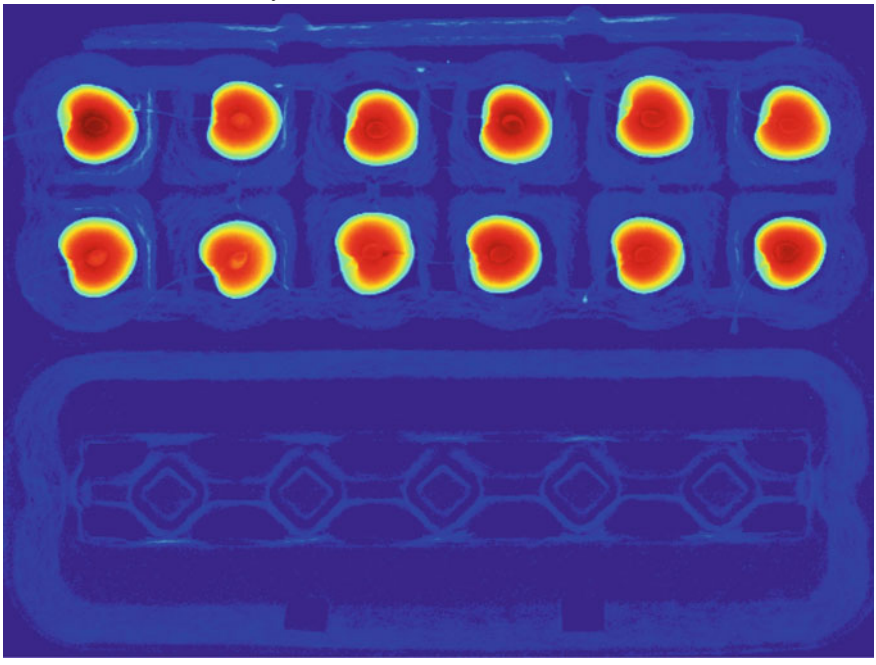
# Chapter 2
# Images for X-ray Testing

**Abstract** In this chapter, we present the dataset that is used in this book to illustrate and test several methods. The database consists of 23,189 X-ray images. The images are organized in a public database called GDXray+ that can be used free of charge, but for research and educational purposes only. The database includes five groups of X-ray images: castings, welds, baggage, natural objects, and settings. Each group has several series, and each series several X-ray images. Most of the series are annotated or labeled. In such cases, the coordinates of the bounding boxes of the objects of interest or the labels of the images are available in standard text files. The size of GDXray+ is 4.5 GB and it can be downloaded from our website.



Cover image: *X-ray image of cherries in an egg crate (X-ray image* `N0006_0027` *colored with 'jet' colormap).*



D. Mery and C. Pieringer, *Computer Vision for X-Ray Testing*,
https://doi.org/10.1007/978-3-030-56769-9_2

## 2.1 Introduction

Public databases of X-ray images can be found for medical imaging,[1] however, to the best knowledge of the author, up until now there have not been any public databases of digital X-ray images for X-ray testing for general purposes.[2]

As a service to the X-ray testing community, we collected more than 23,100 X-ray images for the development, testing, and evaluation of image analysis and computer vision algorithms. The images are organized in a public database called GDXray+.[3] In order to illustrate our database, a random selection of 120 X-ray images is shown in Fig. 2.1. The database includes five groups of X-ray images: castings, welds, baggage, natural objects, and settings. Each group has several series, and each series several X-ray images. Most of the series are annotated or labeled. In those cases, the coordinates of the bounding boxes of the objects of interest or the labels of the images are available. In Table 2.1, we can see some statistics. The size of GDXray+ is 4.54 GB, and it can be downloaded from our website (see Fig. 2.2).

In this chapter, we will view the structure of GDXray+ database, a description for each group (with some series examples), some examples of applications that have been published using images of GDXray+ and some examples in Python that can be used to manipulate the database. More details about GDXray+ are given in Appendix A

## 2.2 Structure of the Database

GDXray+ is available in a public repository. The repository contains 5 group folders one for each group: `Castings`, `Welds`, `Baggage`, `Nature`, and `Settings`. For each group, we define an initial: `C`, `W`, `B`, `N`, and `S`, respectively. As shown in Table 2.1, each group has several series. Each series is stored in an individual sub-folder of the corresponding group folder. The sub-folder name is `Xssss`, where `X` is the initial of the group and `ssss` is the number of the series. For example, the third series of group

[1] See, for example, a good collection in http://www.via.cornell.edu/databases/.

[2] There are some galleries of X-ray images available on the web with a few samples, see, for instance, http://www.vidisco.com/ndt_solutions/ndt_info_center/ndt_x_ray_gallery with approximately 50 X-ray images; and a very large dataset (more than 1 million images) for baggage inspection with no annotations [36].

[3] Available on https://domingomery.ing.puc.cl/material/gdxray/. Originally the name was GDXray [33]. The name comes from 'The Grima X-ray database' (Grima was the name of our Machine Intelligence Group at the Department of Computer Science of the Pontificia Universidad Católica de Chile). Now, we release an extended version of the dataset that we call GDXray+. The X-ray images included in GDXray+ can be used free of charge, but for research and educational purposes only. Redistribution and commercial use is prohibited. Any researcher reporting results which use this database should acknowledge the GDXray+ database by citing this chapter.

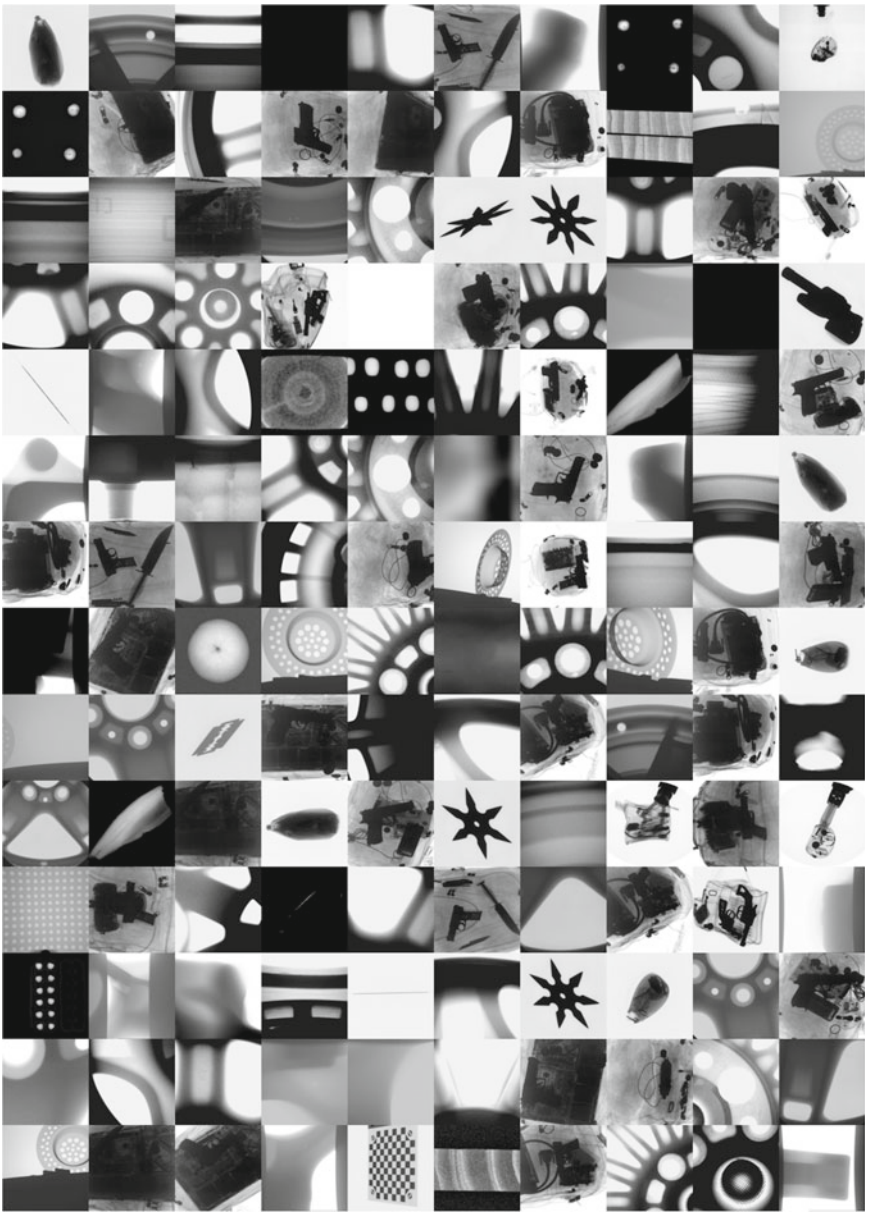


**Fig. 2.1** Random X-ray images of 𝔾𝔻𝕏ray+ database

Castings is stored in sub-folder `C0003` of folder `Castings`. The X-ray images of a series are stored in file `Xssss_nnnn.png`. Again `Xssss` is the name of the series. The number `nnnn` corresponds to the number of the X-ray image of this series. For

**Table 2.1** Statistics of GDXray+ database

| Groups | Series | Images | Size (MB) |
|---|---|---|---|
| Castings | 85 | 3768 | 664.8 |
| Welds | 4 | 98 | 209.5 |
| Baggages | 86 | 10863 | 3403.4 |
| Nature | 13 | 8290 | 191.9 |
| Settings | 8 | 170 | 73.1 |
| Total | 196 | 23189 | 4542.6 |

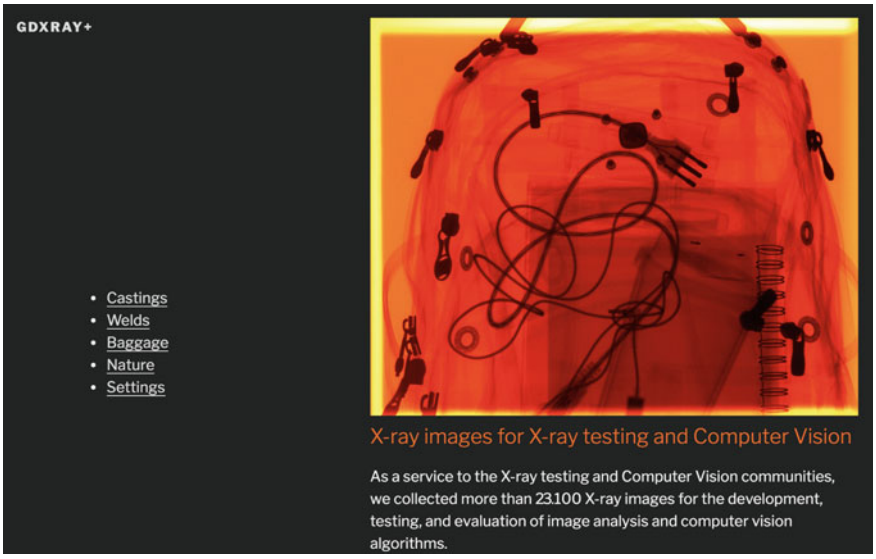



**Fig. 2.2** Screenshot of GDXray+ website. The figure shows X-ray image of a backpack using pseudo coloring ('hot' colormap): `B0083_0031.png`

example, the fifth X-ray image of series `C0003` is `C0003_0005.png` and is stored in folder `Castings/C0003`. The whole structure is summarized in Table 2.2. All X-ray images of GDXray+ are stored in 'png' (Portable Network Graphics)[4] format.

## 2.3 Castings

The group Castings contains 3,768 X-ray images arranged in 68 series. The X-ray images were taken mainly from automotive parts (aluminum wheels and knuckles). Some examples are illustrated in Figs. 2.3, 2.4 and 2.5. The details of each series are

[4] See http://www.libpng.org/pub/png/.

**Table 2.2** Structure of GDXray+

| Database | Groups | Series | X-ray images |
|---|---|---|---|
| GDXray → | Castings → | C0001 → | C0001_0001.png ... C0001_0072.png |
| | | : | |
| | | C0085 → | C0085_0001.png ... C0085_0686.png |
| | Welds → | W0001 → | W0001_0001.png ... W0001_0010.png |
| | | : | |
| | | W0004 → | W0004_0001.png ... W0004_0010.png |
| | Baggage → | B0001 → | B0001_0001.png ... B0001_0014.png |
| | | : | |
| | | B0086 → | B0086_0001.png ... B0086_1000.png |
| | Nature → | N0001 → | N0001_0001.png ... N0001_0013.png |
| | | : | |
| | | N0013 → | N0013_0001.png ... N0013_0006.png |
| | Settings → | S0001 → | S0001_0001.png ... S0001_0018.png |
| | | : | |
| | | S0008 → | S0008_0001.png ... S0008_0018.png |

**Fig. 2.3** Some X-ray images of an aluminum wheel (group Castings series C0001)

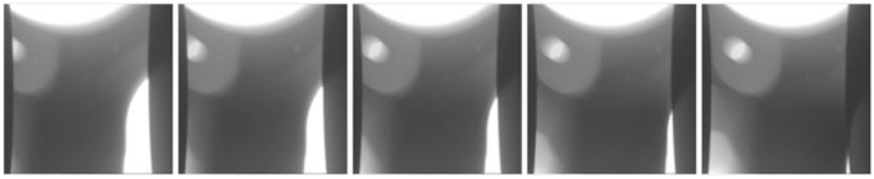


**Fig. 2.4** Some X-ray images of a knuckle (group Castings series C0059)

given in Table A.2. Experiments on these data can be found in several publications as shown in Table 2.3. It is interesting to highlight that series C0001 (see Fig. 2.3) contains not only a sequence of 72 X-ray images taken from an aluminum wheel by rotating its central axis in $5^0$, but also annotations of bounding boxes of the ground truth of 226 small defects and the calibration matrix of each image that relates the 3D coordinates of the aluminum wheel with 2D coordinates of the X-ray image.

**Table 2.3** Applications of series Castings

| Series | Application | References |
|---|---|---|
| C0001 | Detection of defects in multiple views | [3, 16, 22, 26, 39, 40] |
| | Estimation of epipolar geometry with distortion | [24] |
| | Calibration of X-ray imaging system with image intensifiers | [26] |
| | Simulation of casting defects | [26] |
| | Detection defects using deep learning (CNN) and classic features | [5, 6, 19, 20, 35] |
| C0002 | Experiments on detection of defects in single views | [7, 14, 23, 41] |
| C0008 | Simulation of casting defects | [11] |
| C0010 | Detection defects using deep learning (CNN) and classic features | [19, 20, 35] |
| C0015 | Detection defects using deep learning (CNN) and classic features | [5, 6, 19, 20, 35] |
| C0017 | Simulation of casting defects | [13, 28] |
| C0021 | Detection defects using deep learning (CNN) and classic features | [19, 20, 35] |
| C0031 | Detection defects using deep learning (CNN) and classic features | [5, 6, 19, 20, 35] |
| C0032 | Experiments on detection of defects in multiple views | [16] |
| C0034 | Detection defects using deep learning (CNN) and classic features | [5, 6, 19, 20, 35] |
| C0037 | Simulation of casting defects | [13, 28] |
| C0054 | Detection of casting on moving castings | [27] |
| C0055 | Image restoration in blurred X-ray images | [25] |
| C0061 | Detection defects using deep learning (CNN) and classic features | [19, 20, 35] |

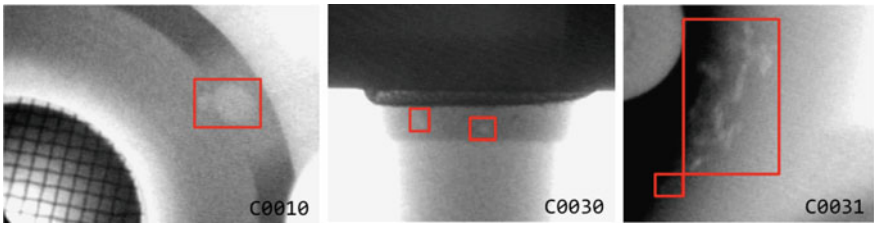



**Fig. 2.5** Some annotated images showing bounding boxes of casting defects

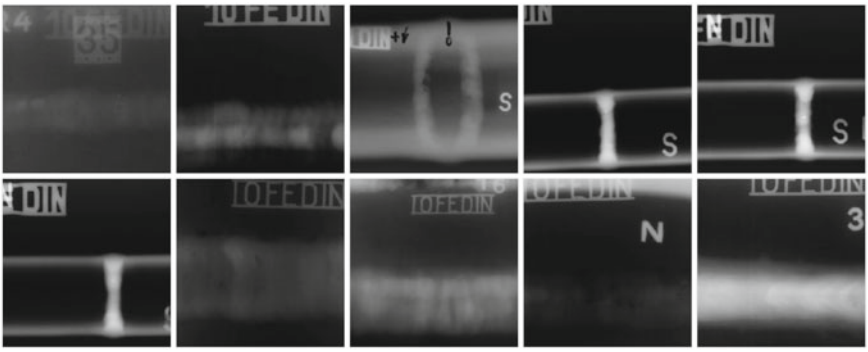


**Fig. 2.6** Some X-ray images of group `Welds` series `W0003`. This series corresponds to the BAM database

**Table 2.4** Applications of series Welds

| Series | Application | References |
|---|---|---|
| `W0001` | Detection of defects in welds using classic methods | [2, 8, 17, 21, 49] |
| | Simulation of welding defects | [12, 17, 28] |
| | Detection of defects in welds using deep learning | [9, 10, 47, 48] |
| `W0002` | Evaluation of performance of detection algorithm | [2] |
| `W0003` | Detection of defects in welds using classic methods | [38, 46, 49] |
| | Detection of defects in welds using deep learning | [9, 10, 48] |

## 2.4 Welds

The group Welds contains 98 images arranged in 4 series. The X-ray images were taken by the Federal Institute for Materials Research and Testing, Berlin (BAM).[5] Some examples are illustrated in Fig. 2.6. The details of each series are given in Table A.4. Experiments on these data can be found in several publications as shown in Table 2.4. It is interesting to highlight that series `W0001` and `W0002` (see Fig. 2.7) contains not only 10 X-ray images selected from the whole BAM database (series `W0003`), but also annotations of bounding boxes and the binary images of the ground truth of 641 defects.

[5] The X-ray images of series `W0001` and `W0003` are included in GDXray, thanks to the collaboration of the Institute for Materials Research and Testing (BAM), Berlin http://dir.bam.de/dir.html.

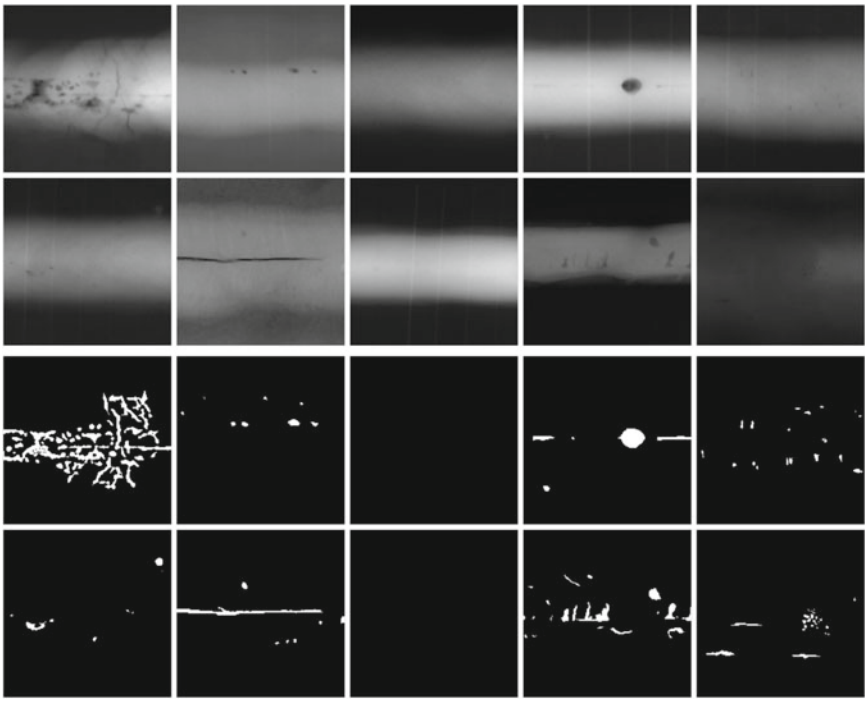


**Fig. 2.7** Some images of group `Welds` series `W0001` (X-ray images) and `W0002` (ground truth)

## 2.5 Baggage

The group Baggage contains 10,863 X-ray images arranged in 86 series. The X-ray images were taken from different containers such as backpacks, pen cases, and wallets, etc. Some examples are illustrated in Figs. 2.8, 2.9, 2.10, 2.11, and 2.12. The details of each series are given in Table A.23. Experiments on these data can be found in several publications as shown in Table 2.5. It is interesting to highlight that series `B0046`, `B0047`, and `B0048` (see, for example, Fig. 2.8) contains 600 X-ray images that can be used for automated detection of handguns, shuriken, and razor blades (bounding boxes for these objects of interest are available as well). In this case, the training can be performed using series `B0049`, `B0050`, and `B0051` that includes X-ray images of individual handguns, shuriken, and razor blades, respectively, taken from different points of view as shown in Fig. 2.9.

**Fig. 2.8** Some X-ray images of a bag containing handguns, *shuriken*, and razor blades (group `Baggage` series `B0048`)

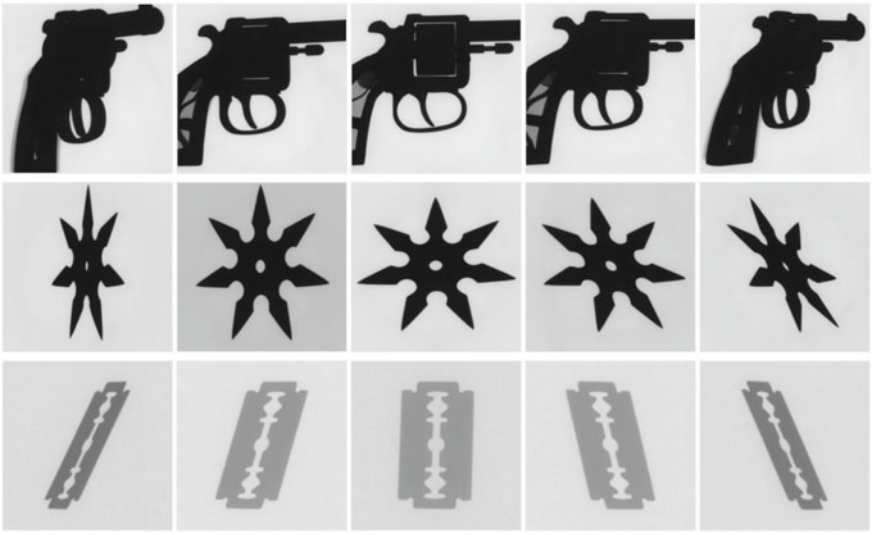


**Fig. 2.9** Some X-ray images of handguns (series `B0049`), *shuriken* (series `B0050`), and razor baldes (series `B0051`) of group `Baggage`

**Table 2.5** Applications of series Baggage

| Series | Application | References |
|---|---|---|
| B0005 | Experiments on detection of pins in multiple views | [16, 44] |
| | Detection of razor blades using active vision | [44] |
| B0007 | Training of a classifier of razor blades | [44] |
| B0009-43 | Experiments on detection of handguns | [4, 32] |
| B0045 | Experiments on detection of objects in multiple views | [18, 34] |
| | Active vision | [44] |
| B0046-51 | Simulation of threat objects | [29] |
| | Detection of threat objects using sparse representations | [29] |
| | Detection of threat objects using 3D reconstruction | [43] |
| | Detection of threat objects using active vision | [42] |
| | Detection of threat objects using deep learning | [1] |
| B0049-51 | Detection of threat objects | [35, 45] |
| B0055 | Experiments on detection of objects in sequences of four views | [18] |
| B0056 | Experiments on detection of objects in sequences of six views | [18] |
| B0057 | Experiments on detection of objects in sequences of eight views | [18] |
| B0058 | Training of a classifier for clips, springs, and razor blades | [18, 34] |
| B0061-73 | Detection of razor blades using active vision | [44] |
| B0078-82 | Detection of threat objects | [35, 45] |
| B0083 | Detection of threat objects | [45] |

**Fig. 2.10** A knife was rotated in $1^0$ and by each position, an X-ray image was captured. In this figure, X-ray images at $0^0$, $10^0$, $20^0$, ... $350^0$ are illustrated (see series B00008 of group Baggage)

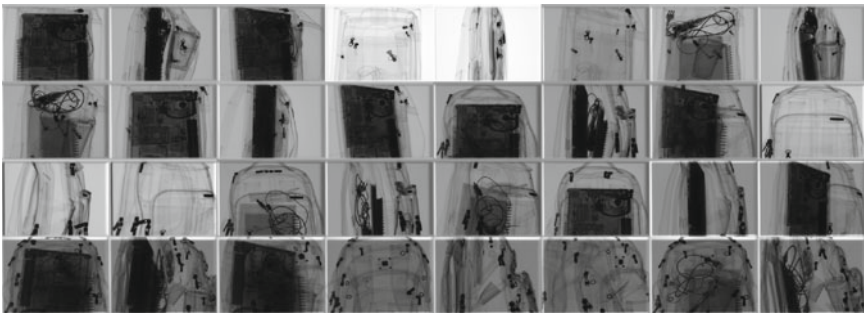


**Fig. 2.11** Backpacks with no threat objects. They can be used to superimpose the isolated threat objects of Fig. 2.9 (see series `B00083` of group `Baggage`). [→ Example 2.1  ]

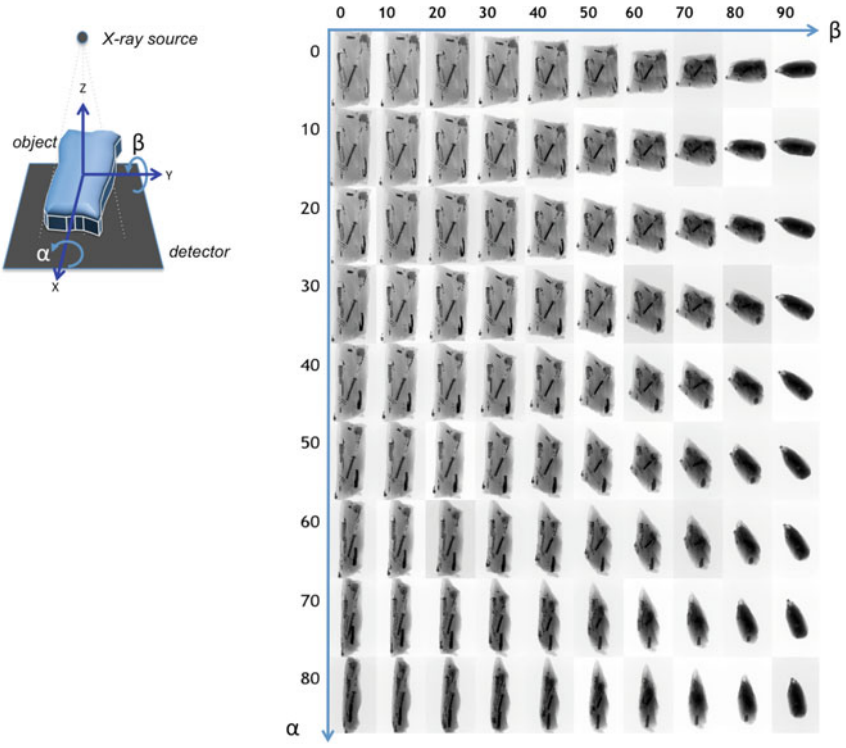



**Fig. 2.12** X-ray images of a pen case from 90 different points of view. They are obtained by rotating $\alpha$ and $\beta$ as shown in the left model (see series `B00045` of group `Baggage`)

## 2.6 Natural Objects

The group Nature contains 8,290 X-ray images arranged in 13 series. The X-ray images were taken from different natural objects such as salmon filets, fruit, and wood pieces. Some examples are illustrated in Figs. 2.13, 2.14, and 2.15. The details of each series are given in Table A.1. Experiments on these data can be found in several publications as shown in Table 2.6. It is interesting to highlight that series `N0012` and `N0013` (see Fig. 2.16) contains not only 6 X-ray images of salmon filets, but also annotations of bounding boxes and the binary images of the ground truth of 73 fish bones. For training proposes, there are more than 7,500 labeled small crops ($10 \times 10$ pixels), of regions of X-ray of salmon filets with and without fish bones in series `N0003`.

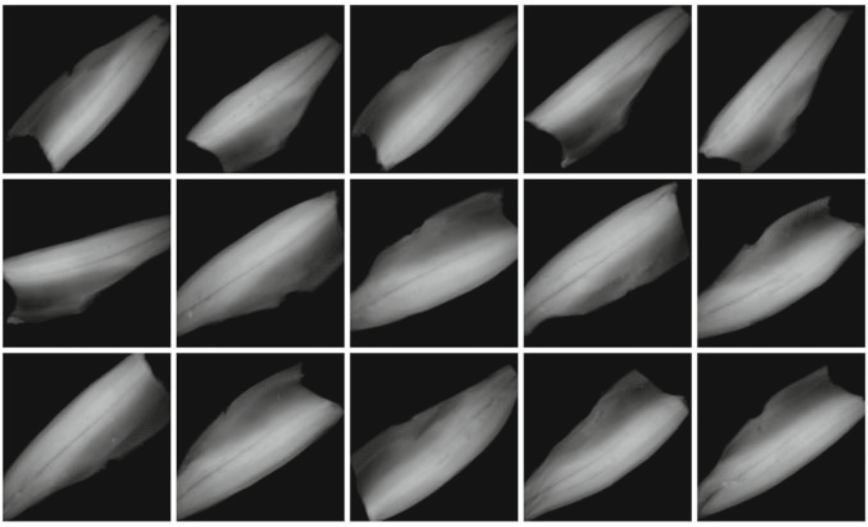


**Fig. 2.13** Some X-ray images of salmon filets (group `Nature` series `N0011`)

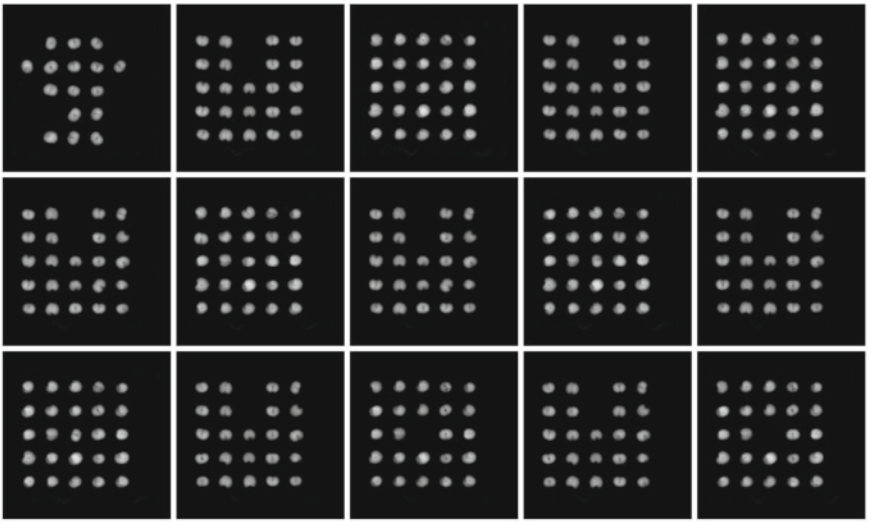


**Fig. 2.14** Some X-ray images of cherries (group `Nature` series `N0006`)

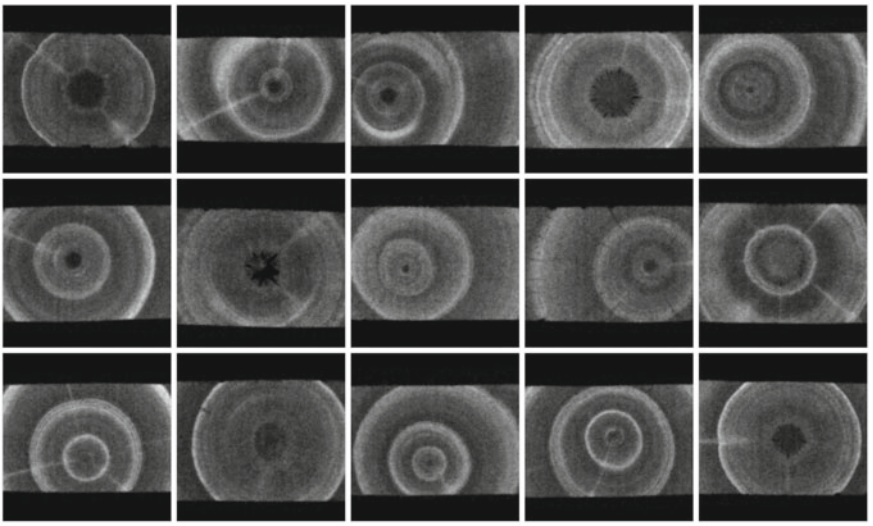


**Fig. 2.15** Some X-ray images of wood (group `Nature` series `N0010`)

**Table 2.6** Applications of series Nature

| Series | Application | References |
|---|---|---|
| N0003 | Automated design of a visual food quality system | [31] |
| N0003 | Automated fish bone detection | [30] |
| N0008 | Quality control of kiwis | [37] |
| N0011 | Automated fish bone detection | [30] |

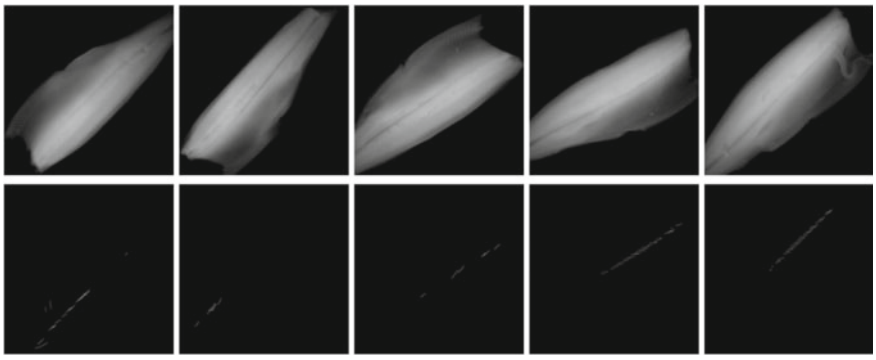


**Fig. 2.16** Some images of group Nature series S0012 (X-ray images of salmon filets) and S0013 (ground truth for fish bones)

## 2.7 Settings

The group Settings contains 170 X-ray images arranged in 8 series. The X-ray images were taken from different calibration objects such checkerboards and 3D objects with regular patterns. Some examples are illustrated in Figs. 2.17 and 2.18. The details of each series are given in Table A.5. Experiments on these data can be found in several publications as shown in Table 2.7. It is interesting to highlight that series S0008 (see Fig. 2.17) contains not only 18 X-ray images of a copper checkerboard, but also the calibration matrix of each view. In addition, series S0007 can be used for modeling the distortion of an image intensifier. The coordinates of each hole of the calibration pattern in each view are available, and the coordinates of the 3D model are given as well.

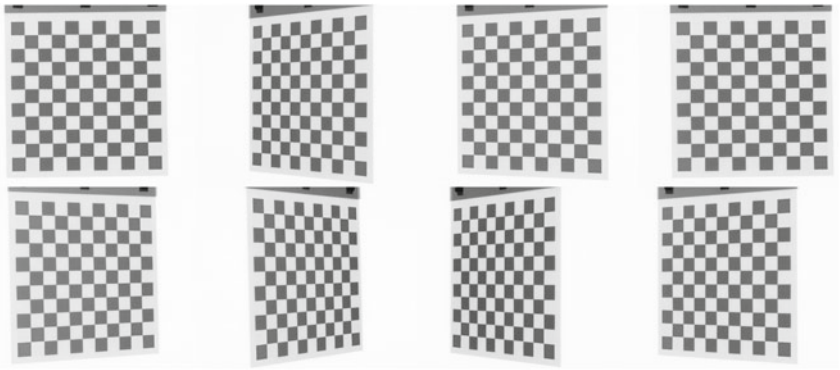


**Fig. 2.17** Some X-ray images of a copper checkerboard used by calibration (group `Settings` series `S0008`)

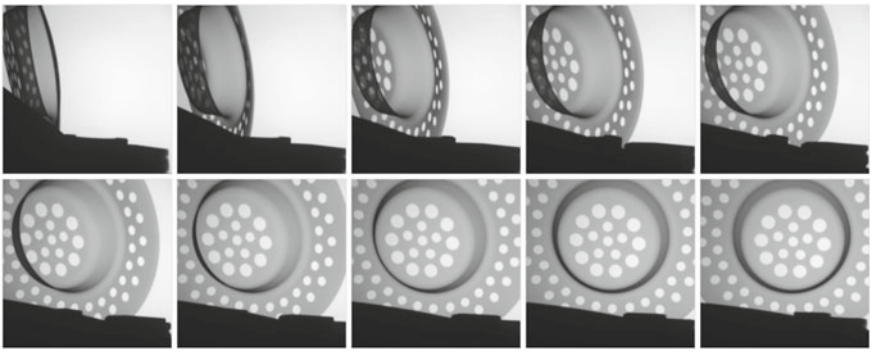


**Fig. 2.18** Some X-ray images of circular pattern in different points of view used by calibration (group `Settings` series `S0007`)

**Table 2.7** Applications of series Settings

| Series | Application | References |
|---|---|---|
| `S0001` | Calibration of a multiple view X-ray imaging system for active vision | [44] |
| `S0002` | Distortion model of an image intensifier | [24, 26] |
| `S0007` | Explicit geometric model of a radioscopic imaging system | [15] |

## 2.8 Python Commands

In order to manipulate GDXray+ database easily, some helpful Python functions were developed in pyxvis Library. In this section, we present a summary of them with some examples.

Python Example 2.1: In this example, we show how simple it is to display the X-ray images of a series of GDXray+.:

**Listing 2.1 : Display of X-ray images of GDXray+.**

```
from pyxvis.io import gdxraydb
from pyxvis.io.visualization import show_series

gdxraydb.xgdx_stats()

image_set = gdxraydb.Baggages()

image_set.describe()

print(image_set.get_dir(60))

try:
    series_dir = image_set.get_dir(83)
except ValueError as err:
    print(err)

show_series(image_set, 8, range(1, 352, 10), n=18, scale=0.2)
```

The output of this code is illustrated in Fig. 2.11. □

Some pyxvis Library functions that can be used to manipulate GDXray+ are the following:

- gdx_browse of gui: This GUI function[6] is used to browse GDXray+ database. An example is illustrated in Fig. 2.19. An additional example using pseudo coloring is shown in Fig. 2.20;, the user can select one of 10 different color maps. This function can be used to display the bounding boxes of an X-ray image. For example, in X-ray image `N0012_0004.png` the bounding boxes of the ground truth are displayed in Fig. 2.19. Each bounding box is stored as a row in file `ground_truth.txt` of folder `N0012` of GDXray+. The format of this file is as follows: one bounding box per row; the first number of the row is the number of the image of the series, and the next four values are the coordinates $x_1, x_2, y_1, y_2$ of a bounding box. Thus, the rectangle of a bounding box is defined by its opposite vertices: $(x_1, y_1)$ and $(x_2, y_2)$.
- show_series of gdxraydb: This function is used to display several images of a series in only one figure (see example in Fig. 2.10 or Example 2.1).
- get_dir of gdxraydb: This function is used to ascertain the path of a series of GDXray+.

[6] GUI: Graphic User Interface.

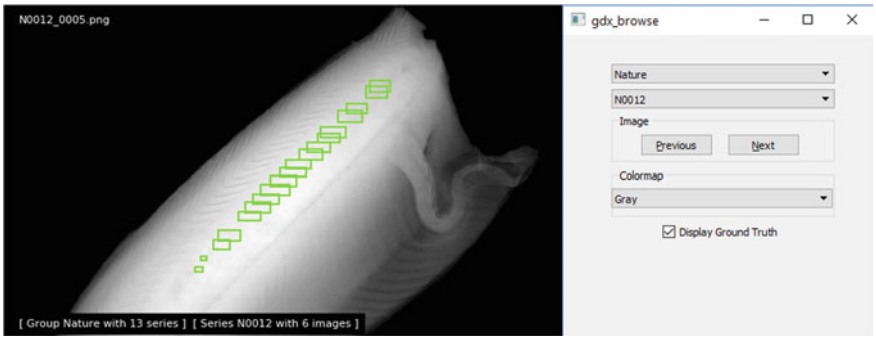



**Fig. 2.19** Example of command gdx_browse that can be used to browse 𝔾𝔻𝕏ray+. The user can click buttons `[Previous]` and `[Next]` to display the next groups, series or images. In addition, the ground truth option can be used to display manual annotations when they are available. In this example, the fish bones of a salmon filet are highlighted. For colored images, see Fig. 2.20

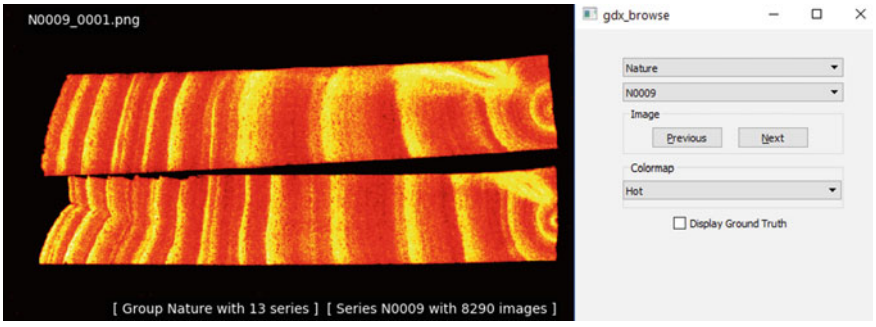



**Fig. 2.20** Example of command gdx_browse using pseudo coloring of a wood X-ray image. For another example in grayscale, see Fig. 2.19

- gdx_stats of gdxraydb: This function is used to compute some statistics of 𝔾𝔻𝕏ray+. The output is Table 2.1.
- load_image of gdxraydb: This function is used to load an image of 𝔾𝔻𝕏ray+. For example, `N0012_0004.png` can be stored in matrix `img` using the following commands:

```
from pyxvis.io import gdxraydb
image_set = gdxraydb.Nature()
img = image_set.load_image(12, 4)
```

- load_data of gdxraydb: This function is used to load a file into a workspace. For instance, the ground truth data of series `N0012` can be stored in matrix `gt` using the following commands:

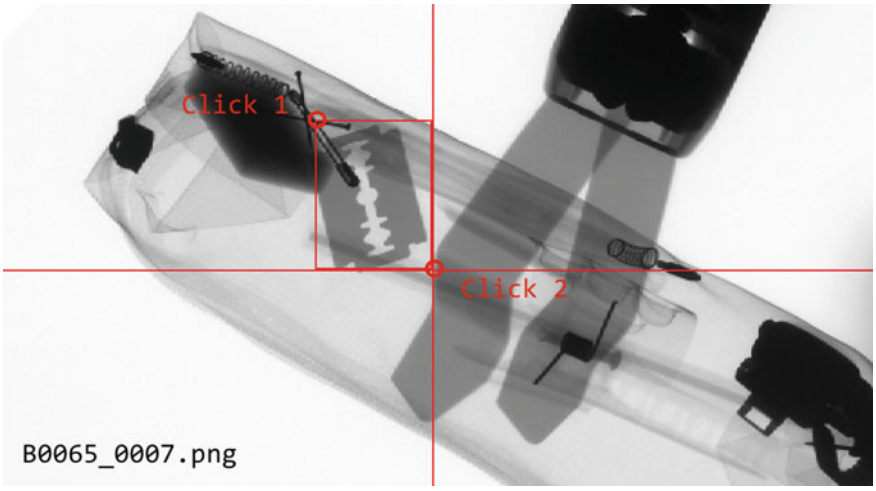



**Fig. 2.21** Annotation: there are some tools that can be used to manually annotate the ground truth of a series. In this example, the user is annotating the razor blades of series `B0065`

```
from pyxvis.io import gdxraydb
image_set = gdxraydb.Nature()
gt = load_set.load_image(12, 'ground_truth.txt')
```

For annotation, there are some open- source tools that can be used to manually annotate bounding boxes of a series of GDXray+. An example isn Fig. 2.21.[7]

## 2.9 Summary

In this chapter, we presented the details of a new public dataset called GDXray+. It consists of more than 21,100 X-ray images. The database includes five groups of X-ray images: castings, welds, baggage, natural objects, and settings. Each group has several series and X-ray images with many labels and annotations that can be used for training and testing purposes in computer vision algorithms. To the best knowledge of the author, up until now, there have not been any public databases of digital X-ray images for general purposes in X-ray testing.

In this chapter, we explained the structure of the GDXray+ database, we gave a description for each group (with some series examples), we presented some examples of applications that have been published using images of GDXray+, and some examples in Python with pyxvis Library, that can be used to manipulate the database.

[7] See, for example, LabelMe http://labelme.csail.mit.edu/Release3.0/ developed by the Computer Science and Artificial Intelligence Laboratory at MIT.

We believe that GDXray+ represents a relevant contribution to the X-ray testing community. On the one hand, students, researchers, and engineers can use these X-ray images to develop, test, and evaluate image analysis and computer vision algorithms without purchasing expensive X-ray equipment. On the other hand, these images can be used as a benchmark in order to test and compare the performance of different approaches on the same data. Moreover, the database can be used in the training programs of human inspectors.

## References


1. 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)
2. Carrasco, M., Mery, D.: Segmentation of welding defects using a robust algorithm. Mater. Eval. **62**(11), 1142–1147 (2004)
3. 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)
4. Damashek, A., Doherty, J.: Detecting guns using parametric edge matching. Technical Report Project for Computer Vision Course: CS231A, Stanford University (2015)
5. 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)
6. 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. Manuf. Syst. **2** (2018)
7. Ghoreyshi, A., Vidal, R., Mery, D.: Segmentation of circular casting defects using a robust algorithm. Insight-Non-Destr. Testing Cond. Monit. **47**(10), 615–617 (2005)
8. Hernández, S., Sáez, D., Mery, D., da Silva, R., Sequeira, M.: Automated defect detection in aluminium castings and welds using neuro-fuzzy classifiers. In: Proceedings of the 16th World Conference on Non-Destructive Testing (WCNDT–2004). Montreal (2004)
9. 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)
10. 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)
11. Huang, Q., Wu, Y., Baruch, J., Jiang, P., Peng, Y.: A template model for defect simulation for evaluating nondestructive testing in X-radiography. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. **39**(2), 466–475 (2009)
12. Liu, L., Cao, D., Wu, Y., Wei, T.: Defective samples simulation through adversarial training for automatic surface inspection. Neurocomputing **360**, 230–245 (2019)
13. Mery, D.: A new algorithm for flaw simulation in castings by superimposing projections of 3D models onto X-ray images. In: Proceedings of the XXI International Conference of the Chilean Computer Science Society (SCCC-2001), pp. 193–202. IEEE Computer Society Press, Punta Arenas (2001)
14. 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)
15. Mery, D.: Explicit geometric model of a radioscopic imaging system. NDT & E Intern. **36**(8), 587–599 (2003)

16. 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)
17. Mery, D.: Automated detection of welding defects without segmentation. Mater. Eval. **69**(6), 657–663 (2011)
18. Mery, D.: Inspection of complex objects using multiple-X-ray views. IEEE/ASME Trans. Mechatron. **20**(1), 338–347 (2015)
19. Mery, D.: Aluminum casting inspection using deep learning: A method based on convolutional neural networks. J. Nondestr. Eval. **39**(1), 12 (2020)
20. 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)
21. Mery, D., Berti, M.A.: Automatic detection of welding defects using texture features. Insight-Non-Destr. Testing Cond. Monit. **45**(10), 676–681 (2003)
22. Mery, D., Carrasco, M.: Automated multiple view inspection based on uncalibrated image sequences. Lecture Notes in Computer Science, vol. 3540, pp. 1238–1247 (2005)
23. Mery, D., Chacón, M., Munoz, L., González, L.: Automated inspection of aluminium castings using fusion strategies. Mater. Eval. (2005). In Press
24. Mery, D., Filbert, D.: The epipolar geometry in the radioscopy: Theory and application. at - Automatisierungstechnik **48**(12), 588–596 (2000). (in German)
25. Mery, D., Filbert, D.: A fast non-iterative algorithm for the removal of blur caused by uniform linear motion in X-ray images. In: Proceedings of the 15th World Conference on Non-Destructive Testing (WCNDT–2000). Rome (2000)
26. 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)
27. Mery, D., Filbert, D.: Automated inspection of moving aluminium castings. In: $8^{th}$ European Conference on Non-Destructive Testing (ECNDT 2002). Barcelona (2002)
28. Mery, D., Hahn, D., Hitschfeld, N.: Simulation of defects in aluminum castings using cad models of flaws and real X-ray images. Insight **47**(10), 618–624 (2005)
29. Mery, D., Katsaggelos, A.K.: A logarithmic X-ray imaging model for baggage inspection: simulation and object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 57–65 (2017)
30. 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)
31. Mery, D., Pedreschi, F., Soto, A.: Automated design of a computer vision system for visual food quality evaluation. Food Bioprocess Technol. **6**(8), 2093–2108 (2013)
32. 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)
33. 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. Nondestr. Eval. **34**(4), 1–12 (2015)
34. 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)
35. 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)
36. 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)
37. Mondragón, G., Leiva, G., Aguilera, J., Mery, D.: Automated detection of softening and hard columella in kiwifruits during postharvest using X-ray testing. In: Proceedings of International Congress on Engineering and Food (2011)

38. Perner, P., Zscherpel, U., Jacobsen, C.: A comparison between neural networks and decision trees based on data from industrial radiographic testing. Pattern Recognit. Lett. **22**(1), 47–54 (2001)
39. Pieringer, C., Mery, D.: Flaw detection in aluminium die castings using simultaneous combination of multiple views. Insight **52**(10), 548–552 (2010)
40. Pizarro, L., Mery, D., Delpiano, R., Carrasco, M.: Robust automated multiple view inspection. Pattern Anal. Appl. **11**(1), 21–32 (2008)
41. Ramírez, F., Allende, H.: Detection of flaws in aluminium castings: a comparative study between generative and discriminant approaches. Insight-Non-Destr. Testing Cond. Monit. **55**(7), 366–371 (2013)
42. Riffo, V., Flores, S., Mery, D.: Threat objects detection in X-ray images using an active vision approach. J. Nondestr. Eval. **36**(3), 44 (2017)
43. 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)
44. Riffo, V., Mery, D.: Active X-ray testing of complex objects. Insight **54**(1), 28–35 (2012)
45. 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)
46. da Silva, R.R., Siqueira, M.H., de Souza, M.P.V., Rebello, J.M., Calôba, L.P.: Estimated accuracy of classification of defects detected in welded joints by radiographic tests. NDT & E Intern. **38**(5), 335–343 (2005)
47. Sizyakin, R., Voronin, V., Gapon, N., Zelensky, A., Pižurica, A.: Automatic detection of welding defects using the convolutional neural network. In: Automated Visual Inspection and Machine Vision III, vol. 11061, p. 110610E. International Society for Optics and Photonics (2019)
48. 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)
49. Yahaghi, E., Movafeghi, A., Mirzapour, M., Rokrok, B.: Defect detections in industrial radiography images by a multi-scale lmmse estimation scheme. Radiat. Phys. Chem. **168**, 108560 (2020)