5.2. Test Results for Arbitrary In-plane Orientation

Up to this point, the thesis has focused on the development of an algorithm for a single detonator orientation. In order to accommodate arbitrary in-plane orientations, several things were carried out, and they are related to each other closely.

The first step was to build all sets of the correlation filters (each set for one orientation) that are needed for the multi-orientation detection system. The second step was to observe every set of the correlation filters using the images that were obtained from the AS&E system to set all the parameters (parameters of the correlation filters, thresholds for the correlation filters, threshold for the Hough transform, and the angle tolerance range for the Hough transform). Usually, building the correlation filters is the most important step. If it is found that the detection angle of the adjacent sets in the system does not merge with each other, we should go back to the first step to make some adjustments.

In order to build the correlation filters, three things should be considered very carefully. First, we need to find the relationship between the physical orientation and the image orientation. By changing the physical orientation with a step size of 5^0, then measuring the image orientation, we obtained the relationship shown in Table 5.2. The corresponding plot is shown in Fig. 5.15. The relationship between the physical orientation and the image orientation is not linear. Hence we can not simply choose the correlation filters at an equal amount of the changes of the physical angle.

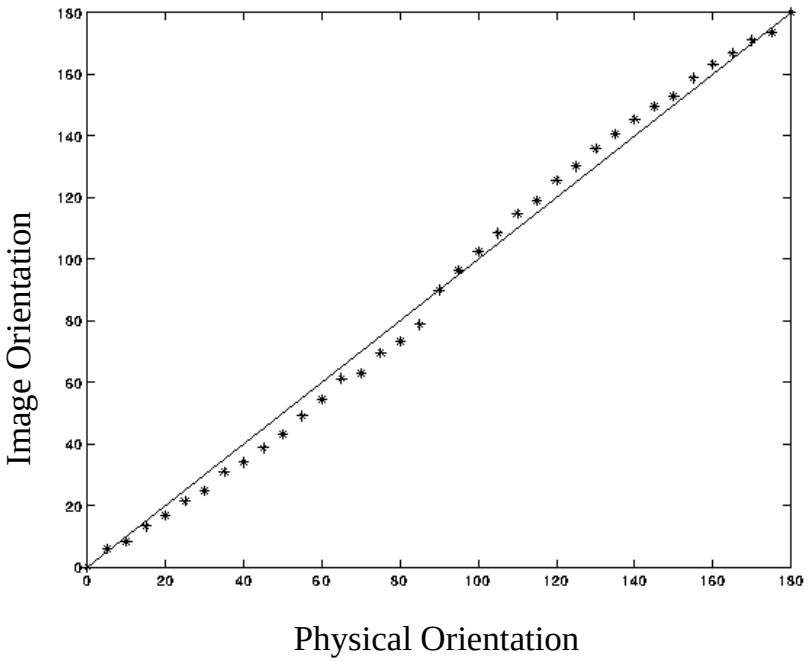


Fig. 5.15. The relationship between physical orientation and orientation in the image.

Second, the angle tolerance for each set of correlation filters should be tested one by one. This is because the angle tolerance is different for different orientations. For example, by using same parameters (thresholds for the correlation filters, threshold for the Hough transform), for orientation $\phi = 40^0$, the angle tolerance is about 29^0. However, for orientation $\phi = 0^0$, the angle tolerance is about 15^0. This is because that the same amount of angle change at different orientations will cause different shape changes. When the orientation of the orientation is closer to horizontal or vertical direction, its shape is more sensitive to the orientation change than when its orientation is around $+45^0$ or -45^0.

Table 5.2. The relationship between physical orientation and orientation in the image.

Physical Angle (degrees)	Image Angle (degrees)	Physical Angle (degrees)	Image Angle (degrees)	Physical Angle (degrees)	Image Angle (degrees)
0	0	65	61.2	130	135.8
5	5.9	70	62.9	135	140.6
10	8.3	75	69.6	140	145.4
15	13.3	80	73.2	145	149.6
20	16.7	85	78.7	150	152.9
25	21.4	90	90	155	158.8
30	24.8	95	96.3	160	163.3
35	31.0	100	102.5	165	166.8
40	34.0	105	108.4	170	171.0
45	38.7	110	114.8	175	173.5
50	43.2	115	118.9	180	180
55	49.1	120	125.7		
60	54.4	125	130.1		

Third, the size of the filters should be chosen carefully. This is because that the angle tolerance and the filter effectiveness are dramatically affected by the size of the filters. We have shown this in previous sections of this chapter. Normally, we test several different filters for each orientation, then choose the one with the best output results. After the orientation tolerance is tested, the angle tolerance range for the Hough transform is determined correspondingly.

Once the detection system for multi- orientation was built, much testing was done to adjust the system. The testing results of our current system are shown in the following section.

In our multi-orientation detection system, ten sets of the correlation filters are built. Their physical angles and the corresponding image angles are shown in Table 5.3.

Table 5.3. The angle of the correlation filter in the multi-orientation detection system.

Set Number	Physical Angle (degrees)	Image Angle (degrees)
1	0	0
2	10	8.3
3	25	21.4
4	55	49.1
5	80	73.2
6	90	90
7	100	102.5
8	120	125.6
9	150	152.9
10	170	171.0

Nineteen luggage images have been tested using this multi-orientation detection system. All these bags contain a detonator. In this section, some examples will be shown, and the results will be analyzed in detail.

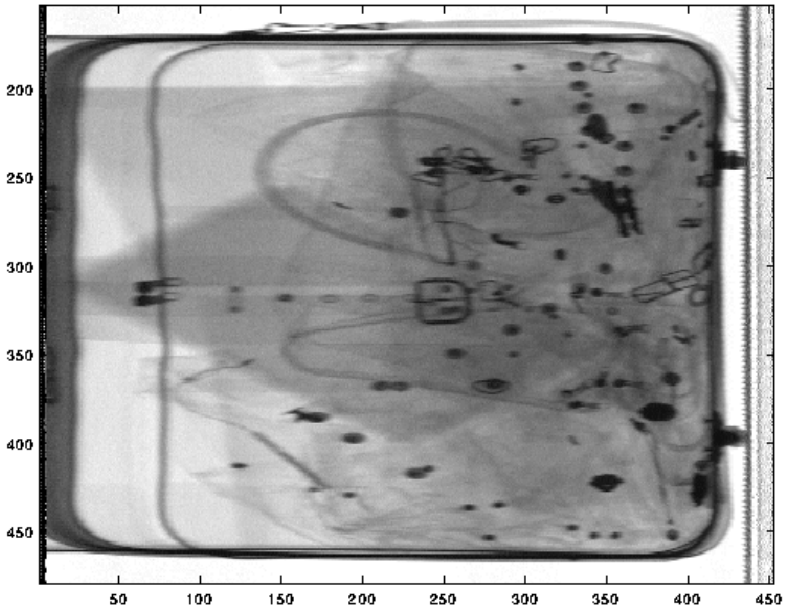
Figs. 5.16 - 5.18 show some of the cases in which the detonator was detected successfully. In those figures (*a*) is the original image and (*b*) shows the output image. In Fig. 5.16 a detonator is located around row 400 and column 150, and this image is an example of a simple case without overlapping. The detonator is detected with no false alarm. In Fig. 5.17 a detonator (around row 150 and column 225) intersects a dark point, while in

Fig. 5.18 a detonator (around row 75 and column 120) overlaps a board. In both cases, the software successfully detects the detonator with no false alarm.

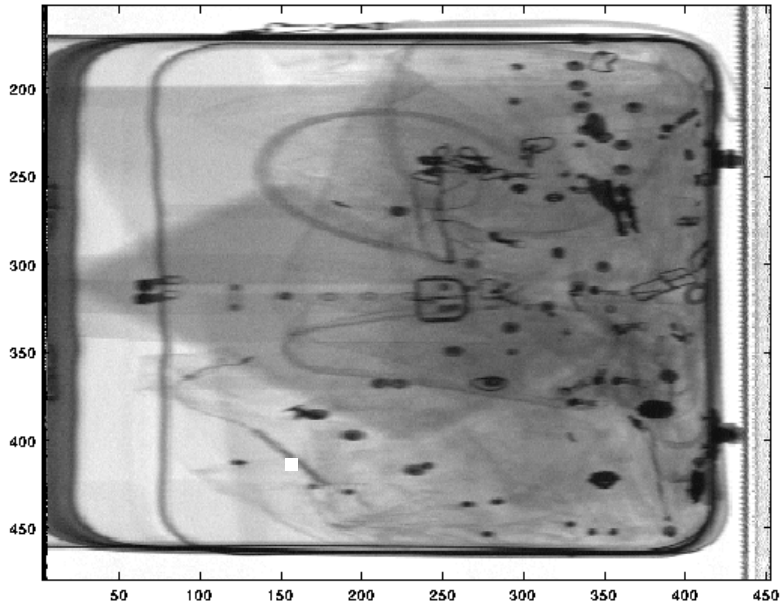
In some of our experiments, false alarms occurred. Fig. 5.19 shows one of these cases. While the detonator (around row 50 and column 160) was detected, there was a false alarm around row 110 and column 175.

Fig. 5.20 and Fig. 5.22 show two more cases for which the detection system failed. In Fig. 5.20, there are two reasons that caused the detection to fail. One is that a small dark object below the detonator results in an absence of peak points around the left end, causing the left end points to be far away from the peak points. Since the image fusion algorithm only searches for end points around the ends of the peak points, the detonator was not detected. Another reason is that the threshold we set for the end point (0.83) resulted in an absence of detected right end points. Once we reduced the threshold value to 0.75, the right end points showed up, and the detonator was detected (Fig. 5.21). However a lower threshold value will result in false alarm in other luggage images, so 0.83 is still used in the multi-orientation detection system.

In Fig. 5.22, the luggage was put at the most distant part of the conveyor, and the detonator (around row 200 and column 390) was intersected with a dark object, causing fewer peak points to appear (Fig. 5.22*b*). This resulted in the maximum value of the corresponding Hough array smaller than the threshold.

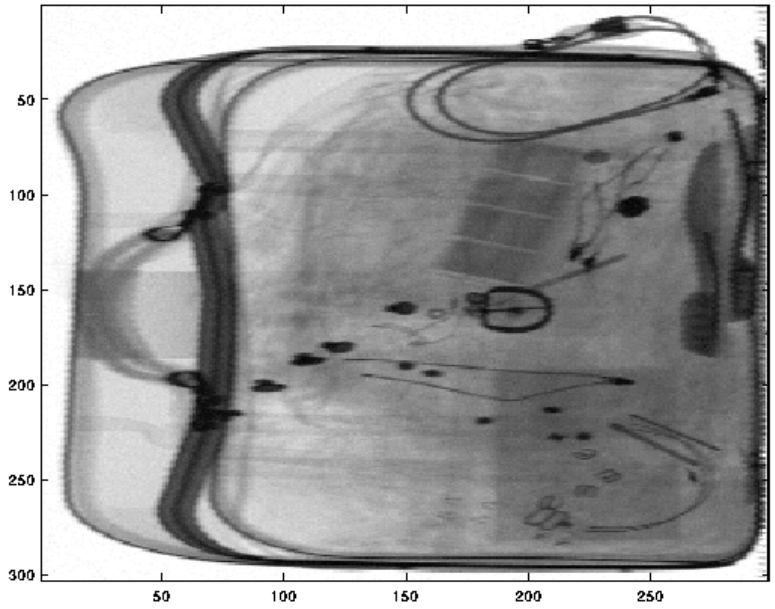


(*a*)

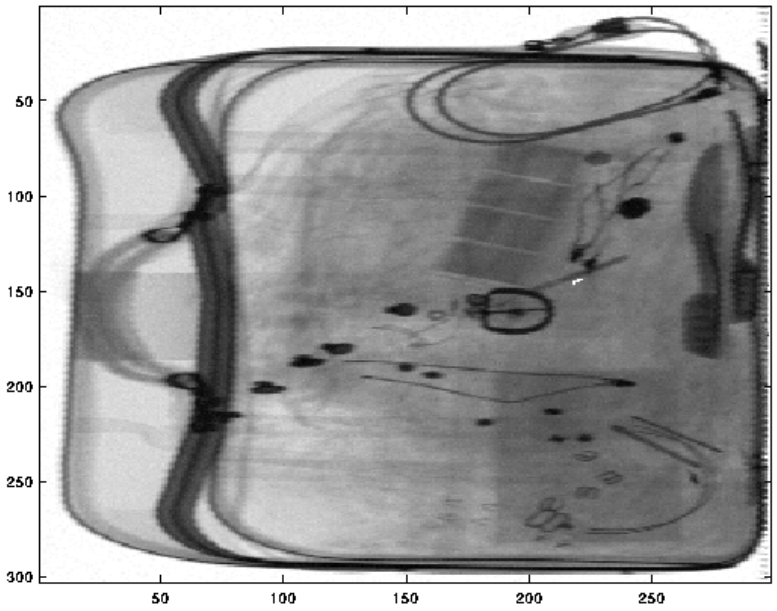


(*b*)

Fig. 5.16. Example of arbitrary orientation algorithm. The detonator is around row 400 and column 150 and it does not overlap other objects. (*a*) Original Image. (*b*) Final result. The detonator has been detected, and no false matches are present. Since there are only a few pixels highlighted and it is difficult to see, the output has been enlarged.

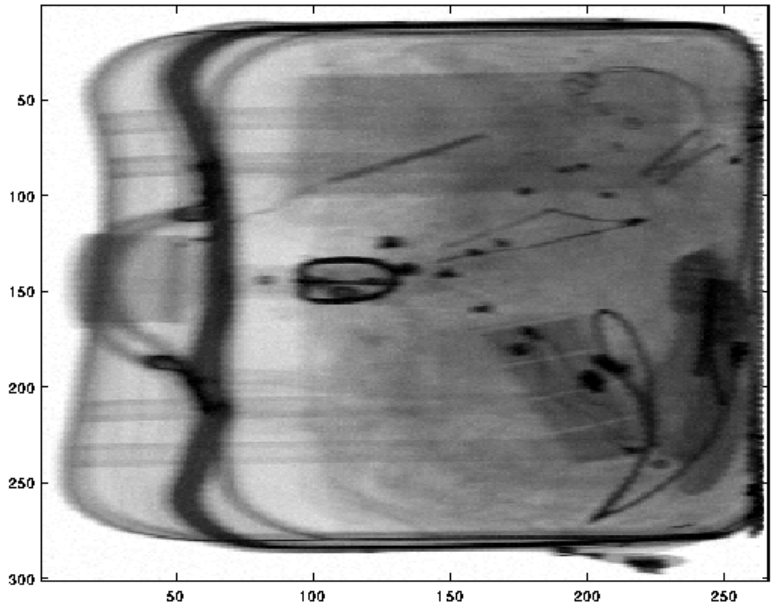


(*a*)

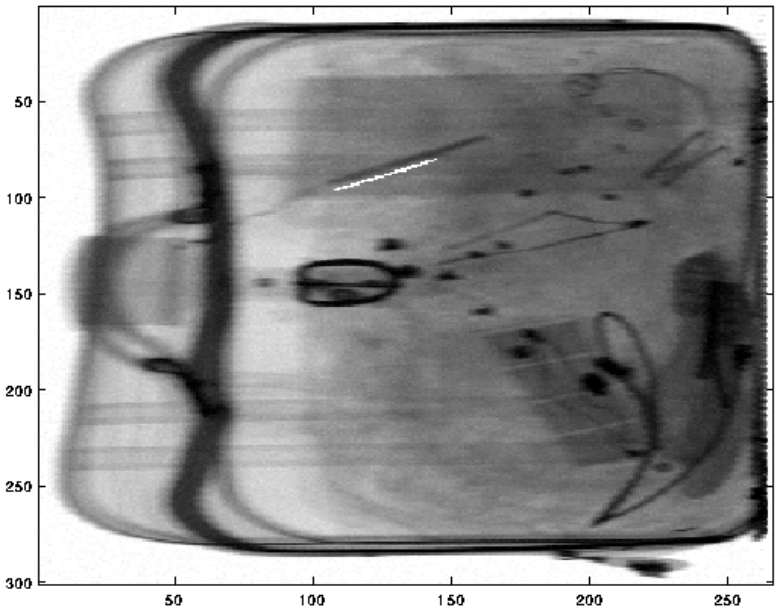


(*b*)

Fig. 5.17. Example of arbitrary orientation algorithm. The detonator is around row 150 and column 225 and it overlaps a dark point. (*a*) Original Image. (*b*) Final result. The detonator has been detected, and no false matches are present.

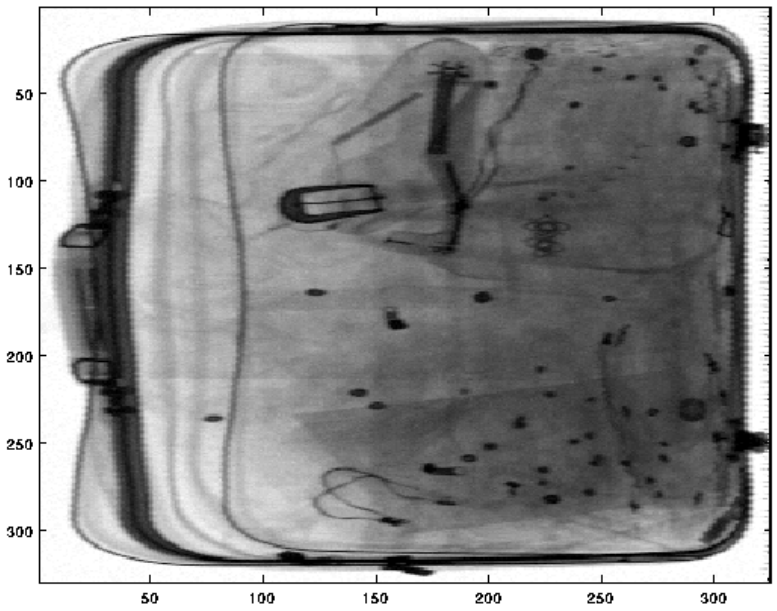


(*a*)

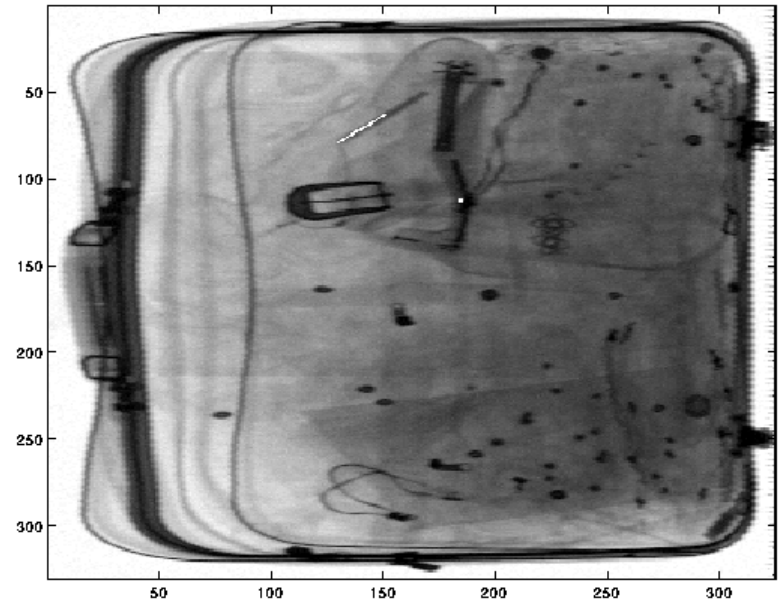


(*b*)

Fig. 5.18. Example of arbitrary orientation algorithm. The detonator is around row 70 and column 120 and it overlaps a board. (*a*) Original Image. (*b*) Final result. The detonator has been detected, and no false matches are present.

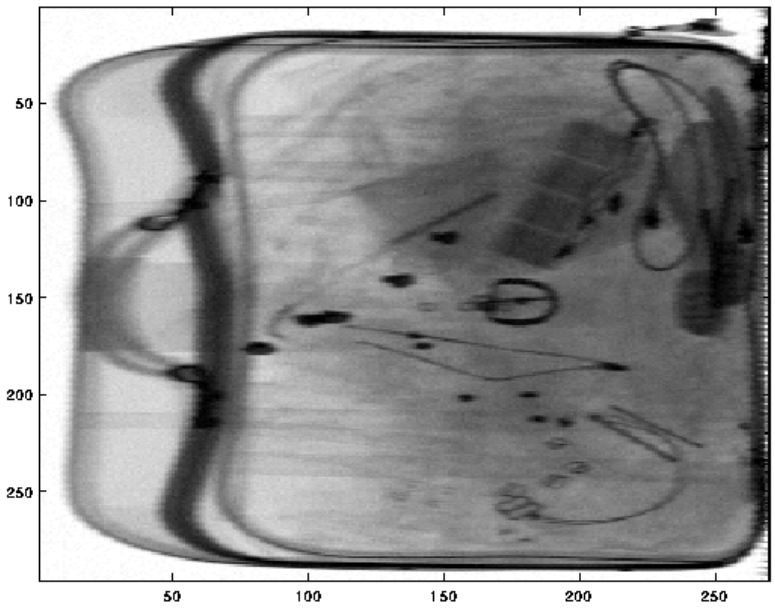


(*a*)

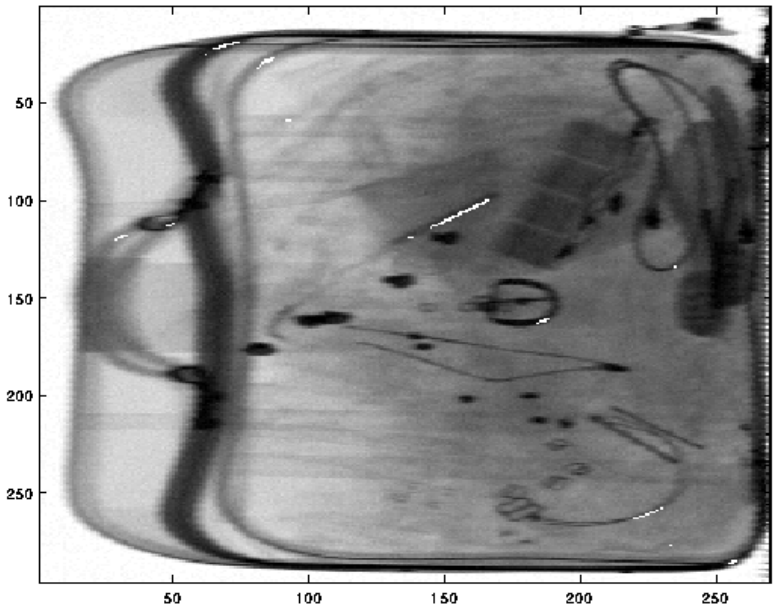


(*b*)

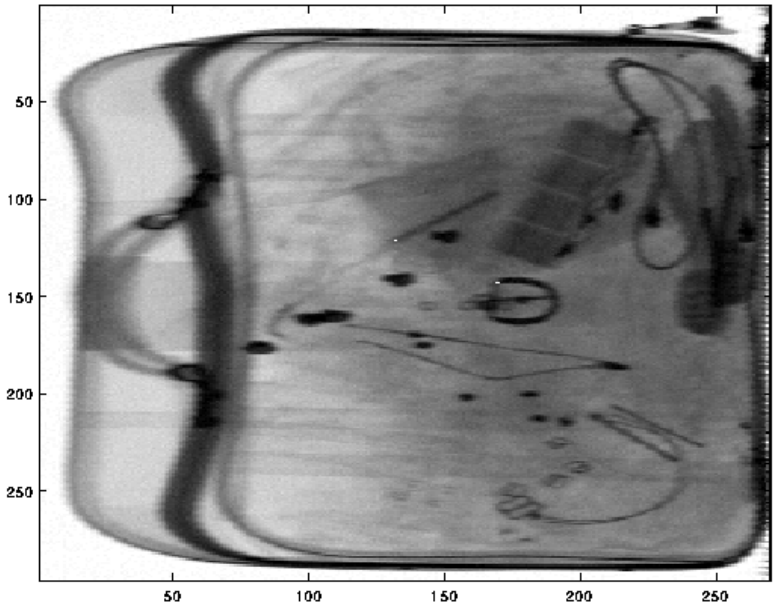
Fig. 5.19. Example of a false positive. The detonator is around row 50 and column 150. (*a*) Original Image. (*b*) Final result. The detonator has been detected, and there is a false match around row 110 and column 190.


50
100
150
200
250
50
100
150
200
250

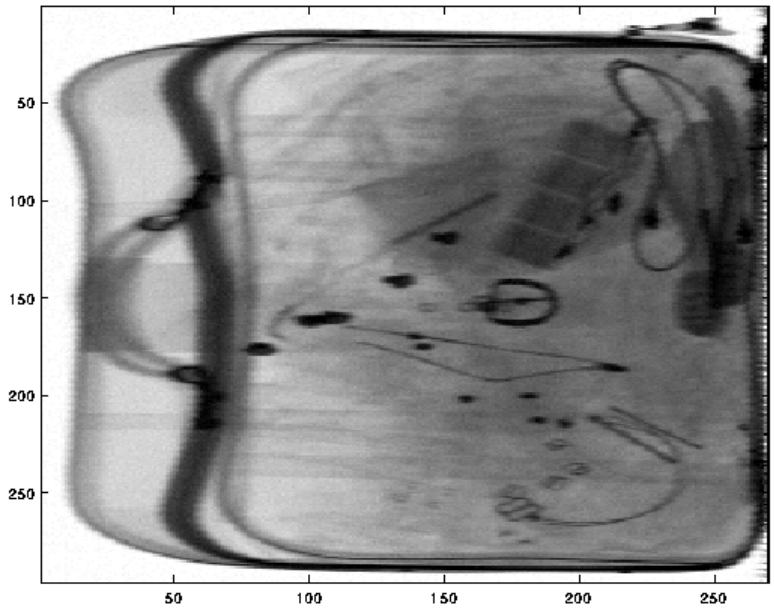
(*a*)


50
100
150
200
250
50
100
150
200
250

(*b*)

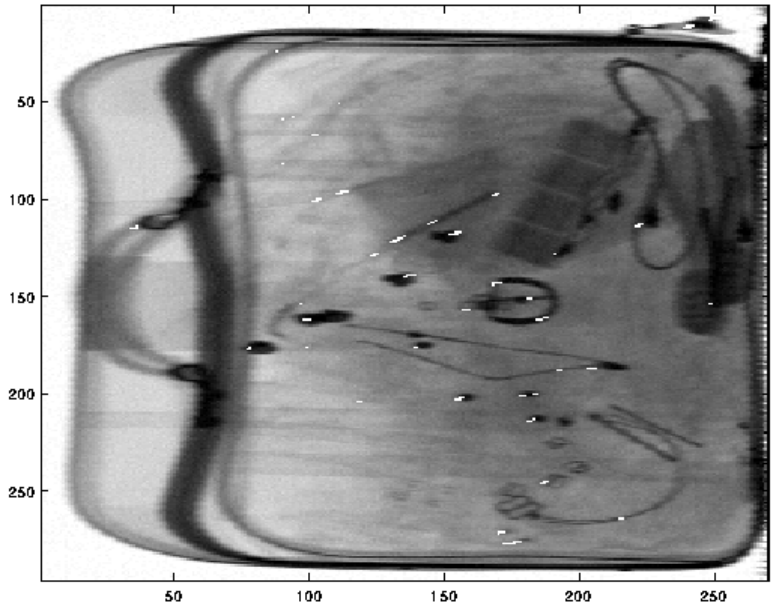


(*c*)

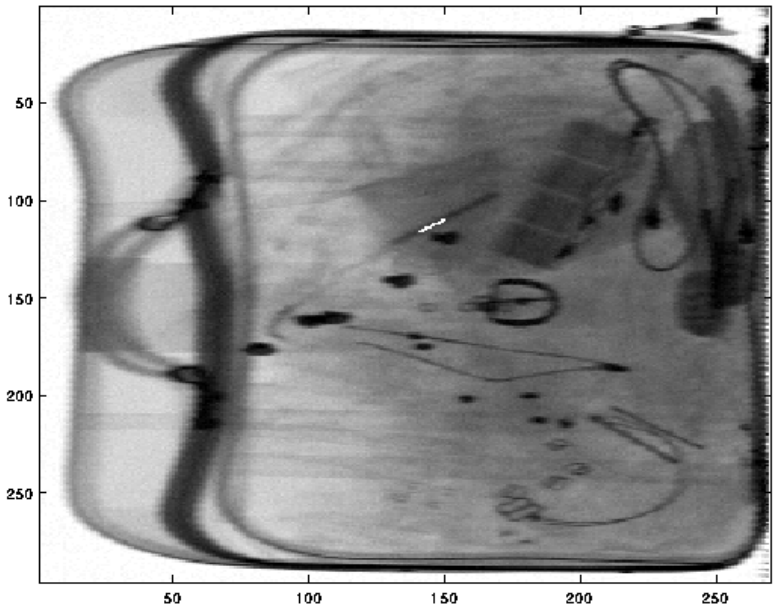


(*d*)

Fig. 5.20. Example of a false negative. The threshold for the end detectors is 0.83. The detonator overlaps with dark background. (*a*) Original image. (*b*) Output using the middle-point template. (*c*) Output using the end-point templates. (*d*) Final result. The detonator was not detected.

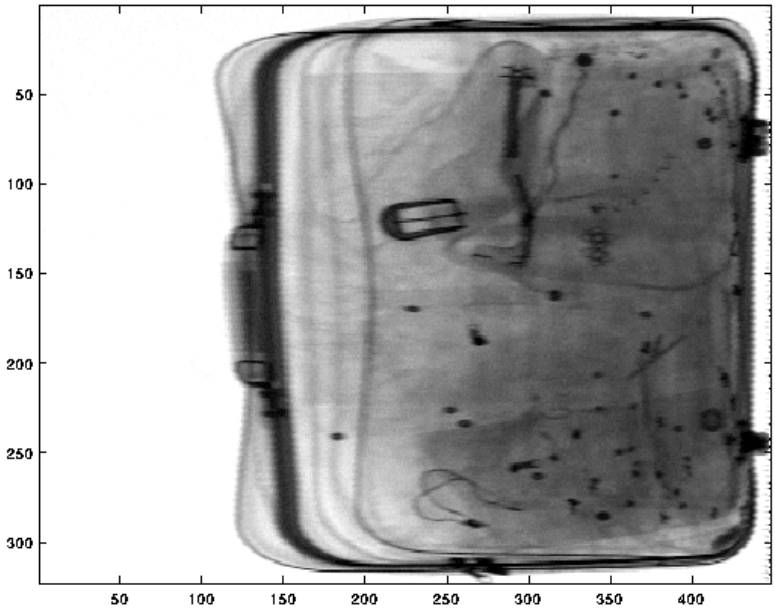


(*a*)

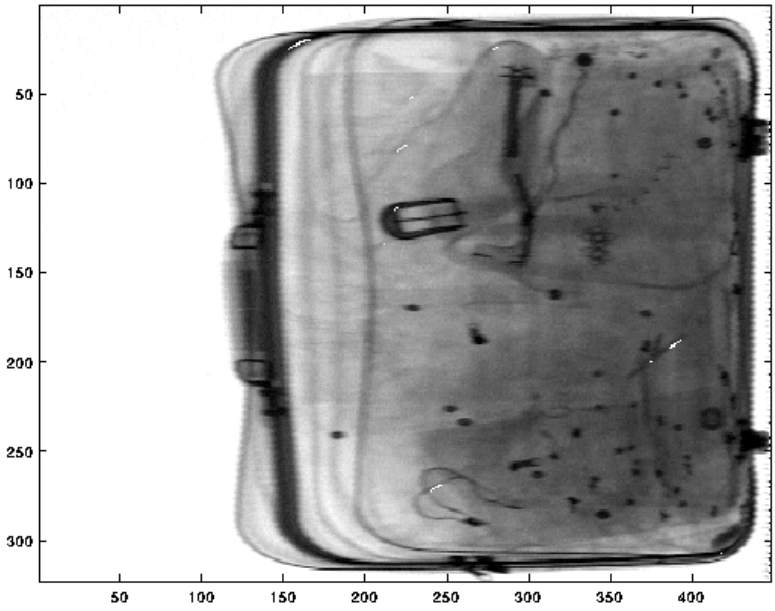


(*b*)

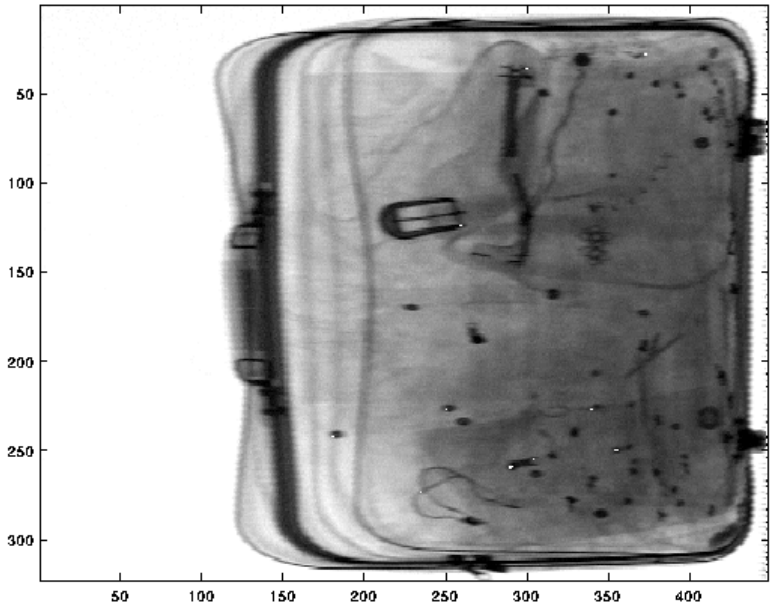
Fig. 5.21. Effect of thresholding on detection ability. The original image is the same as in Fig. 5.20. The threshold for the end detectors is 0.75. (*a*) Output using the same end-point templates as Fig. 5.20. (*b*) Final result. The detonator has been detected, and no false matches are present.


50
100
150
200
250
300
50
100
150
200
250
300
350
400

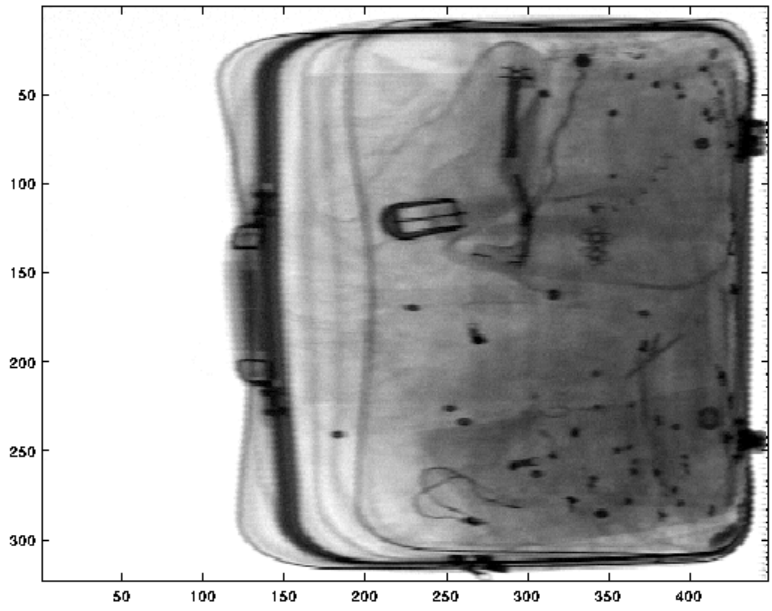
(*a*)


50
100
150
200
250
300
50
100
150
200
250
300
350
400

(*b*)



(*c*)



(*d*)

Fig. 5.22. Second example of a false negative. The luggage was placed at the furthest end of the conveyor. The detonator overlaps a noisy background. (*a*) Original Image. (*b*) Output using the middle-point template. (*c*) Output using the end-point templates. (*d*) Final result. The detonator was not detected.