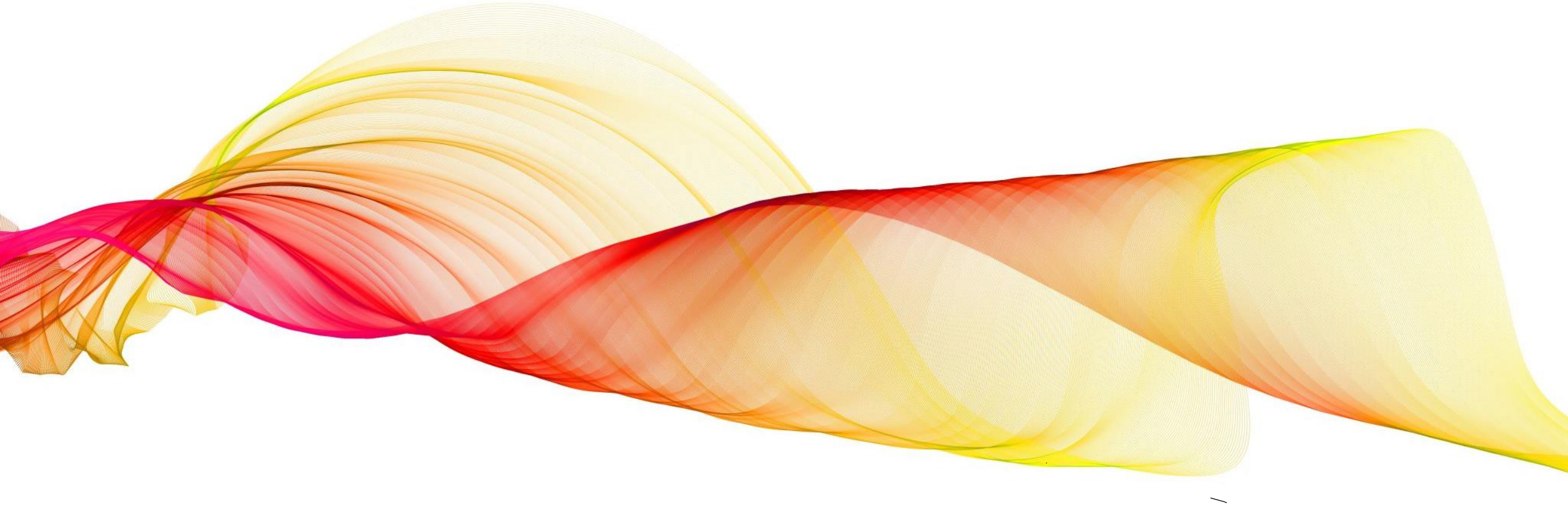
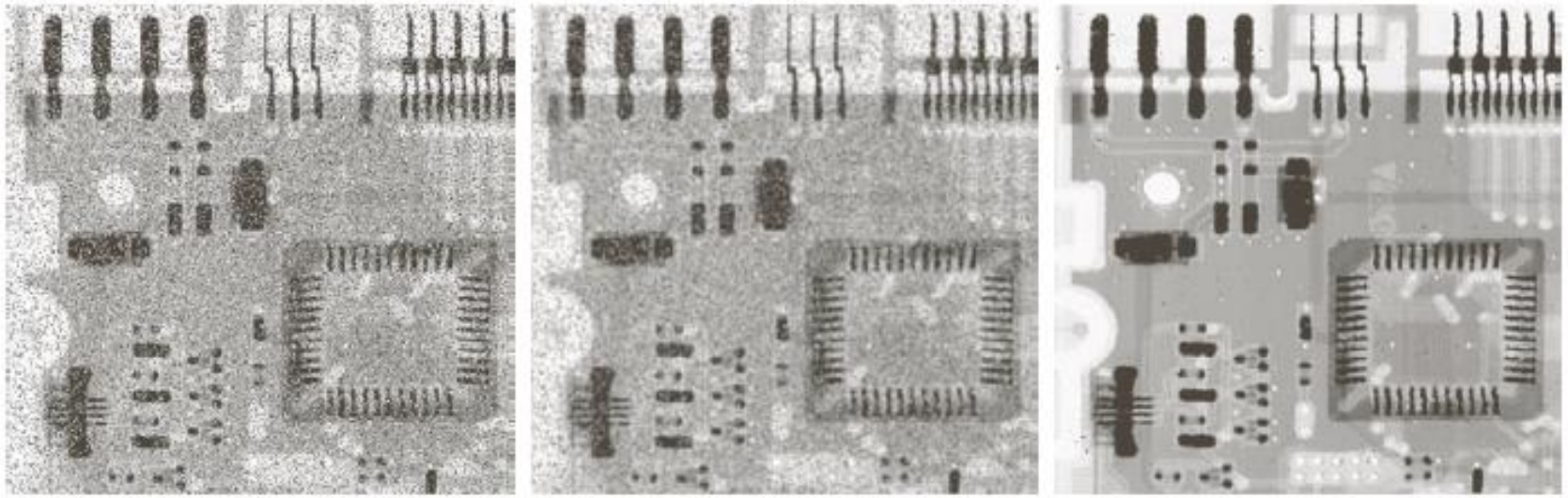
Spatial Filtering

Noise removal



Order-Statistic (Non-linear) filter (median)



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Median filter Algo

1. Define the size of the filter mask
2. Image Padding depending on the size of filter mask
3. Apply the filter mask on input image
4. Arrange the values in the ascending/descending order
5. Pick the median value and assign to pixel location under observation
6. Crop the output image to make it of same size as input image

Median Filtering in OpenCV

```
medianBlur(image, smoothed_image, 5);
```

Inserting Gaussian Noise

```
import cv2
import numpy as np
# Load the image
img = cv2.imread('test_image.jpg')
# Generate random Gaussian noise
mean = 0
stddev = 180
noise = np.zeros(img.shape, np.uint8)
cv2.randn(noise, mean, stddev)
# Add noise to image
noisy_img = cv2.add(img, noise)
# Save noisy image
cv2.imwrite('noisy_img.jpg', noisy_img)
```

151	151	153	152	151
153	150	153	152	151
150	153	152	151	154
153	155	150	154	154
151	151	153	152	152

5	-1	-3	4	0
-10	-1	4	7	-9
-2	0	-6	9	-6
2	5	-10	5	10
6	1	1	1	10

156	150	150	156	151
143	149	157	159	142
148	153	146	160	148
155	160	140	159	164
157	152	154	153	162

156	156	150	150	156	151	151
156	156	150	150	156	151	151
143	143	149	157	159	142	142
148	148	153	146	160	148	148
155	155	160	140	159	164	164
157	157	152	154	153	162	162
157	157	152	154	153	162	162

156	156	150	150	156	151	151
156	156	150	150	156	151	151
143	143	149	157	159	142	142
148	148	153	146	160	148	148
155	155	160	140	159	164	164
157	157	152	154	153	162	162
157	157	152	154	153	162	162

1	1	1
1	1	1
1	1	1

151	151	153	152	151
153	150	153	152	151
150	153	152	151	154
153	155	150	154	154
151	151	153	152	152

151	151	153	152	151
149	150	153	152	150
150	150	153	152	154
153	151	153	154	157
155	153	153	155	160

156	156	150	150	156	151	151
156	156	150	150	156	151	151
143	143	149	157	159	142	142
148	148	153	146	160	148	148
155	155	160	140	159	164	164
157	157	152	154	153	162	162
157	157	152	154	153	162	162

1	2	1
2	4	2
1	2	1

151	151	153	152	151
153	150	153	152	151
150	153	152	151	154
153	155	150	154	154
151	151	153	152	152

152	151	152	153	150
148	150	153	153	148
149	150	151	154	152
154	152	151	155	159
155	153	152	155	160

151	151	153	152	151
153	150	153	152	151
150	153	152	151	154
153	155	150	154	154
151	151	153	152	152

		1		
	0	0		1
0	1		0	1
1	1			
0	1		1	

151	151	255	152	151
153	0	0	152	255
0	255	152	0	255
255	255	150	154	154
0	255	153	255	152

151	151	151	255	152	151	151
151	151	151	255	152	151	151
153	153	0	0	152	255	255
0	0	255	152	0	255	255
255	255	255	150	154	154	154
0	0	255	153	255	152	152
0	0	255	153	255	152	152

151	151	151	255	152	151	151
151	151	151	255	152	151	151
153	153	0	0	152	255	255
0	0	255	152	0	255	255
255	255	255	150	154	154	154
0	0	255	153	255	152	152
0	0	255	153	255	152	152

151	151	153	152	151
153	150	153	152	151
150	153	152	151	154
153	155	150	154	154
151	151	153	152	152

151	151	152	152	152
151	151	152	152	152
153	152	152	152	154
255	153	154	154	154
255	153	255	153	154

151	151	151	255	152	151	151
151	151	151	255	152	151	151
153	153	0	0	152	255	255
0	0	255	152	0	255	255
255	255	255	150	154	154	154
0	0	255	153	255	152	152
0	0	255	153	255	152	152

1	1	1
1	1	1
1	1	1

151	151	153	152	151
153	150	153	152	151
150	153	152	151	154
153	155	150	154	154
151	151	153	152	152

134	140	140	169	174
112	124	124	152	180
147	135	124	141	181
141	163	181	158	170
141	164	209	175	175

151	151	151	255	152	151	151
151	151	151	255	152	151	151
153	153	0	0	152	255	255
0	0	255	152	0	255	255
255	255	255	150	154	154	154
0	0	255	153	255	152	152
0	0	255	153	255	152	152

1	2	1
2	4	2
1	2	1

151	151	153	152	151
153	150	153	152	151
150	153	152	151	154
153	155	150	154	154
151	151	153	152	152

141	142	161	168	170
111	104	104	139	200
124	149	123	124	191
159	197	174	152	169
111	181	197	191	171