered images is thresholded obtaining a binary image. A potential region r is segmented if size, shape, and contrast criteria are fulfilled. This approach was used to detect small parts (like pen tips or pins in a pencil case).
- Crossing line profile (CLP): Laplacian of Gaussian edges are computed from the X-ray image. The closed and connected contours of the edge image define region candidates. Gray level profiles along straight lines crossing each region candidate in the middle are extracted. A potential region r is segmented if the profile that contains the most similar gray levels in the extremes fulfills contrast criteria [99].

This approach was used to detect discontinuities in a homogeneous material, e.g., flaws in automotive parts.

- **SIFT matching:** SIFT descriptors are extracted from the X-ray image. They are compared with SIFT descriptors extracted from the image of a reference object of interest. A potential region r is segmented if the descriptors fulfill similarity criteria [49, 91]. This approach was used to detect razor blades in a bag.

Other general segmentation approaches can be used as well. For example, methods based on saliency maps [118], Haar basis features [171], histogram of oriented gradients [30], corner detectors [55], SURF descriptors [18], Maximally Stable regions [95], Local Binary Patterns [133], etc.

Description: Each segmented potential region r is characterized using a SIFT descriptor. The scale of the extracted descriptor, i.e., the width in pixels of the spatial histogram of 4×4 bins is set to $\sqrt{A_r}$, where A_r is the corresponding area of the region r .

Tracking and Analysis: The tracking and analysis algorithms were covered in detail in Sect. 9.2.2. Results are shown in Fig. 9.30.

9.4.4 Example Using Deep Learning

In Sect. 7.7.6, we illustrated already many examples in baggage inspection using deep learning methods for object detection. The reader is referred to those examples and Sect. 7.7 to see the detection methods that are proposed for baggage inspection. Here, we include additional results in Fig. 9.33 to illustrate an example using deep learning.

9.5 Natural Products

In order to ensure food safety inspection, several applications have been developed by the natural products industry. The difficulties inherent in the detection of defects and contaminants in food products have limited the use of X-ray into the packaged foods sector. However, the need for NDT has motivated a considerable research effort in this field spanning many decades [54].

9.5.1 State of the Art

The most important advances are: detection of foreign objects in packaged foods [76]; detection of fish bones in fishes [111]; identification of insect infestation in

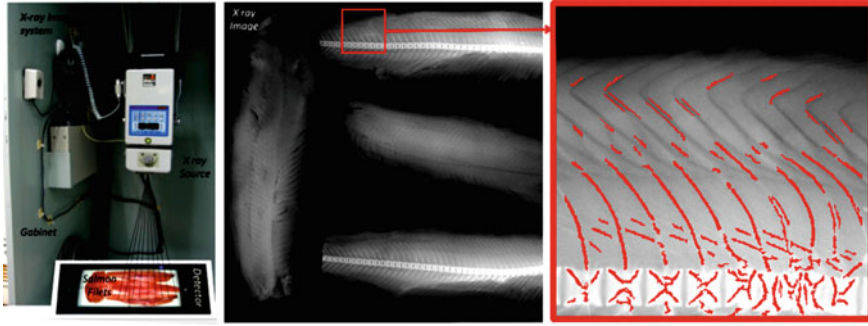


Fig. 9.34 Detection of fish bones using sliding-windows [111]

9.7, some applications are summarized. We observe that deep learning methods in this field are rarely used. This trend is sure to change in the next few years.

9.5.2 An Application

In countries where fish is often consumed, fish bones are some of the most frequently ingested foreign bodies encountered in foods. In the production of fish fillets, fish bone detection is performed by human inspection using their sense of touch and vision which can lead to misclassification. Effective detection of fish bones in the quality control process would help avoid this problem. For this reason, an X-ray machine vision approach to automatically detect fish bones in fish fillets was developed. This section describes our approach to detect fish bones automatically and the corresponding experiments with salmon and trout fillets based on [111].

Pre-Processing and Segmentation

The fish bones are only present in certain space frequencies of the spectrum: they are not too thin (minimal 0.5mm) nor too thick (maximal 2mm). The segmentation of potential fish bones is based on a band-pass filter using their spectrum with respect to their surroundings as shown in Fig. 9.35. The proposed approach to detect potential fish bones has four steps:

Enhancement: The original X-ray image X (Fig. 9.35b) is enhanced linearly by modifying the original histogram in order to increase contrast [50]: The enhanced image Y is

$$Y = aX + b \tag{9.6}$$

Band-Pass Filtering: The enhanced image Y is filtered using a radial symmetric 17×17 pixels mask H (Fig. 9.35a). Mask H was estimated from 20 X-ray images

Table 9.7 State of art on natural products

Authors	Year	Ref	\mathbb{X}_1^* energies	\mathbb{X}_2^* views	\mathbb{X}_3^* algorithms
			1 2 3	1 2 3	1 2 3
Bej et al.	2015	[19]	☒ □ □	☒ □ □	☒ ☒ □
van Deal et al.	2016	[29]	☒ □ □	☒ □ □	☒ ☒ □
van Deal et al.	2019	[168]	☒ □ □	□ □ ☒	☒ ☒ □
Douarre et al.	2016	[33]	☒ □ □	□ □ ☒	☒ ☒ □
Guelpa et al.	2015	[52]	☒ □ □	☒ □ □	☒ ☒ □
Haff and Slaughter	2004	[53]	☒ □ □	☒ □ □	☒ ☒ □
Jiang et al.	2008	[65]	☒ □ □	☒ □ □	☒ ☒ □
Karunakaran et al.	2004	[69]	☒ □ □	☒ □ □	☒ ☒ □
Kelkar et al.	2015	[70]	☒ □ □	☒ □ □	☒ ☒ □
Kotwaliwale et al.	2014	[73]	☒ □ □	☒ □ □	☒ ☒ □
Kwon et al.	2008	[76]	☒ □ □	☒ □ □	☒ ☒ □
Mathanker et al.	2011	[96]	☒ □ □	☒ □ □	☒ ☒ □
Mery et al.	2011	[111]	☒ □ □	☒ □ □	☒ ☒ □
Neethirajan et al.	2014	[125]	☒ □ □	☒ □ □	☒ ☒ □
Nielsen et al.	2014	[128]	☒ □ □	☒ □ □	☒ ☒ □
Nugraha et al.	2019	[130]	☒ □ □	□ □ ☒	☒ ☒ □
Ogawa et al.	2003	[132]	☒ □ □	☒ □ □	☒ ☒ □
Orina et al.	2018	[134]	☒ □ □	☒ □ □	☒ ☒ □
Schoeman et al.	2016	[148]	☒ □ □	□ □ ☒	☒ ☒ □
van Deal et al.	2019	[168]	☒ □ □	□ □ ☒	☒ ☒ □
Zhong et al.	2019	[191]	☒ □ □	☒ □ □	☒ ☒ □
		*1	Mono	Mono	Simple
		2	Dual	Multi	Medium
		3	Multi	CT	Complex

□ not used, ☒ used

by minimizing the error rate as mention in [23] and applied to fish bones (all fish bones should be found and there should be no false alarms). The filtered image \mathbf{Z} (Fig. 9.35c) is then the convolution of \mathbf{Y} with mask \mathbf{H} :

$$\mathbf{Z} = \mathbf{Y} * \mathbf{H} \tag{9.7}$$

Thresholding: Those pixels in \mathbf{Z} that have gray values greater than a certain threshold θ are marked in a binary image \mathbf{B} . The threshold is defined to ensure that all fish bones are detected, i.e., false alarms are allowed in this step. The pixels of \mathbf{B} are defined as

$$B_{ij} = \begin{cases} 1 & \text{if } Z_{ij} > \theta \\ 0 & \text{else} \end{cases} \tag{9.8}$$

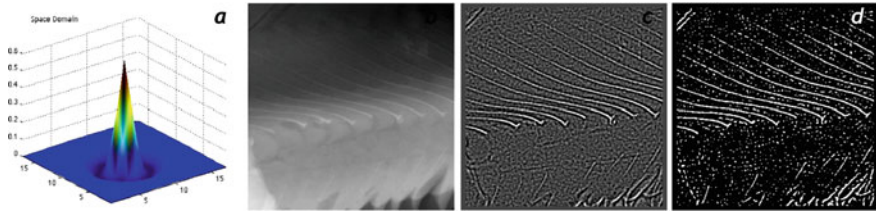


Fig. 9.35 Segmentation of potential fish bones: **a** Convolution mask **H** in space domain, **b** original X-ray image **X** of a salmon fillet, **c** filtered image **Z**, **d** potential fish bones image **P** after thresholding and removing objects deemed too small

Removal of Small Objects: All connected pixels in **B** containing fewer than A pixels are removed as shown in Fig. 9.35d. This image, called **P**, defines the potential fish bones.

Feature Extraction, Selection, And Classification

The segmented potential fish bones—contained in image **P**—are divided into small 10×10 pixels windows called *detection windows*. In a training phase, using a priori knowledge of the fish bones, the detection windows are manually labeled as one of two classes: *bones* and *no-bones*. The first class corresponds to those regions where the potential fish bones are indeed fish bones. Alternatively, the second class corresponds to false alarms. Intensity features of the enhanced X-ray image **Y** are extracted for both classes. We use enhanced image **Y**, instead of pre-processed image **X**, because after our experiments the detection performance was higher. Features extracted from each area of an X-ray image region are divided into four groups as shown in Sect. 9.3.2. In these experiments, 279 features are extracted from each detection window. Afterwards, the features are selected in order to decide on the relevant features for the two defined classes. In addition, a classifier is designed. The best results, after evaluation a 10-fold cross-validation was achieved by Sequential Forward Selection (as feature selection technique) and Support Vector Machine with RBF kernel (as classifier).

Experimental Results

First, the proposed method was tested with 20 representative salmon fillets obtained at a local fish market. The average size of these fillets was 15×10 cm². According to pre-processing and segmentation techniques explained above, several regions of interest were obtained where fish bones could be located. The area occupied by these regions of interest corresponds to approx. 12% of the salmon fillets as shown in Fig. 9.35. More results are presented in Fig. 9.36.

From the mentioned regions of interest 7697, detection windows of 10×10 pixels were obtained (available in series N0003 of GDXray+). Each window was labeled with '1' for class *bones* and '0' for *no-bones* (see file labels.txt in directory of N0003). From each window, 279 features were extracted. After the feature extrac-

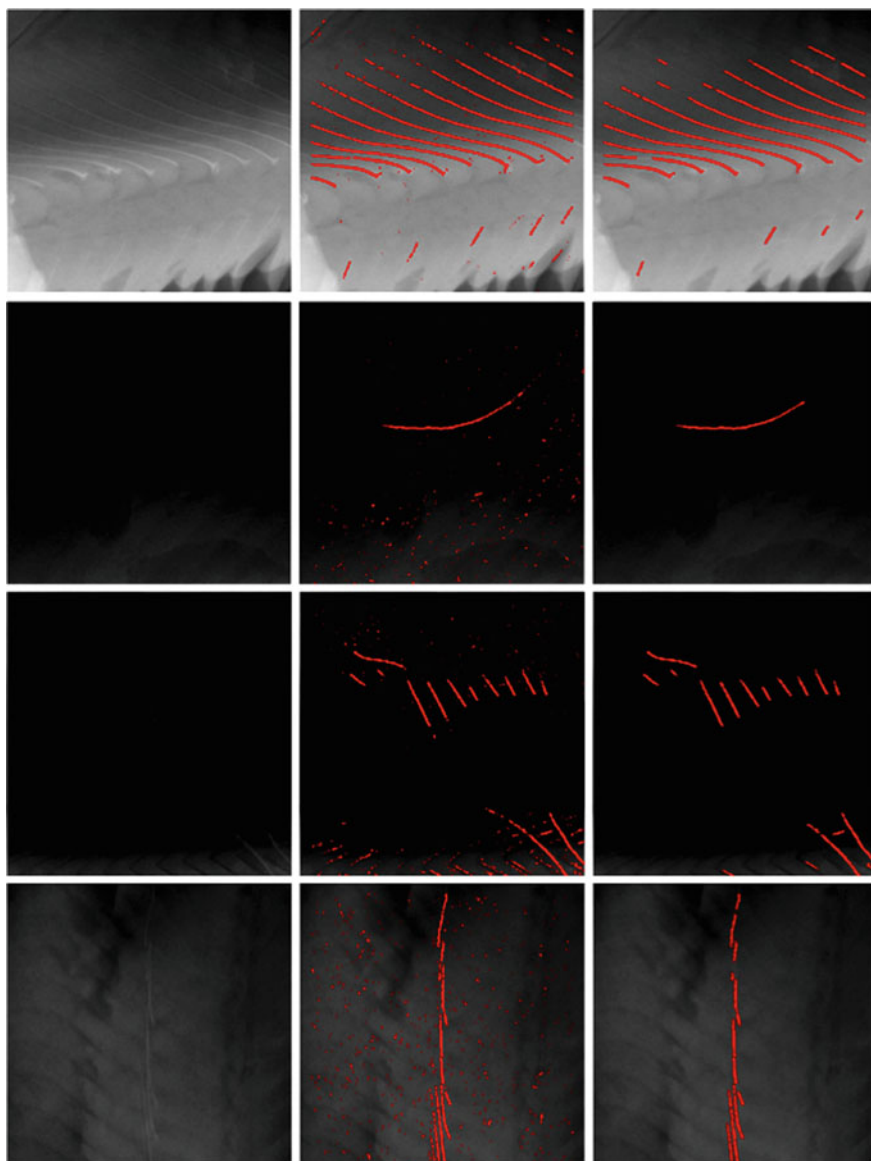


Fig. 9.36 Results obtained in four X-ray images. The columns correspond to enhanced images, classified fish bones and post processed fish bones. The first row corresponds to the example shown in Fig. 9.35

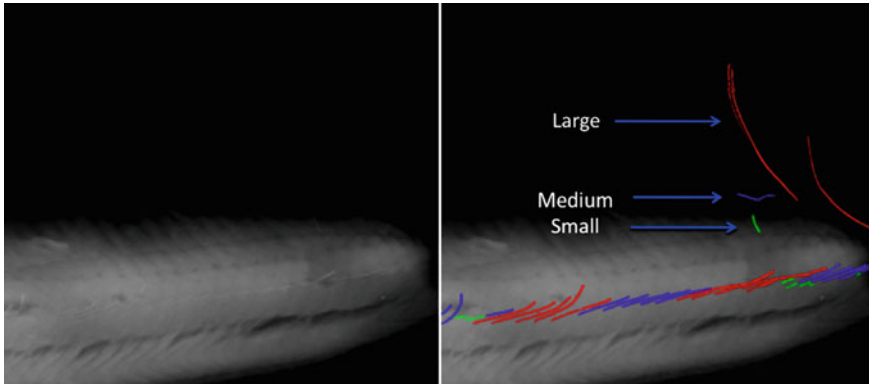


Fig. 9.37 Results obtained on 3878 samples using cross-validation with five folds. See results in Table 9.8

Table 9.8 Performance in the detection of fishbones

Fishbone	Sensibility (%)	1-Specificity (%)	Size
Large (red)	100	0	>0.64 mm × 12 mm
Medium (blue)	100	3	Between
Small (green)	93	6	<0.48 mm × 8.5 mm

tion, 75% of the samples from each class were randomly chosen to perform the feature selection. The best performance was achieved using Sequential Forward Selection. 24 features were selected. The features gave information about the spatial distribution of pixels, i.e., how coarse or fine the texture is. The selected features correspond mainly to statistical features (12) and filter banks (7), however, it is worth nothing that the two most discriminative features are LBP features (in this case LBP 48 and LBP 11). On the other hand, from the standard features there is only one feature (standard deviation of the intensity).

In order to investigate the sensibility (S_n) and 1-specificity ($1 - S_p$) of the fish bones depending on their largeness, three size groups were constructed: *large* for fish bones larger than 12mm, *small* for fish bones smaller than 8.5mm, and *medium* for fish bones between both sizes. In this experiment, 3878 fish bones were manually selected. The performance was calculated using a cross-validation with 5 folds. The results are summarized in Fig. 9.37 and Table 9.8. All medium and large fish bones were detected (with $1 - S_p = 0\%$ and 3% respectively), whereas 93% of small fish bones were correctly detected with $1 - S_p = 6\%$. This means that cross-validation yielded a detection performance of 100%, 98.5%, and 93.5% (computed using $(S_n + S_p)/2$) for large, medium, and small fish bones respectively.

Finally, in order to validate the proposed methodology, the last experiment was carried out using representative fish bones and representative trout fillets provided by a Chilean salmon industry. The size of the fish bones were between 14 and 47 mm

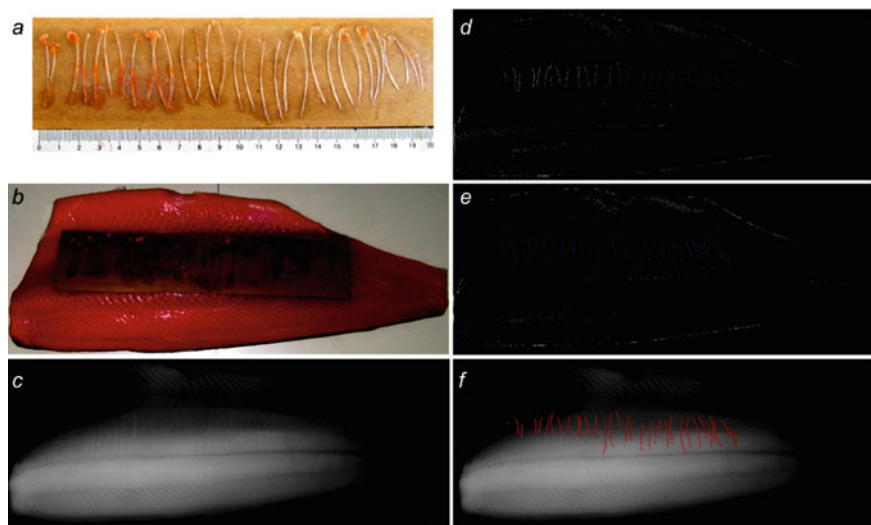


Fig. 9.38 Results obtained on a trout fillet using a fish bone strip with 33 fish bones: **a** strip, **b** strip over the fillet, **c** X-ray image, **d** segmentation, **e** classification, **f** post-processing. All fish bones were detected ($S_n = 1$), in this example there was no false alarm ($1 - S_p = 0$)

(larger than the small-size and mid-size groups considered above). The fish bones were arranged in strips that were superimposed onto trout fillets. Thus, the number of fish bones to be detected was a priori known. According to the absorption law, an X-ray image of a fillet with a fish bone inside, and an X-ray image with a fish bone laid on the fillet top are almost identical. Similar methodologies are used in industrial X-ray inspection of materials in order to simulate discontinuities [98]. The only difference could be that the position of a real fish bone (inside of a fillet) achieves a more realistic location related to the fish tissues, however, after our experience, the obtained images were found to be very similar. Fig. 9.38 shows the detection of one fish bone strip on a trout fillet. Using the same classifier trained in the last experiment, i.e., no new training was necessary, the proposed method was able to detect all fish bones with a 1% false positive rate. In this case, 15 X-ray images were tested, with 459 *bones* and 10413 *no-bones*.

Conclusion

The need for more information on the quality control of several fish types by means of quantitative methods can be satisfied using X-ray testing, a non-destructive technique that can be used to objectively measure intensity and geometric patterns in non-uniform surfaces. In addition the method can also determine other physical features such as image texture, morphological elements, and defects in order to automatically determine the quality of a fish fillet. The promising results outlined in this work show that a very high classification rate was achieved in the quality control of salmon and trout when using a large number of features combined with efficient feature selection

and classification. The key idea of the proposed method was to select, from a large universe of features, only those features that were relevant for the separation of the classes. Cross-validation yielded a detection performance of 100%, 98.5%, and 93.5% for large, medium, and small fish bones respectively. The proposed method was validated on trout with representative fish bones provided by a Chilean salmon industry yielding a performance of 99%. Although the method was validated with salmon and trout fillets only, we believe that the proposed approach opens new possibilities not only in the field of automated visual inspection of salmons and trout, but also in other similar fish.

9.5.3 An Example

In order to illustrate the methodology explained in the previous section, the reader can see Example 6.13, where the whole process is presented. In this example, 200 small X-ray images (100×100 pixels) of salmon filets, 100 with fish bones and 100 without fish bones are used. The images are available in series N0002 of GD \times Ray+.

9.6 Further Applications

There are many applications in which X-rays can be used as a NDT and E method. In this section, we mention only cargos and electronic circuits.

9.6.1 Cargo Inspection

With the ongoing development of international trade, cargo inspection becomes more and more important. X-ray testing has been used for the evaluation of the contents of cargo, trucks, containers, and passenger vehicles to detect the possible presence of many types of contraband. See an example in Fig. 9.39. Some approaches are presented in Table 9.9. There still is not much research on cargo inspection, and the complexity of this inspection task is very high. Nowadays, there are some approaches that use dual-energy, computed tomography, and deep learning. For this reason, X-ray systems are still only semi-automatic, and they require human supervision.

9.6.2 Electronic Circuits

In this industrial application of X-rays, the idea is to inspect circuit boards or integrated circuits in order to detect flaws in manufacturing, e.g., broken traces, missing

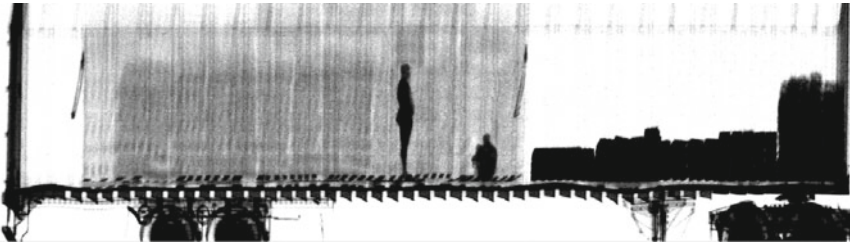


Fig. 9.39 X-ray image of a cargo. Collected by U.S. Customs and Border Protection a bureau of the United States Department of Homeland Security, via [Wikimedia Commons](#)

Table 9.9 State of art on cargo inspection

Authors	Year	Ref	\mathbb{X}_1^* energies	\mathbb{X}_2^* views	\mathbb{X}_3^* algorithms
			1 2 3	1 2 3	1 2 3
Duan	2008	[36]	⊗ □ □	⊗ ⊗ □	⊗ ⊗ □
Frosio	2011	[46]	⊗ □ □	⊗ □ □	⊗ ⊗ □
Kolkoori	2014	[71]	⊗ □ □	⊗ □ □	⊗ □ □
Kolokytha et al.	2018	[72]	⊗ □ □	⊗ □ □	⊗ ⊗ □
Jaccard et al.	2016	[63]	⊗ □ □	⊗ □ □	⊗ ⊗ □
Jaccard et al.	2017	[64]	⊗ □ □	⊗ □ □	⊗ ⊗ □
Lee et al.	2018	[78]	□ ⊗ □	⊗ □ □	⊗ ⊗ □
Rogers et al.	2017	[144]	□ ⊗ □	⊗ □ □	□ □ ⊗
Zhu	2008	[193]	⊗ □ □	⊗ □ □	⊗ ⊗ □
Zhu	2010	[192]	⊗ □ □	⊗ □ □	⊗ ⊗ □
		*1	Mono	Mono	Simple
		2	Dual	Multi	Medium
		3	Multi	CT	Complex

□ not used , ⊗ used

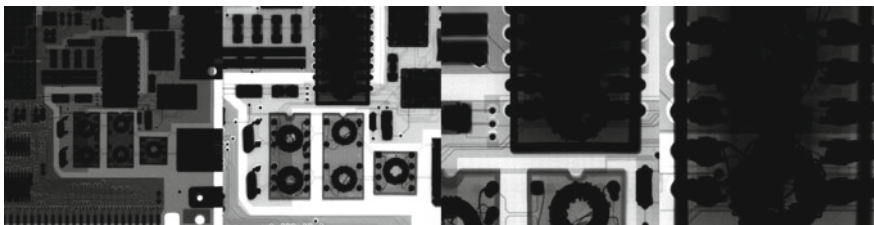


Fig. 9.40 X-ray image of a printed circuit board. By SecretDisc (Own work) via [Wikimedia Commons](#)

components, cracks, dilapidations, etc. An example is shown in Fig. 9.40. Some approaches are presented in Table 9.10. In this area, automated systems are very effective, and the inspection task is very fast and obtains a high performance.

Table 9.10 State of art on electronic circuit boards

Authors	Year	Ref	\mathbb{X}_1^* energies 1 2 3	\mathbb{X}_2^* views 1 2 3	\mathbb{X}_3^* algorithms 1 2 3
Adato et al.	2016	[3]	☒ ☒ ☒	☒ □ □	☒ ☒ □
Alam et al.	2017	[10]	☒ □ □	□ □ ☒	☒ ☒ □
Ghosh et al.	2018	[48]	☒ □ □	☒ □ □	☒ ☒ □
Goto et al.	2019	[51]	☒ □ □	☒ □ □	□ □ ☒
Favata and Shahbazmo-hamadi	2018	[38]	☒ □ □	☒ □ ☒	☒ ☒ □
Lin et al.	2017	[85]	☒ □ □	☒ □ □	☒ ☒ □
Mahmood et al.	2015	[93]	☒ □ □	☒ □ □	☒ ☒ □
Uehara et al.	2013	[166]	☒ □ □	☒ □ □	☒ ☒ □
Wang et al.	2014	[175]	☒ □ □	☒ □ □	☒ ☒ □
Wu et al.	2014	[177]	☒ □ □	☒ □ □	☒ ☒ □
Zakaria et al.	2020	[184]	☒ □ □	☒ □ □	☒ ☒ □
		*1	Mono	Mono	Simple
		2	Dual	Multi	Medium
		3	Multi	CT	Complex

□ not used, ☒ used

9.7 Summary

In this chapter, relevant applications on X-ray testing were described. We covered X-ray testing in:

- **Castings:** To ensure the safety of the construction of automative parts, it is necessary to check every part thoroughly using X-ray testing. We presented the state of the art, a defect detection approach based on a tracking principle, and a Python implementation of a classifier that is able to detect casting defects in single X-ray images.
- **Welds:** In welding processes, a mandatory inspection using X-ray testing is required in order to detect defects like porosity, inclusion, lack of fusion, lack of penetration, and cracks. We presented the state of the art, a defect detection approach based on sliding windows, and a Python implementation of a classifier that is able to detect defects using sliding windows methodology in single X-ray images.
- **Baggage:** In baggage screening, every piece of luggage must be inspected using X-ray testing in order to detect dangerous objects. We presented the state of the art, a recognition approach based on multiple-view analysis, and a Matlab implementation of tracking principle that is able to detect objects in the sequence X-ray images of a pen case.

- **Natural products:** We presented some applications of X-ray testing in natural products, such as inspection of fruit, identification of infections, and detection of fish bones. We reviewed the state of the art, a fish bones detection approach based on sliding windows, and a Python implementation of a classifier that is able to detect fish bones in cropped images with and without fish bones.
- **Others:** There are several industrial applications that use X-ray testing. We mentioned only cargos and electronic circuits giving some references of the state of art.

References

1. Abidi, B.R., Zheng, Y., Gribok, A.V., Abidi, M.A.: Improving weapon detection in single energy X-ray images through pseudocoloring. *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.* **36**(6), 784–796 (2006)
2. Abusaeeda, O., Evans, J., Downes, D., Chan, J.: View synthesis of KDEX imagery for 3D security X-ray imaging. In: *Proceedings of the 4th International Conference on Imaging for Crime Detection and Prevention (ICDP-2011)* (2011)
3. Adato, R., Uyar, A., Zangeneh, M., Zhou, B., Joshi, A., Goldberg, B., Unlu, M.S.: Rapid mapping of digital integrated circuit logic gates via multi-spectral backside imaging. *arXiv preprint arXiv:1605.09306* (2016)
4. Ajmi, C., El Ferchichi, S., Laabidi, K.: New procedure for weld defect detection based-gabor filter. In: *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, pp. 11–16. *IEEE* (2018)
5. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. *arXiv preprint arXiv:1805.06725* (2018)
6. Akcay, S., Breckon, T.: Towards automatic threat detection: a survey of advances of deep learning within X-ray security imaging. *arXiv preprint arXiv:2001.01293* (2020)
7. Akcay, S., Breckon, T.P.: An evaluation of region based object detection strategies within X-ray baggage security imagery. In: *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 1337–1341. *IEEE* (2017)
8. Akçay, S., Kundegorski, M.E., Devereux, M., Breckon, T.P.: Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery. In: *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1057–1061. *IEEE* (2016)
9. Akcay, S., Kundegorski, M.E., Willcocks, C.G., Breckon, T.P.: Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery. *IEEE Trans. Inf. Forens. Secur.* **13**(9), 2203–2215 (2018)
10. Alam, M., Shen, H., Asadizanjani, N., Tehranipoor, M., Forte, D.: Impact of x-ray tomography on the reliability of integrated circuits. *IEEE Trans. Device Mater. Reliability* **17**(1), 59–68 (2017)
11. Anand, R., Kumar, P., et al.: Flaw detection in radiographic weldment images using morphological watershed segmentation technique. *NDT & E Int.* **42**(1), 2–8 (2009)
12. Aydin, I., Karakose, M., Erhan, A.: A new approach for baggage inspection by using deep convolutional neural networks. In: *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pp. 1–6. *IEEE* (2018)
13. Bandara, A., Kan, K., Morii, H., Koike, A., Aoki, T.: X-ray computed tomography to investigate industrial cast al-alloys. *Product. Eng.* **14**(2), 147–156 (2020)
14. Baniukiewicz, P.: Automated defect recognition and identification in digital radiography. *J. Nondestruct. Eval.* **33**(3), 327–334 (2014)

15. Baştan, M.: Multi-view object detection in dual-energy X-ray images. *Mach. Vis. Appl.* **26**(7–8), 1045–1060 (2015)
16. Baştan, M., Yousefi, M.R., Breuel, T.M.: Visual words on baggage X-ray images. In: *Computer Analysis of Images and Patterns*, pp. 360–368. Springer, Berlin (2011)
17. von Bastian, C., Schwaninger, A., Michel, S.: Do Multi-view X-ray Systems Improve X-ray Image Interpretation in Airport Security Screening?, vol. 52. GRIN Verlag (2010)
18. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: speeded up robust features. In: 9th European Conference on Computer Vision (ECCV2006). Graz Austria (2006)
19. Bej, G., Akuli, A., Pal, A., Dey, T., Chaudhuri, A., Alam, S., Khandai, R., Bhattacharyya, N.: X-ray imaging and general regression neural network (GRNN) for estimation of silk content in cocoons. In: *Proceedings of the 2nd International Conference on Perception and Machine Intelligence*, pp. 71–76. ACM (2015)
20. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1798–1828 (2013)
21. Bentley, J.: Multidimensional binary search trees used for associative searching. *Commun. ACM* **18**(9), 509–517 (1975)
22. Boerner, H., Strecker, H.: Automated X-ray inspection of aluminum casting. *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(1), 79–91 (1988)
23. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**(6), 679–698 (1986)
24. Carrasco, M., Mery, D.: Automatic multiple view inspection using geometrical tracking and feature analysis in aluminum wheels. *Mach. Vis. Appl.* **22**(1), 157–170 (2011)
25. Chan, J., Evans, P., Wang, X.: Enhanced color coding scheme for kinetic depth effect X-ray (KDEX) imaging. In: 2010 IEEE International Carnahan Conference on Security Technology (ICCST), pp. 155–160 (2010)
26. Chen, Z., Zheng, Y., Abidi, B.R., Page, D.L., Abidi, M.A.: A combinational approach to the fusion, denoising and enhancement of dual-energy X-ray luggage images. In: *Workshop of IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2005)* (2005)
27. Cogranne, R., Reira, F.: Statistical detection of defects in radiographic images using an adaptive parametric model. *Signal Process.* **96**, 173–189 (2014)
28. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002)
29. van Dael, M., Lebotsa, S., Herremans, E., Verboven, P., Sijbers, J., Opara, U., Cronje, P., Nicolaï, B.: A segmentation and classification algorithm for online detection of internal disorders in citrus using X-ray radiographs. *Postharvest Biol. Technol.* **112**, 205–214 (2016)
30. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Conference on Computer Vision and Pattern Recognition (CVPR2005)*, vol. 1, pp. 886–893 (2005)
31. Ding, J., Li, Y., Xu, X., Wang, L.: X-ray image segmentation by attribute relational graph matching. In: 8th IEEE International Conference on Signal Processing, vol. 2 (2006)
32. Dollár, P.: Piotr’s Image and Video Matlab Toolbox (PMT). <http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html>
33. Douarre, C., Schielein, R., Frindel, C., Gerth, S., Rousseau, D.: Deep learning based root-soil segmentation from X-ray tomography images. *bioRxiv* p. 071662 (2016)
34. Du, W., Shen, H., Fu, J., Zhang, G., He, Q.: Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning. *NDT & E Int.* **107**, 102,144 (2019)
35. Du, Z., Hu, Y., Buttar, N.A., Mahmood, A.: X-ray computed tomography for quality inspection of agricultural products: a review. *Food Sci. Nutr.* **7**(10), 3146 (2019)
36. Duan, X., Cheng, J., Zhang, L., Xing, Y., Chen, Z., Zhao, Z.: X-ray cargo container inspection system with few-view projection imaging. *Nuclear Instrum. Methods Phys. Res. A* **598**, 439–444 (2009)
37. Faugeras, O.: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge (1993)

38. Favata, J., Shahbazmohamadi, S.: Realistic non-destructive testing of integrated circuit bond wiring using 3-d x-ray tomography, reverse engineering, and finite element analysis. *Microelectron. Reliability* **83**, 91–100 (2018)
39. Feng, X., Zhang, H., Yu, P.: X-ray fluorescence application in food, feed, and agricultural science: a critical review. *Critic. Rev. Food Sci. Nutr.* 1–11 (2020)
40. Ferguson, M., Ak, R., Lee, Y.T.T., Law, K.H.: Automatic localization of casting defects with convolutional neural networks. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 1726–1735. IEEE (2017)
41. Ferguson, M.K., Ronay, A., Lee, Y.T.T., Law, K.H.: Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart Sustain. Manufact. Syst.* **2** (2018)
42. Filbert, D., Klatt, R., Heinrich, W., Purschke, M.: Computer aided inspection of castings. In: IEEE-IAS Annual Meeting, pp. 1087–1095. Atlanta, USA (1987)
43. Flitton, G., Breckon, T.P., Megherbi, N.: A comparison of 3d interest point descriptors with application to airport baggage object detection in complex ct imagery. *Pattern Recognit.* **46**(9), 2420–2436 (2013)
44. Flitton, G., Mouton, A., Breckon, T.P.: Object classification in 3d baggage security computed tomography imagery using visual codebooks. *Pattern Recognit.* **48**(8), 2489–2499 (2015)
45. Franzel, T., Schmidt, U., Roth, S.: Object detection in multi-view X-Ray images. *Pattern Recognit.* 144–154 (2012)
46. Frosio, I., Borghese, N., Lissandrello, F., Venturino, G., Rotondo, G.: Optimized acquisition geometry for X-ray inspection. In: 2011 IEEE Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–6 (2011)
47. Gao, W., Hu, Y.H.: Real-time X-ray radiography for defect detection in submerged arc welding and segmentation using sparse signal representation. *Insight-Non-Destruct. Test. Condit. Monitor.* **56**(6), 299–307 (2014)
48. Ghosh, P., Forte, D., Woodard, D.L., Chakraborty, R.S.: Automated detection of pin defects on counterfeit microelectronics. In: ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis, p. 57. ASM International (2018)
49. Gobi, A.F.: Towards generalized benthic species recognition and quantification using computer vision. In: 4th Pacific-Rim Symposium on Image and Video Technology (PSIVT2010), Singapore, Nov. 14–17, 2010, pp. 94–100 (2010)
50. Gonzalez, R., Woods, R.: *Digital Image Processing*, 3rd edn. Prentice Hall, Pearson (2008)
51. Goto, K., Kato, K., Nakatsuka, S., Saito, T., Aizawa, H.: Anomaly detection of solder joint on print circuit board by using adversarial autoencoder. In: Fourteenth International Conference on Quality Control by Artificial Vision, vol. 11172, p. 111720T. International Society for Optics and Photonics (2019)
52. Guelpa, A., du Plessis, A., Kidd, M., Manley, M.: Non-destructive estimation of maize (*zea mays* l.) kernel hardness by means of an X-ray micro-computed tomography (μ ct) density calibration. *Food Bioprocess Technol.* **8**(7), 1419–1429 (2015)
53. Haff, R., Slaughter, D.: Real-time X-ray inspection of wheat for infestation by the granary weevil, *sitophilus granarius* (l.). *Trans. Am. Soc. Agric. Eng.* **47**, 531–537 (2004)
54. Haff, R., Toyofuku, N.: X-ray detection of defects and contaminants in the food industry. *Sens. Instrum. Food Quality Safety* **2**(4), 262–273 (2008). <https://doi.org/10.1007/s11694-008-9059-8>
55. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conferences, pp. 147–152 (1988)
56. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2003)
57. Hassan, J., Awan, A.M., Jalil, A.: Welding defect detection and classification using geometric features. In: 2012 10th International Conference on Frontiers of Information Technology, pp. 139–144. IEEE (2012)
58. Hecker, H.: Ein neues Verfahren zur robusten Röntgenbildauswertung in der automatischen Gußteilprüfung. Ph.D. thesis, vom Fachbereich Elektrotechnik, Technische Universität Berlin (1995)

59. Heinrich, W.: Automated inspection of castings using X-ray testing. Ph.D. thesis, Institute for Measurement and Automation, Faculty of Electrical Engineering, Technical University of Berlin (1988). (in German)
60. Heitz, G., Chechik, G.: Object separation in X-ray image sets. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2010), pp. 2093–2100 (2010)
61. Hou, W., Wei, Y., Guo, J., Jin, Y., et al.: Automatic detection of welding defects using deep neural network. In: Journal of Physics: Conference Series, vol. 933, p. 012006. IOP Publishing (2018)
62. Hou, W., Wei, Y., Jin, Y., Zhu, C.: Deep features based on a dcnn model for classifying imbalanced weld flaw types. *Measurement* **131**, 482–489 (2019)
63. Jaccard, N., Rogers, T.W., Morton, E.J., Griffin, L.D.: Tackling the X-ray cargo inspection challenge using machine learning. In: Anomaly Detection and Imaging with X-Rays (ADIX), vol. 9847, p. 98470N. International Society for Optics and Photonics (2016)
64. Jaccard, N., Rogers, T.W., Morton, E.J., Griffin, L.D.: Detection of concealed cars in complex cargo X-ray imagery using deep learning. *J. X-ray Sci. Technol.* **25**(3), 323–339 (2017)
65. Jiang, J., Chang, H., Wu, K., Ouyang, C., Yang, M., Yang, E., Chen, T., Lin, T.: An adaptive image segmentation algorithm for X-ray quarantine inspection of selected fruits. *Comput. Electron. Agricul.* **60**, 190–200 (2008)
66. Jin, C., Kong, X., Chang, J., Cheng, H., Liu, X.: Internal crack detection of castings: a study based on relief algorithm and adaboost-svm. *Int. J. Adv. Manufact. Technol.* 1–10 (2020)
67. Kaftandjian, V., Dupuis, O., Babot, D., Zhu, Y.M.: Uncertainty modelling using dempstershafer theory for improving detection of weld defects. *Pattern Recognit. Lett.* **24**(1), 547–564 (2003)
68. Kamalakannan, A., Rajamanickam, G.: Spatial smoothing based segmentation method for internal defect detection in X-ray images of casting components. In: 2017 Trends in Industrial Measurement and Automation (TIMA), pp. 1–6. IEEE (2017)
69. Karunakaran, C., Jayas, D., White, N.: Identification of wheat kernels damaged by the red flour beetle using X-ray images. *Biosyst. Eng.* **87**(3), 267–274 (2004)
70. Kelkar, S., Boushey, C.J., Okos, M.: A method to determine the density of foods using X-ray imaging. *J. Food Eng.* (2015)
71. Kolkoori, S., Wrobel, N., Deresch, A., Redmer, B., Ewert, U.: Dual high-energy X-ray digital radiography for material discrimination in cargo containers. In: 11th European Conference on Non-Destructive Testing (ECNDT 2014), October 6–10, 2014. Prague, Czech Republic (2014)
72. Kolokytha, S., Flisch, A., Lüthi, T., Plamondon, M., Visser, W., Schwaninger, A., Hardmeier, D., Costin, M., Vienne, C., Sukowski, F.: Creating a reference database of cargo inspection X-ray images using high energy radiographs of cargo mock-ups. *Multimedia Tools Appl.* **77**(8), 9379–9391 (2018)
73. Kotwaliwale, N., Singh, K., Kalne, A., Jha, S.N., Seth, N., Kar, A.: X-ray imaging methods for internal quality evaluation of agricultural produce. *J. food Sci. Technol.* **51**(1), 1–15 (2014)
74. Kumar, J., Anand, R., Srivastava, S.: Flaws classification using ann for radiographic weld images. In: 2014 International Conference on Signal Processing and Integrated Networks (SPIN), pp. 145–150 (2014)
75. Kumar, J., Anand, R., Srivastava, S.: Multi - class welding flaws classification using texture feature for radiographic images. In: 2014 International Conference on Advances in Electrical Engineering (ICAEE), pp. 1–4 (2014)
76. Kwon, J., Lee, J., Kim, W.: Real-time detection of foreign objects using X-ray imaging for dry food manufacturing line. In: Proceedings of IEEE International Symposium on Consumer Electronics (ISCE 2008), pp. 1–4 (2008)
77. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
78. Lee, D., Lee, J., Min, J., Lee, B., Lee, B., Oh, K., Kim, J., Cho, S.: Efficient material decomposition method for dual-energy X-ray cargo inspection system. *Nuclear Instrum. Methods Phys. Res. Sect. A: Accelerat., Spectrom., Detect. Assoc. Equip.* **884**, 105–112 (2018)

79. Li, J., Oberdorfer, B., Schumacher, P.: Determining casting defects in thixomolding mg casting part by computed tomography. In: *Shape Casting*, pp. 99–103. Springer, Berlin (2019)
80. Li, W., Li, K., Huang, Y., Deng, X.: A new trend peak algorithm with X-ray image for wheel hubs detection and recognition. In: *Computational Intelligence and Intelligent Systems*, pp. 23–31. Springer, Berlin (2015)
81. Li, X., Tso, S.K., Guan, X.P., Huang, Q.: Improving automatic detection of defects in castings by applying wavelet technique. *IEEE Trans. Indust. Electron.* **53**(6), 1927–1934 (2006)
82. Liao, T.: Classification of welding flaw types with fuzzy expert systems. *Fuzzy Sets Syst.* **108**, 145–158 (2003)
83. Liao, T.: Classification of weld flaws with imbalanced class data. *Expert Systems with Applications* **35**(3), 1041–1052 (2008)
84. Liao, T.W.: Improving the accuracy of computer-aided radiographic weld inspection by feature selection. *NDT&E Int.* **42**, 229–239 (2009)
85. Lin, C.S., Chan, B.E., Huang, Y.C., Chen, H.T., Lin, Y.C.: X-ray imaging inspection system for blind holes in the intermediate layer of printed circuit boards with neural network identification. *J. Test. Eval.* **45**(3), 1005–1015 (2017)
86. Lin, J., Yao, Y., Ma, L., Wang, Y.: Detection of a casting defect tracked by deep convolution neural network. *Int. J. Adv. Manufact. Technol.* **97**(1–4), 573–581 (2018)
87. Lindgren, E.: Detection, 3-D positioning, and sizing of small pore defects using digital radiography and tracking. *EURASIP J. Adv. Signal Process.* **2014**(1), 1–17 (2014)
88. Liu, B., Zhang, X., Gao, Z., Chen, L.: Weld defect images classification with vgg16-based neural network. In: *International Forum on Digital TV and Wireless Multimedia Communications*, pp. 215–223. Springer (2017)
89. Liu, D., Wang, Z.: A united classification system of X-ray image based on fuzzy rule and neural networks. In: *3rd International Conference on Intelligent System and Knowledge Engineering*, 2008. ISKE 2008, vol. 1, pp. 717–722 (2008)
90. Liu, J., Leng, X., Liu, Y.: Deep convolutional neural network based object detector for x-ray baggage security imagery. In: *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1757–1761. IEEE (2019)
91. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
92. Lu, Q., Conners, R.: Using image processing methods to improve the explosive detection accuracy. *IEEE Trans. Appl. Rev., Part C: Syst., Man, Cybern.* **36**(6), 750–760 (2006)
93. Mahmood, K., Carmona, P.L., Shahbazmohamadi, S., Pla, F., Javidi, B.: Real-time automated counterfeit integrated circuit detection using x-ray microscopy. *Appl. Opt.* **54**(13), D25–D32 (2015)
94. Mansoor, M., Rajashankari, R.: Detection of concealed weapons in X-ray images using fuzzy K-NN. *Int. J. Comput. Sci., Eng. Inf. Technol.* **2**(2) (2012)
95. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **22**(10), 761–767 (2004)
96. Mathanker, S., Weckler, P., Bowser, T., Wang, N., Maness, N.: Adaboost classifiers for pecan defect classification. *Comput. Electron. Agricul.* **77**(1), 60–68 (2011)
97. Mery, D.: BALU: A toolbox Matlab for computer vision, pattern recognition and image processing. <http://dmery.ing.puc.cl/index.php/balu>
98. Mery, D.: Flaw simulation in castings inspection by radioscopy. *Insight* **43**(10), 664–668 (2001)
99. Mery, D.: Crossing line profile: a new approach to detecting defects in aluminium castings. In: *Proceedings of the Scandinavian Conference on Image Analysis (SCIA 2003)*. Lecture Notes in Computer Science, vol. 2749, pp. 725–732 (2003)
100. Mery, D.: Automated radioscopic testing of aluminum die castings. *Mater. Eval.* **64**(2), 135–143 (2006)
101. Mery, D.: Automated detection in complex objects using a tracking algorithm in multiple X-ray views. In: *Proceedings of the 8th IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS 2011)*, in Conjunction with CVPR 2011, Colorado Springs, pp. 41–48 (2011)

102. Mery, D.: Automated detection of welding defects without segmentation. *Mater. Eval.* **69**(6), 657–663 (2011)
103. Mery, D.: Inspection of complex objects using multiple-X-ray views. *IEEE/ASME Trans. Mechatron.* **20**(1), 338–347 (2015)
104. Mery, D.: Aluminum casting inspection using deep learning: a method based on convolutional neural networks. *J. Nondestruct. Eval.* **39**(1), 12 (2020)
105. Mery, D., Arteta, C.: Automatic defect recognition in X-ray testing using computer vision. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1026–1035. IEEE (2017)
106. Mery, D., Berti, M.A.: Automatic detection of welding defects using texture features. *Insight-Non-Destruct. Test. Condit. Monitor.* **45**(10), 676–681 (2003)
107. Mery, D., Filbert, D.: Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence. *IEEE Trans. Robot. Autom.* **18**(6), 890–901 (2002)
108. Mery, D., Filbert, D.: Classification of potential defects in automated inspection of aluminium castings using statistical pattern recognition. In: 8th European Conference on Non-Destructive Testing (ECNDT 2002), pp. 1–10. Barcelona (2002)
109. Mery, D., Filbert, D., Jaeger, T.: Image processing for fault detection in aluminum castings. In: MacKenzie, D., Totten, G. (eds.) *Anal. Charact. Alumin. Alloys*. Marcel Dekker, New York (2003)
110. Mery, D., Filbert, D., Parspour, N.: Improvement in automated aluminum casting inspection by finding correspondence of potential flaws in multiple radioscopic images. In: Proceedings of the 15th World Conference on Non-Destructive Testing (WCNDT–2000). Rome (2000)
111. Mery, D., Lillo, I., Riffo, V., Soto, A., Cipriano, A., Aguilera, J.: Automated fish bone detection using X-ray testing. *J. Food Eng.* **2011**(105), 485–492 (2011)
112. Mery, D., Riffo, V., Mondragon, G., Zuccar, I.: Detection of regular objects in baggages using multiple X-ray views. *Insight* **55**(1), 16–21 (2013)
113. Mery, D., Riffo, V., Zscherpel, U., Mondragón, G., Lillo, I., Zuccar, I., Lobel, H., Carrasco, M.: GDxray: the database of X-ray images for nondestructive testing. *J. Nondestruct. Eval.* **34**(4), 1–12 (2015)
114. Mery, D., Riffo, V., Zuccar, I., Pieringer, C.: Automated X-ray object recognition using an efficient search algorithm in multiple views. In: Proceedings of the 9th IEEE CVPR Workshop on Perception Beyond the Visible Spectrum, Portland (2013)
115. Mery, D., Svec, E., Arias, M., Riffo, V., Saavedra, J.M., Banerjee, S.: Modern computer vision techniques for X-ray testing in baggage inspection. *IEEE Trans. Syst., Man, Cybern.: Syst.* **47**(4), 682–692 (2016)
116. Miao, C., Xie, L., Wan, F., Su, C., Liu, H., Jiao, J., Ye, Q.: Sixray: a large-scale security inspection X-ray benchmark for prohibited item discovery in overlapping images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2119–2128 (2019)
117. Michel, S., Koller, S., de Ruyter, J., Moerland, R., Hogervorst, M., Schwaninger, A.: Computer-based training increases efficiency in X-Ray image interpretation by aviation security screeners. In: 2007 41st Annual IEEE International Carnahan Conference on Security Technology, pp. 201–206 (2007)
118. Montabone, S., Soto, A.: Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image Vis. Comput.* **28**(3), 391–402 (2010)
119. Mouton, A., Breckon, T.P.: Materials-based 3d segmentation of unknown objects from dual-energy computed tomography imagery in baggage security screening. *Pattern Recognit.* **48**(6), 1961–1978 (2015)
120. Mu, W., Gao, J., Jiang, H., Wang, Z., Chen, F., Dang, C.: Automatic classification approach to weld defects based on pca and svm. *Insight-Non-Destruct. Test. Condit. Monitor.* **55**(10), 535–539 (2013)
121. Muniategui, A., del Barrio, J.A., Vinuesa, X.A., Masenlle, M., de la Yedra, A.G., Moreno, R.: One dimensional fourier transform on deep learning for industrial welding quality control. In: International Work-Conference on Artificial Neural Networks, pp. 174–185. Springer (2019)

122. Muravyov, S., Pogadaeva, E.Y.: Computer-aided recognition of defects in welded joints during visual inspections based on geometric attributes. *Russian J. Nondestruct. Test.* **56**, 259–267 (2020)
123. Murphy, E.: A rising war on terrorists. *Spectrum, IEEE* **26**(11), 33–36 (1989)
124. Murray, N., Riordan, K.: Evaluation of automatic explosive detection systems. In: 29th Annual 1995 International Carnahan Conference on Security Technology, 1995. Proceedings. Institute of Electrical and Electronics Engineers, pp. 175–179 (1995). <https://doi.org/10.1109/CCST.1995.524908>
125. Neethirajan, S., Karunakaran, C., Symons, S., Jayas, D.: Classification of vitreousness in durum wheat using soft X-rays and transmitted light images. *Comput. Electron. Agricul.* **53**(1), 71–78 (2006)
126. Nercessian, S., Panetta, K., Agaian, S.: Automatic detection of potential threat objects in X-ray luggage scan images. In: 2008 IEEE Conference on Technologies for Homeland Security, pp. 504–509 (2008). 10.1109/THS.2008.4534504
127. Nercessian, S., Panetta, K., Agaian, S.: Automatic detection of potential threat objects in X-ray luggage scan images. In: 2008 IEEE Conference on Technologies for Homeland Security, pp. 504–509 (2008)
128. Nielsen, M.S., Christensen, L.B., Feidenhans, R.: Frozen and defrosted fruit revealed with X-ray dark-field radiography. *Food Control* **39**, 222–226 (2014)
129. Noble, A., Gupta, R., Mundy, J., Schmitz, A., Hartley, R.: High precision X-ray stereo for automated 3D CAD-based inspection. *IEEE Trans. Robot. Autom.* **14**(2), 292–302 (1998)
130. Nugraha, B., Verboven, P., Janssen, S., Wang, Z., Nicolai, B.M.: Non-destructive porosity mapping of fruit and vegetables using X-ray ct. *Postharvest Biol. Technol.* **150**, 80–88 (2019)
131. Oertel, C., Bock, P.: Identification of objects-of-interest in X-Ray images. In: Applied Imagery and Pattern Recognition Workshop, 2006. AIPR 2006. 35th IEEE, p. 17 (2006)
132. Ogawa, Y., Kondo, N., Shibusawa, S.: Inside quality evaluation of fruit by X-ray image. In: 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. vol. 2, pp. 1360–1365 (2003)
133. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
134. Orina, I., Manley, M., Kucheryavskiy, S., Williams, P.J.: Application of image texture analysis for evaluation of X-ray images of fungal-infected maize kernels. *Food Anal. Methods* **11**(10), 2799–2815 (2018)
135. Pan, H., Pang, Z., Wang, Y., Wang, Y., Chen, L.: A new image recognition and classification method combining transfer learning algorithm and mobilenet model for welding defects. *IEEE Access* (2020)
136. Pieringer, C., Mery, D.: Flaw detection in aluminium die castings using simultaneous combination of multiple views. *Insight* **52**(10), 548–552 (2010)
137. Pizarro, L., Mery, D., Delpiano, R., Carrasco, M.: Robust automated multiple view inspection. *Pattern Anal. Appl.* **11**(1), 21–32 (2008)
138. Ramírez, F., Allende, H.: Detection of flaws in aluminium castings: a comparative study between generative and discriminant approaches. *Insight-Non-Destruct. Test. Condit. Monitor.* **55**(7), 366–371 (2013)
139. Ren, J., Ren, R., Green, M., Huang, X.: Defect detection from X-ray images using a three-stage deep learning algorithm. In: 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), pp. 1–4. IEEE (2019)
140. Riffo, V., Flores, S., Mery, D.: Threat objects detection in X-ray images using an active vision approach. *J. Nondestruct. Eval.* **36**(3), 44 (2017)
141. Riffo, V., Godoy, I., Mery, D.: Handgun detection in single-spectrum multiple X-ray views based on 3d object recognition. *J. Nondestruct. Eval.* **38**(3), 66 (2019)
142. Riffo, V., Mery, D.: Active X-ray testing of complex objects. *Insight* **54**(1), 28–35 (2012)
143. Riffo, V., Mery, D.: Automated detection of threat objects using adapted implicit shape model. *IEEE Trans. Syst., Man, Cybern.: Syst.* **46**(4), 472–482 (2016)

144. Rogers, T.W., Jaccard, N., Griffin, L.D.: A deep learning framework for the automated inspection of complex dual-energy X-ray cargo imagery. In: *Anomaly Detection and Imaging with X-Rays (ADIX) II*, vol. 10187, p. 101870L. International Society for Optics and Photonics (2017)
145. Saavedra, D., Banerjee, S., Mery, D.: Detection of threat objects in baggage inspection with X-ray images using deep learning. *Neural Comput. Appl.* pp. 1–17. Springer (2020)
146. Sangwan, D., Jain, D.K.: An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery. *Pattern Recognit. Lett.* (2019)
147. Schmidt-Hackenberg, L., Yousefi, M.R., Breuel, T.M.: Visual cortex inspired features for object detection in X-ray images. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2573–2576. IEEE (2012)
148. Schoeman, L., Williams, P., du Plessis, A., Manley, M.: X-ray micro-computed tomography (μ ct) for non-destructive characterisation of food microstructure. *Trends Food Sci. Technol.* **47**, 10–24 (2016)
149. Shao, J., Du, D., Chang, B., Shi, H.: Automatic weld defect detection based on potential defect tracking in real-time radiographic image sequence. *NDT & E Int.* **46**, 14–21 (2012)
150. Shi, D.H., Gang, T., Yang, S.Y., Yuan, Y.: Research on segmentation and distribution features of small defects in precision weldments with complex structure. *NDT & E Int.* **40**, 397–404 (2007)
151. Sigman, J.B., Spell, G.P., Liang, K.J., Carin, L.: Background adaptive faster R-CNN for semi-supervised convolutional object detection of threats in X-ray images. In: *Anomaly Detection and Imaging with X-Rays (ADIX) V*, vol. 11404, p. 1140404. International Society for Optics and Photonics (2020)
152. da Silva, R., Mery, D.: State-of-the-art of weld seam inspection using X-ray testing: Part I - image processing. *Mater. Eval.* **65**(6), 643–647 (2007)
153. da Silva, R., Mery, D.: State-of-the-art of weld seam inspection using X-ray testing: Part II - pattern recognition. *Mater. Eval.* **65**(9), 833–838 (2007)
154. da Silva, R.R., Calóba, L.P., Siqueira, M.H., Rebello, J.M.: Pattern recognition of weld defects detected by radiographic test. *Ndt & E Int.* **37**(6), 461–470 (2004)
155. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR [arXiv:abs/1409.1556](https://arxiv.org/abs/1409.1556) (2014)
156. Singh, M., Singh, S.: Optimizing image enhancement for screening luggage at airports. In: *Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, 2005. CIHSPS 2005*, pp. 131–136 (2005). <https://doi.org/10.1109/CIHSPS.2005.1500627>
157. Singh, S., Singh, M.: Explosives detection systems (eds) for aviation security. *Signal Process.* **83**(1), 31–55 (2003)
158. Sivic, J., Zisserman, A.: Efficient visual search of videos cast as text retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(4), 591–605 (2009)
159. Steitz, J.M.O., Saedan, F., Roth, S.: Multi-view X-ray R-CNN. *arXiv preprint [arXiv:1810.02344](https://arxiv.org/abs/1810.02344)* (2018)
160. Strecker, H.: Automatic detection of explosives in airline baggage using elastic X-ray scatter. *Medicamundi* **42**, 30–33 (1998)
161. Suyama, F.M., Delgado, M.R., da Silva, R.D., Centeno, T.M.: Deep neural networks based approach for welded joint detection of oil pipelines in radiographic images with double wall double image exposure. *NDT & E Int.* **105**, 46–55 (2019)
162. Tang, Y., Zhang, X., Li, X., Guan, X.: Application of a new image segmentation method to detection of defects in castings. *Int. J. Adv. Manufact. Technol.* **43**(5–6), 431–439 (2009)
163. Tang, Z., Tian, E., Wang, Y., Wang, L., Yang, T.: Non-destructive defect detection in castings by using spatial attention bilinear convolutional neural network. *IEEE Trans. Indust. Inf.* **1** (2020)
164. Tong, T., Cai, Y., Sun, D.: Defects detection of weld image based on mathematical morphology and thresholding segmentation. In: 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4. IEEE (2012)

165. Turcsany, D., Mouton, A., Breckon, T.P.: Improving feature-based object recognition for X-ray baggage security screening using primed visualwords. In: IEEE International Conference on Industrial Technology (ICIT), pp. 1140–1145 (2013)
166. Uehara, M., Yashiro, W., Momose, A.: Effectiveness of X-ray grating interferometry for non-destructive inspection of packaged devices. *J. Appl. Phys.* **114**(13), 134,901 (2013)
167. Uroukov, I., Speller, R.: A preliminary approach to intelligent X-ray imaging for baggage inspection at airports. *Signal Process. Res.* **4**, 1–11 (2015)
168. Van Dael, M., Verboven, P., Zanella, A., Sijbers, J., Nicolai, B.: Combination of shape and X-ray inspection for apple internal quality control: in silico analysis of the methodology based on X-ray computed tomography. *Postharvest Biol. Technol.* **148**, 218–227 (2019)
169. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. In: Proceedings of the International Conference on Multimedia, pp. 1469–1472. ACM (2010)
170. Vilar, R., Zapata, J., Ruiz, R.: An automatic system of classification of weld defects in radiographic images. *NDT & E Int.* (2009)
171. Viola, P., Jones, M.: Robust real-time object detection. *Int. J. Comput. Vis.* **57**(2), 137–154 (2004)
172. Wales, A., Halbherr, T., Schwaninger, A.: Using speed measures to predict performance in X-ray luggage screening tasks. In: 43rd Annual 2009 International Carnahan Conference on Security Technology, 2009, pp. 212–215 (2009)
173. Wang, Y., Shi, F., Tong, X.: A welding defect identification approach in X-ray images based on deep convolutional neural networks. In: International Conference on Intelligent Computing, pp. 53–64. Springer (2019)
174. Wang, Y., Sun, Y., Lv, P., Wang, H.: Detection of line weld defects based on multiple thresholds and support vector machine. *NDT & E Int.* **41**(7), 517–524 (2008)
175. Wang, Y., Wang, M., Zhang, Z.: Microfocus X-ray printed circuit board inspection system. *Optik-Int. J. Light and Electron Opt.* **125**(17), 4929–4931 (2014)
176. Wells, K., Bradley, D.: A review of X-ray explosives detection techniques for checked baggage. *Appl. Radiat. Isotopes* (2012)
177. Wu, J.h., Yan, X.y., Wang, G.: High-resolution pcb board defect detection system based on non-destructive detection. *Instrum. Tech. Sens.* **6**, 028 (2013)
178. Xu, C., Han, N., Li, H.: A dangerous goods detection approach based on yolov3. In: Proceedings of the 2018 2Nd International Conference on Computer Science and Artificial Intelligence, CSAI '18, pp. 600–603. ACM, New York (2018). <https://doi.org/10.1145/3297156.3297199> <http://doi.acm.org/10.1145/3297156.3297199>
179. Yahaghi, E., Mirzapour, M., Movafeghi, A.: Enhancing flaw detection in aluminum castings by two different mixed noise removal methods. *Phys. Script.* **95**(7), 075,302 (2020)
180. Yang, J., Zhao, Z., Zhang, H., Shi, Y.: Data augmentation for X-ray prohibited item images using generative adversarial networks. *IEEE Access* **7**, 28894–28902 (2019)
181. Yirong, Z., Dong, D., Baohua, C., Linhong, J., Jiluan, P.: Automatic weld defect detection method based on kalman filtering for real-time radiographic inspection of spiral pipe. *NDT & E Int.* (2015)
182. Yongwei, Y., Liuqing, D., Cuilan, Z., Jianheng, Z.: Automatic localization method of small casting defect based on deep learning feature. *Chinese J. Sci. Instrum.* **2016**(6), 21 (2016)
183. Yuanxi, W., Liu, X.: Dangerous goods detection based on transfer learning in X-ray images. *Neural Comput. Appl.* (2019). <https://doi.org/10.1007/s00521-019-04360-0>
184. Zakaria, S., Amir, A., Yaakob, N., Nazemi, S.: Automated detection of printed circuit boards (pcb) defects by using machine learning in electronic manufacturing: Current approaches. *MS&E* **767**(1), 012,064 (2020)
185. Zapata, J., Vilar, R., Ruiz, R.: Automatic inspection system of welding radiographic images based on ann under a regularisation process. *J. Nondestruct. Eval.* **31**(1), 34–45 (2012)
186. Zentai, G.: X-ray imaging for homeland security. In: IEEE International Workshop on Imaging Systems and Techniques (IST 2008) pp. 1–6 (2008)

187. Zhang, J., Guo, Z., Jiao, T., Wang, M.: Defect detection of aluminum alloy wheels in radiography images using adaptive threshold and morphological reconstruction. *Appl. Sci.* **8**(12), 2365 (2018)
188. Zhang, N., Zhu, J.: A study of X-ray machine image local semantic features extraction model based on bag-of-words for airport security. *Int. J. Smart Sens. Intell. Syst.* **1**, 45–64 (2015)
189. Zhao, X., He, Z., Zhang, S.: Defect detection of castings in radiography images using a robust statistical feature. *JOSA A* **31**(1), 196–205 (2014)
190. Zhao, X., He, Z., Zhang, S., Liang, D.: A sparse-representation-based robust inspection system for hidden defects classification in casting components. *Neurocomputing* **153**, 1–10 (2015)
191. Zhong, J., Zhang, F., Lu, Z., Liu, Y., Wang, X.: High-speed display-delayed planar X-ray inspection system for the fast detection of small fishbones. *J. Food Process Eng.* **42**(3), e13,010 (2019)
192. Zhu, Z., Hu, Y.C., Zhao, L.: Gamma/X-ray linear pushbroom stereo for 3D cargo inspection. *Mach. Vis. Appl.* **21**(4), 413–425 (2010)
193. Zhu, Z., Zhao, L., Lei, J.: 3D measurements in cargo inspection with a gamma-ray linear pushbroom stereo system. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR-05)* (2005)
194. Zou, L., Yusuke, T., Hitoshi, I.: Dangerous objects detection of X-ray images using convolution neural network. In: *International Conference on Security with Intelligent Computing and Big-Data Services*, pp. 714–728. Springer (2018)