Chapter 5

Image Processing

5.1 Introduction

This chapter, mainly inspired from [16], details the main operations of image processing, necessary to extract the useful information from the images taken by the camera.

The applications developed in the frame of this thesis are varied, but most of them roughly require the same sequence of operations. Some treatments are to be applied only for one or another particular application, but we do not make any distinction at this point. The exact sequence of operations needed for each application will be specified in the corresponding chapter (see Part II).

The image processing operations used in this work are subdivided into six principal areas:

1. image acquisition, i.e. the process that yields a visual image (Section 5.2);
2. preprocessing, dealing with basic techniques as subtraction or thresholding, performed on the whole image (Section 5.3);
3. segmentation, the aim of which is to partition an image into objects of interests (Section 5.4);
4. feature computation (e.g. size, shape) suitable for differentiating one type of object from another (Section 5.5);
5. selection, which allows to identify these objects and to select the pertinent one(s), i.e. one or several laser curves (Section 5.6);
6. postprocessing, a series of processes applied on the selected object(s) only, once it (they) is (are) identified (Section 5.7).

We illustrate the different treatments on a typical image represented on Fig. 5.1. It was obtained in a dark room with artificial lighting. The target has been arbitrarily chosen as a tree, and is illuminated by one laser plane. The principles detailed in this chapter are of course valid under natural lighting, for any object and for higher amounts of planes.

Note that in this chapter, we describe some operations which require from the user to specify the value of one or more thresholds or criteria. We would like to draw the attention to the reader that after an empirical learning stage (specific for each application), the—robust—values of those thresholds do not change anymore, so that the image processing algorithms are thus fully automatic. They are implemented in the C++ language.

Fig. 5.1. Typical image.

5.2 Image Acquisition

Visual information is converted to electrical signal by the CCD sensor (see [3] or [16] for a detailed explanation of the CCD camera functioning).

In order to be in a form suitable for computer processing, an image must be digitized both spatially and in amplitude (intensity). Digitization of the ICS coordinates (x_{pix}, y_{pix}) will be referred to as *image sampling*, while amplitude digitization will be called *intensity* or *gray-level quantization*. The latter term is applicable to monochrome images and reflects the fact that these images vary from black to white in shades of gray. The terms *intensity* and *gray level* will be used interchangeably, and will be noted I.

Suppose that a continuous image is sampled uniformly into an array of N_{rows} rows (variable y_{pix}) and N_{col} columns (variable x_{pix}), where each sample is also quantized into N_{lev} levels of intensity. This array, called a *digital image*, may be represented as:

$$I\left(x_{pix}, y_{pix}\right) = \begin{bmatrix} I\left(0,0\right) & \cdots & I\left(0, N_{col}-1\right) \\ \vdots & \ddots & \vdots \\ I\left(N_{rows}-1, 0\right) & \cdots & I\left(N_{rows}-1, N_{col}-1\right) \end{bmatrix} \tag{5.1}$$

where x_{pix} and y_{pix} are now discrete variables. Each element of the array is called a *pixel*.

As detailed in Chapter 4, the CCD cameras we used all had 576 rows and 768 columns, with 256 gray levels, so that an acquired image is a 576-by-768 matrix of elements, the values of which are ranging from 0 to 255.

5.3 Preprocessing

5.3.1 Subtraction

In order to simplify the image processing, two pictures of the scene are actually taken: the first one I_{on} with the laser diode on (as on Fig 5.1), the other I_{off} with the laser diode off (see Fig. 5.2).

Fig. 5.2. Typical image with the laser diode off.

The two images are then subtracted, simply by making the difference between the intensity of each pixel of the two images:

$$I_{sub} = I_{on} - I_{off} \tag{5.2}$$

where I_{sub} is the image resulting from the subtraction. Negative intensity values not being allowed, they will be replaced by 0 if some pixels of I_{off} happen to be brighter than the corresponding ones in I_{on} (e.g. because of the noise affecting the CCD sensor).

The result when applied on the two reference pictures is on Fig. 5.3.

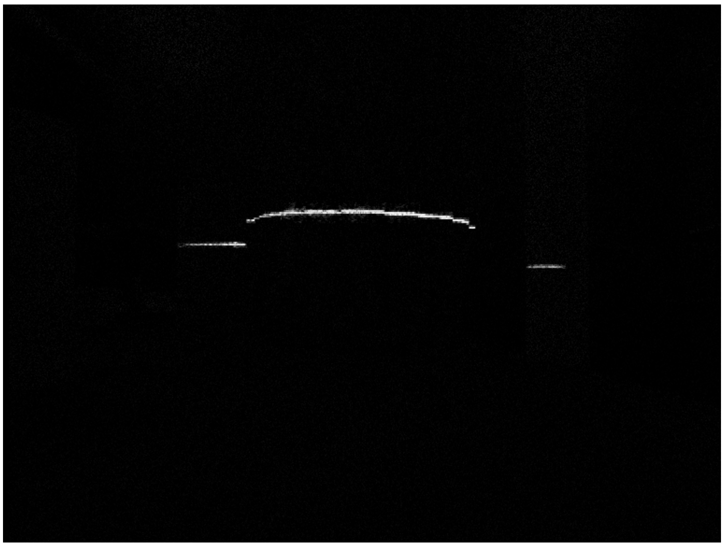

Fig. 5.3. Typical image after subtraction.

After this operation, the pertinent objects (i.e. the laser curve projected on the tree in our example) mostly remain, but some parasite objects will generally also be present, because of e.g. variabilities of the scene illumination and of the camera position between the two captures, quantization errors, noise, ...

Even though the ultimate aim of the present work is to develop ambulatory portable instruments, we exclusively focused on laboratory-made measurements, where the devices as well as the objects were immobile. In real conditions, there would unavoidably be camera and/or object movements between the two acquisitions, inferring an erroneous subtraction. Adequate rotation, translation and scaling of I_{off} could be applied to maximize the correlation with I_{on} and thus overcome the problem. It is however beyond the scope of this thesis.

Note that for illustration purposes, we will not apply subtraction on our example, so that a sufficient amount of clearly visible parasite objects remains on the image.

5.3.2 Thresholding

Image thresholding is one of the principal techniques used by computer vision systems for object detection, whose aim is to separate background pixels from object pixels.

Classically, the transformation implies to set the intensity of background pixels to 0 and to let the object pixels unchanged. The discrimination criterion between the two classes of objects is a separative gray level called *threshold* and noted t. Pixels with their gray level below t will be set to 0, while the others keep their intensity value.

This operation requires the adequate determination of t, which is done by analyzing the image histogram. The latter is simply a vector composed of N_{lev} columns, each element of which is the percentage of image pixels occupying a given gray level.

In our applications, a typical histogram presents a first mode in the low gray levels, followed by a rapid slope until the mid-gray levels. It then remains fairly steady until high intensities, where a second mode is observed (see Fig. 5.4 representing the histogram of our reference image I_{on}; the small peak at level 0 is not mentioned in the analysis here above). This last peak actually contains the major part of the pertinent objects to be extracted.

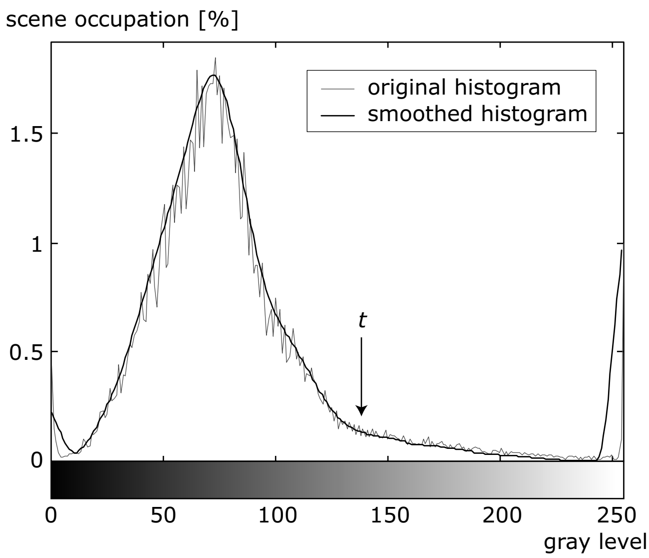

Fig. 5.4. Histogram of the typical image.

To determine the threshold t, we firstly apply a low-pass filter to the original histogram, so that we can work on a smoothed version, allowing us to avoid parasite glitches (see again Fig. 5.4). Starting from gray level 0, we search for the first maximum (around level 75 in our example), then for the highest negative slope (around level 90), and finally considers the threshold t as the gray level at which the slope goes below a "critical" value (empirically determined). In our example, we find $t = 135$, and the original image I_{on} after thresholding—

noted I_t—is represented on Fig. 5.5 (we remind the reader that for illustrative purposes we did not perform image subtraction before the thresholding).

Fig. 5.5. Typical image after thresholding.

This way of doing has the consequence of eliminating the major part of background pixels, while the pertinent objects are kept, together with some parasite bright objects. These have of course to be eliminated. It is to be noted that we opted for security by choosing the "critical" slope hereabove mentioned rather high (in absolute value), so that the calculated threshold is far enough from the high gray levels, i.e. where the laser objects lay. There is thus a limited risk that the darkest pixels of the laser objects are eliminated during thresholding.

5.3.3 Multi-level thresholding

In some applications the image histograms present more than two modes (see Fig. 5.6 representing the histogram of a three-mode image), implying the separation of the image pixels in three classes—one background class and two object classes— and thus the determination of two distinct thresholds t_1 and t_2. They are chosen as the two first local minimums of the histogram curve ($t_1 = 100$ and $t_2 = 215$ in the example of Fig. 5.6).

Note that in practice, the output of a multi-level thresholding is composed of two different images I_{t_1} and I_{t_2}, each of them containing the image pixels of one object class.

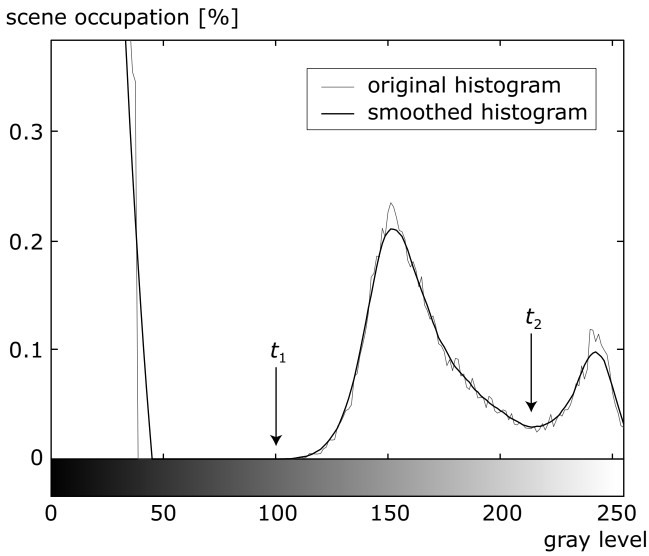

Fig. 5.6. Histogram of the a three-mode image.

5.4 Segmentation

Segmentation is the process that subdivides a sensed scene into its constituent parts or objects. It is one of the most important elements of an automated vision system because it is at this stage of processing that objects are extracted from a scene for subsequent recognition and analysis.

The segmentation algorithm we used is based on the principle of similarity or *region growing*, i.e. grouping pixels into larger regions.

The approach we used is called *pixel aggregation*, where we start with a set of "seed" points (the intensities of which are different from 0) and from these grow regions by appending to each seed point those neighboring pixels that have the similar property of having a gray level greater than 0.

The aim is thus to obtain a list of objects, and for each object the list of its constitutive pixels.

In practice, a mask of 2x2 pixels is moved across the whole image, always from left to right, starting from its top-left corner and moving to the right (at the end of the line, we pass to the next one). Each pixel of the mask is identified by a capital letter (see Fig. 5.7).

We remind the reader that after thresholding, the image pixels are subdivided into two distinct classes: the background and object ones. At each position of the segmentation mask, its bottom-right pixel D is observed (this

A	B
C	D

Fig. 5.7. Mask of 2x2 pixels used for segmentation.

requires to add one row and one column, both virtual and null, at the top and at the left of the image respectively, in order to be able to treat every pixel of the image). If it is:

- an object pixel, it belongs to an object, but which one? To answer that question, we observe the three other pixels of the mask:
  - :: if A, B and C are background pixels, D belongs to a new object which has to be added to the list;
  - :: otherwise, it belongs to an object that has already been created. The current pixel is added to the pixel list of the "oldest" object, i.e. that which has been created the earliest in the segmentation process.
- a background pixel, nothing is done and the mask is moved by one pixel to the right (or to the left of the next line if we have reached the end of a line)

Note that for any position of the mask, if A, B and C are referenced as belonging to different objects (which should not be the case), these are merged into one unique object.

As an illustration, Fig. 5.8(a) shows a 5x5 pixel image, while Fig. 5.8(b) shows that portion after segmentation (we replaced the intensity value by the object number 0_i of the considered pixel).

5.5 Object Feature Determination

Segmentation provides us with a list of objects and their constitutive pixels. From there it is easy to compute a series of features helpful to select the searched object(s), i.e. one or more laser curves. Some of the features we used are:

- the amount of pixels constituting the object;
- the coordinates of the four corners of a (x_{pix}–y_{pix}) rectangle enclosing the object;

(a)

202	208	0	0	0
0	215	212	0	0
0	0	0	0	233
0	0	0	0	226
248	252	254	0	231

(b)

O_1	O_1	-	-	-
-	O_1	O_1	-	-
-	-	-	-	O_2
-	-	-	-	O_2
O_3	O_3	O_3	-	O_2

Fig. 5.8. Result of the segmentation for a 5x5 pixel portion of an image. Figure shows the portion (a) before and (b) after segmentation.

- the height-to-width ratio of this rectangle;
- the distance between the central column of this rectangle and that of the image.

5.6 Object Selection

From the collection of objects obtained after image segmentation, we still need to identify the pertinent one(s) or, equivalently, to eliminate the parasite ones. It will be done by using the features computed at the previous stage.

We now detail the typical three sequences applied to the object list.

5.6.1 Amount of pixels

The first feature used to discriminate parasite objects from the researched one(s) is the amount of pixels composing them. We actually eliminate those composed of less than N_{pix} pixels, this value being determined empirically. See Fig. 5.9 when this operation is applied on the typical image of Fig. 5.5, with $N_{pix} = 10$.

5.6.2 Height-to-width ratio

The second criterion is based on the height-to-width ratio of the rectangle enclosing each object.

Depending on the shape of the searched objects, the value of the "critical" ratio will vary, e.g. if we are searching for mainly horizontal laser curves, we will eliminate all the objects having a large ratio. If necessary, the information

Fig. 5.9. Typical image after elimination of the objects composed of less than 10 pixels.

can be completed by the pixel density in the enclosing rectangle, e.g. if the searched objects are straight lines that might be tilted (such objects would present a high height-to-width ratio but a low pixel density).

Fig. 5.10 shows our reference image after elimination of the objects having a low height-to-width ratio. Straight lines or mostly horizontal curves only remain after this operation.

5.6.3 Distance from central column

A last feature used for the final selection is the distance between the central column of the enclosing rectangles and that of the image. We actually suppose that the user will aim at the center of the target to be measured, and thus only keep the objects the closest to the central column of the image (their amount depends on the amount of laser curves that we are looking for). See Fig. 5.11 to see the result when this criterion is applied to our reference image. The laser curve is now identified!

5.7 Postprocessing

This section deals with treatments applied on the extracted laser curve(s), in order to restore it (them) as close as possible to the original pattern(s),

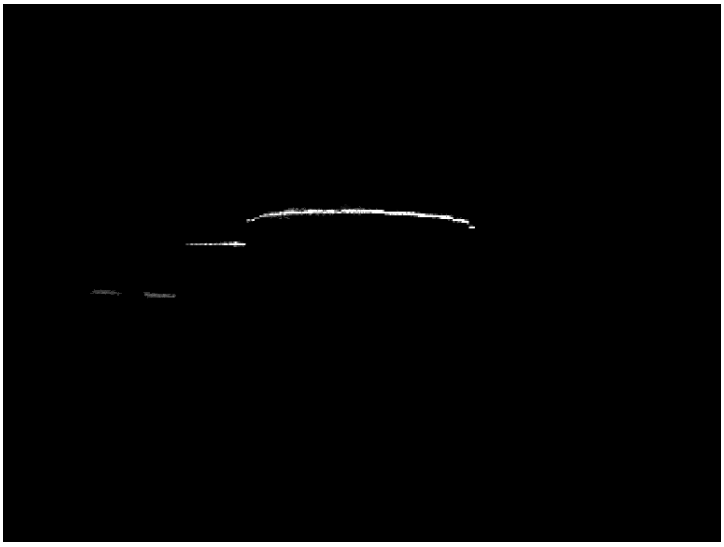

Fig. 5.10. Typical image after elimination of the objects with a low height-to-width ratio.

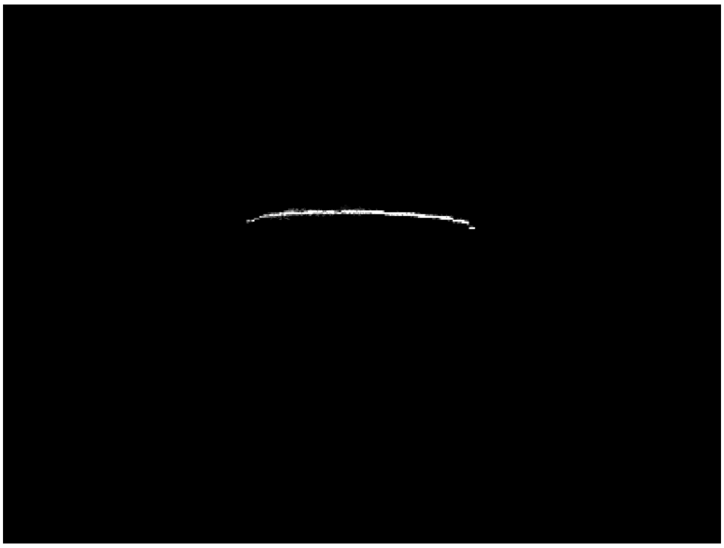

Fig. 5.11. Typical image where only the object the closest from the image's central column is kept.

taking into account the physics of laser, and the geometry of the objects to be measured.

Note that to our opinion, the two following sections (5.7.1 and 5.7.2) are original contributions.

5.7.1 Subpixel location of a laser curve

An extracted laser curve generally presents a thickness of several pixels. In order to refine it, one needs to find its "central thread," i.e. the locus of the "centers" of each transversal cut of the curve.

The intensity profile of a laser curve in its transversal direction is theoretically perfectly Gaussian. It means that for the often encountered situation of Fig. 5.11, where the laser curve is essentially horizontal, the here above mentioned centers are to be evaluated column by column.

In the general case of a random curve (thus not horizontal), the procedure is the same, seen the symmetry of a Gaussian profile. The cut of the laser curve is thus not required to be transversal, and we may also proceed column by column (excepted in the limit case of a vertical curve, not encountered in practice).

A simple but raw method would be to consider the central brightest pixel in each column as the center, but it would lead to absolute errors on the real location going up to 0.5 pixel, because of *image sampling*.

We will rather exploit the overflowing of the laser curve on several pixels to determine its location with subpixel accuracy.

As said above, in theory, the intensity profile of a laser curve in its transversal direction is perfectly Gaussian. On the image it not the case though, because of the effects of:

- *image sampling* modifying the shape of the profile;
- *saturation*: the laser objects being usually very bright, one or more of their central pixels are at the saturation level of the CCD sensor, which obviously spoils the appearance of these objects;
- *DC offset* i.e. the luminance of the target under study, coming from an illumination external to the laser source;
- *thresholding* eliminating the darkest pixels of the laser object.

Fig. 5.12 illustrates these effects.

The appearance of a laser curve on the image is thus eventually very different from the original one, which will unavoidably affect the subpixel determination of its center in each column.

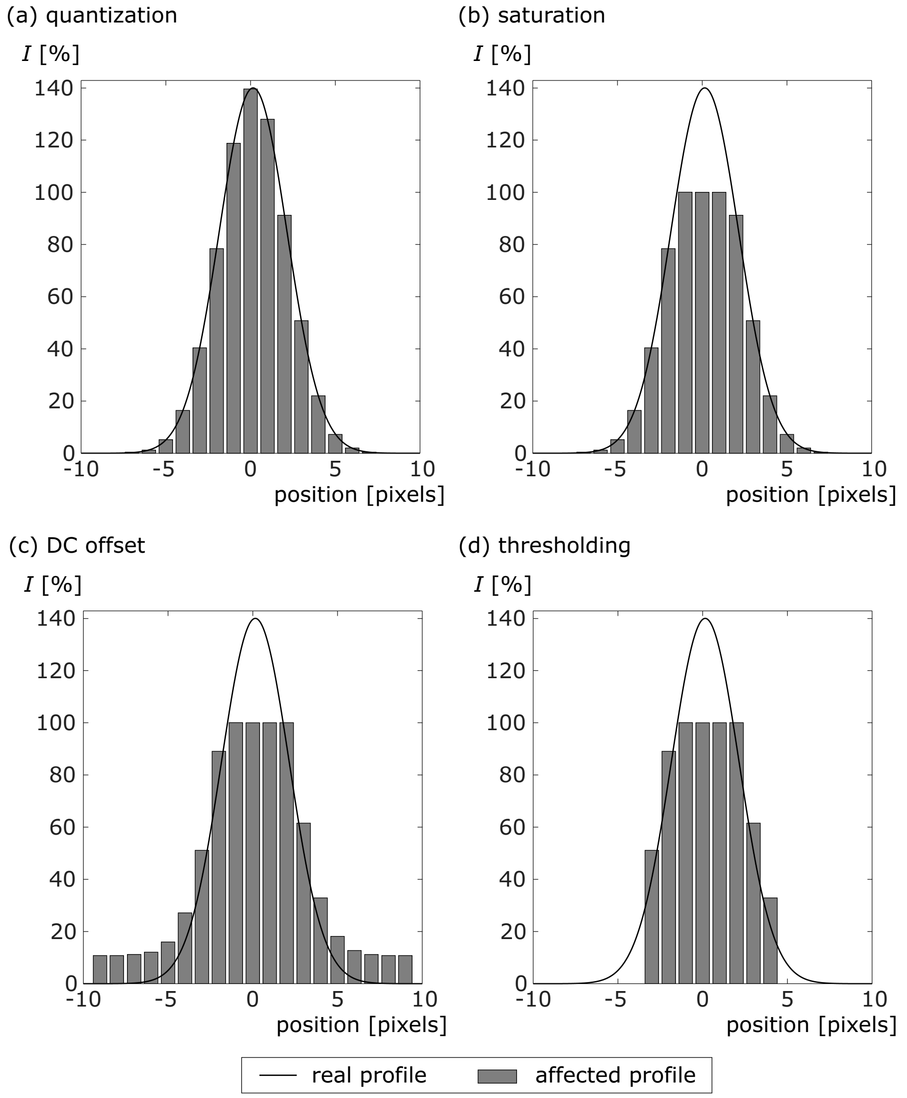

Fig. 5.12. Effects of (a) quantization, (b) saturation, (c) DC offset and (d) thresholding on the Gaussian intensity profile of a laser curve in the transversal direction. I denotes the intensity in the image, in percentage of saturation.

A number of methods can be used for this task. The most usual ones are reported in [25] and compared regarding their robustness against the above-mentioned perturbations. It appears that the *grey scale centroid*, also called *weighted center of gravity* in the literature [26], is the most reliable one.

With this method, the subpixel location of the center y_{subpix} in column x_{pix} is the sum of the ordinates $y_{pix,i}$ of the n pixels belonging to the laser curve in the considered column, weighted by their intensity $I\left(x_{pix}, y_{pix,i}\right)$:

$$y_{subpix}\left(x_{pix}\right) = \frac{\sum_{i=1}^{n} y_{pix,i} I\left(x_{pix}, y_{pix,i}\right)}{\sum_{i=1}^{n} I\left(x_{pix}, y_{pix,i}\right)} \tag{5.3}$$

In order to appreciate the performances of the method, we simulated 5000 Gaussian profiles of various positions ($-0.5 \ldots 0.5$ pixels around a central position) and widths ($3 \ldots 15$ pixels), submitted to image sampling and various levels of thresholding ($70 \ldots 100$), saturation (peak of $500 \ldots 3000$ saturated at 255) and DC offset ($0 \ldots 40$).

For each profile we computed the subpixel location of its center by (5.3), and compared it to its real value. TABLE 5.1 presents statistical figures of the absolute error on the center Δy_{subpix} for the whole of these experiments (last line presents the percentage of profiles for which $|\Delta y_{subpix}|$ is below 0.15 pixel; σ denotes a standard deviation). They prove the efficacy of the method, and the improvement provided by a subpixel location method: without it, the maximum absolute error would be of 0.5 pixel, compared to 0.15 pixel with the *grey scale centroid* method.

TABLE 5.1. Statistical Figures of the Absolute Error on the Subpixel Location

max $\lvert\Delta y_{subpix}\rvert$	0.18 pixel
mean Δy_{subpix}	0.005 pixel
$\sigma_{\Delta y_{subpix}}$	0.016 pixel
$\lvert\Delta y_{subpix}\rvert < 0.15$ pixel	95%

Note that with this technique, the subpixel location of each pixel of the laser curve in the x_{pix} direction will not be possible (e.g. to locate the extremities of the curve). The maximum absolute error in this direction is thus of 0.5 pixel.

Fig. 5.13 shows a close-up on the laser curve of Fig. 5.11 after subpixel location. Remark that, as it is an image representation of the curve, its subpixel character cannot be appreciated (each vertical step corresponds to a 1-pixel variation in the y_{pix} direction).

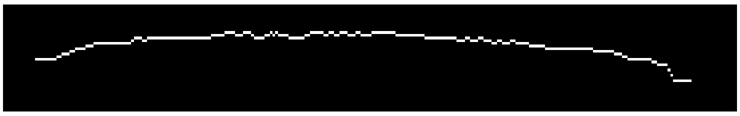

Fig. 5.13. Laser curve on the reference image after subpixel location.

5.7.2 Speckle filtering of a laser curve

Another possible postprocessing operation is the filtering of speckle, that could be present on the laser pattern. See Section 7.3 to see in which conditions it might happen. In our applications, speckle arose only in one case, described in Chapter 12.

To be exact, this operation—when needed—must be applied before the subpixel location. It is presented in a different order in this chapter though, for a reason that will appear at the end of this section.

The proposed approach is a low-pass filtering of the laser curve: the useful information of the curve is essentially concentrated in its low-to-middle frequency components, and laser speckle is a high frequency noise. This type of filtering has the advantage of eliminating a large amount of noise, at the expense of a reduction in a small amount of useful signal [27].

The applied filter is a two-dimensional finite impulse response (2-D FIR) filter, and consists in applying on each pixel of the laser curve a mask which is a combination of the intensity of the surrounding pixels. We chose Gaussian weighted coefficients for this combination [28].

This being set, one still needs to decide about the size of the 2-D filter window, which will define the quantity of neighboring pixels to take into account when applying the filter on one particular pixel.

This size was determined empirically, and is of 7x57 pixels. The smallest side of the window is in the longitudinal direction of the laser curve, in order to preserve a sufficient resolution in that direction. See Fig. 5.14 showing a 3-D representation of the filter window.

In order to verify the efficacy of such a filter, we simulated sinusoidal laser curves of various frequencies (200–500 pixels of wavelength) and amplitudes (40–100 pixels peak to peak) affected by various levels of speckle (by mean of a Matlab® function). For each curve we applied the low-pass filtering of the speckle and finally calculated the subpixel location of the center, in each column. This explains why the latter operation has been presented earlier in this chapter, despite the fact that it is to be applied after the speckle filtering.

Fig. 5.15(a)–(d) shows an example of sinusoidal laser curve in the different stages of processing, and Fig. 5.15(e) the subpixel location of the same curve

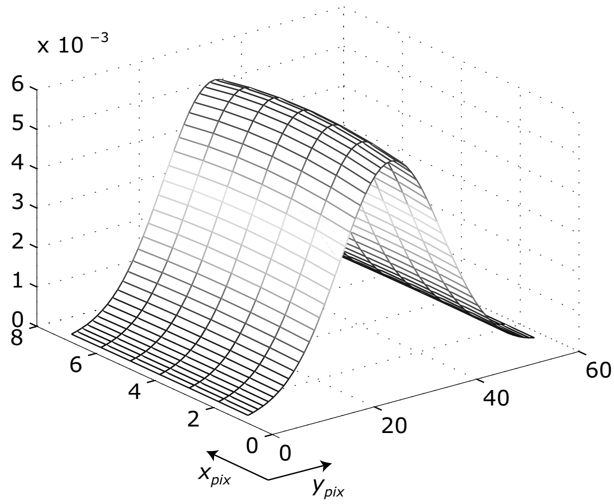

Fig. 5.14. 3-D representation of the low-pass Gaussian 2-D FIR filter.

if the speckle is not filtered. The improvement brought by the FIR filtering is visually obvious on certain portions of the curve.

Fig. 5.16 presents the error in the location of the example curve (i.e. the difference between the original laser curve, before the introduction of the speckle, and the treated one). The distinction is made between the filtered signal and the unfiltered one, of poorer quality.

TABLE 5.2 presents statistical results of the absolute error on the center Δy_{subpix} for the whole of the simulated sinusoidal curves (last line presents the percentage of profiles for which $|\Delta y_{subpix}|$ is below 1 pixel; σ denotes a standard deviation). They prove that an absolute error of ± 1 pixel on the location of the laser curve is achievable.

TABLE 5.2. Statistical Figures of the Absolute Error after Speckle Filtering

max $\lvert\Delta y_{subpix}\rvert$	1.35 pixel
mean Δy_{subpix}	0.02 pixel
$\sigma_{\Delta y_{subpix}}$	0.11 pixel
$\lvert\Delta y_{subpix}\rvert < 1$ pixel	94.5%

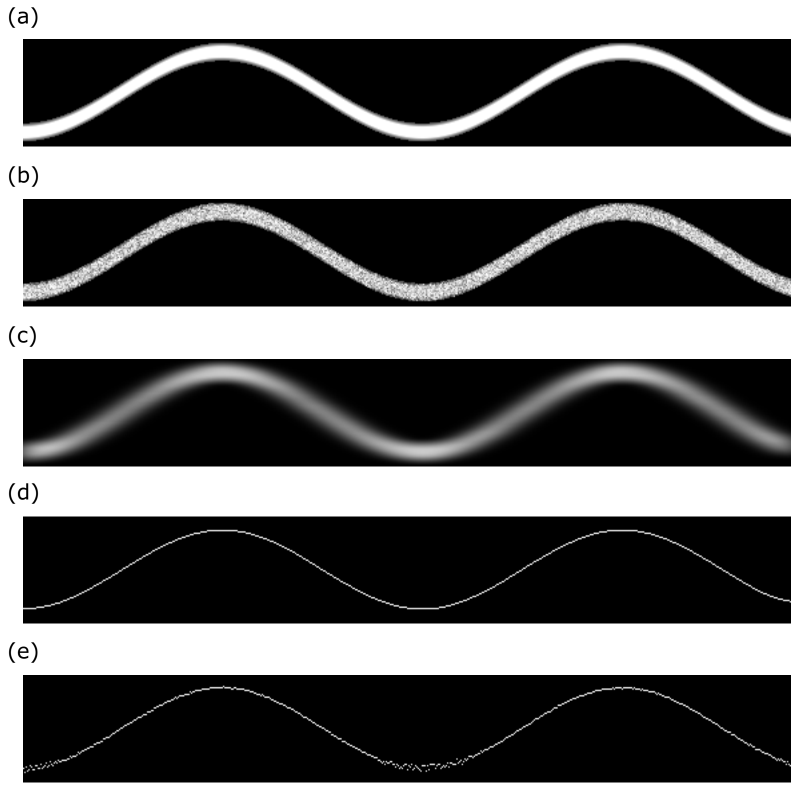

Fig. 5.15. Example of speckle filtering on a sinusoidal laser curve (amplitude of 60 pixels peak to peak, period of 300 pixels). Images go from (a) the original curve to (b) that affected by speckle, (c) the filtered one and (d) the curve after subpixel location. For comparison, (e) shows the subpixel location of the unfiltered curve.

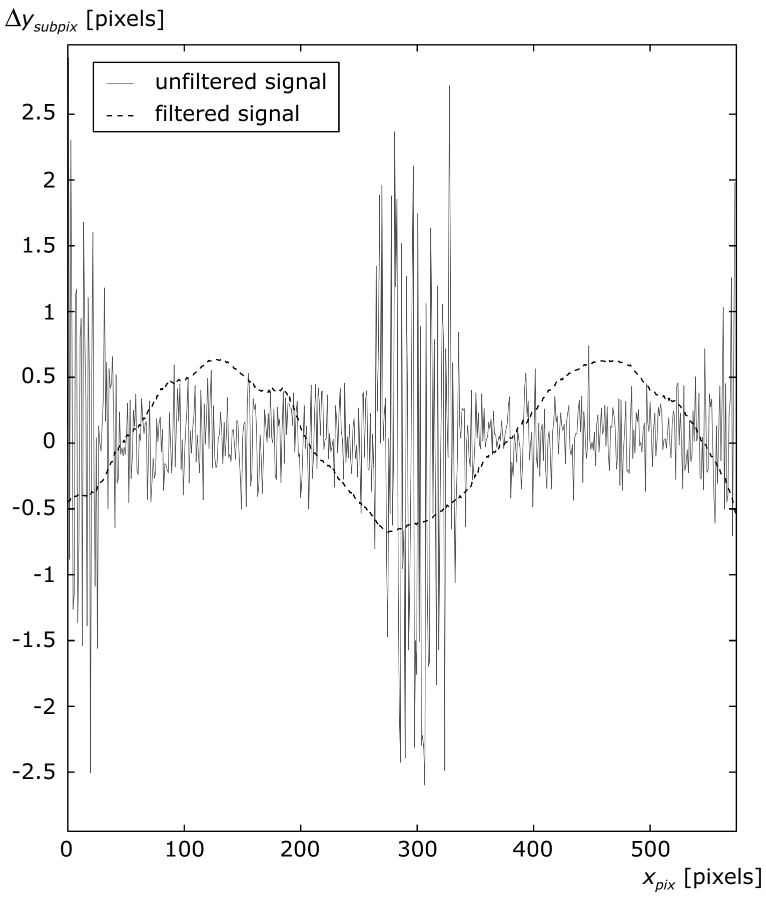

Fig. 5.16. Error on the subpixel location of the sinusoidal laser curve of Fig. 5.15 affected by speckle. Both filtered and unfiltered error curves are presented.

5.7.3 Curve smoothing

In some cases, it may be required to smooth the laser curve(s). It will typically be necessary when the piece to be measured presents some irregularities on its surface (e.g. bark protuberances on trees or holes in walls) that would affect the accuracy on the quantity to be determined (e.g. the radius of a tree or the angle between two walls). Smoothing allows us to get rid of those perturbations.

For each point of the laser curve to be treated, the two extremes excepted, we calculate the mean ordinate $y_{smoothpix}$ of the considered point and its two surrounding pixels in the x_{pix} direction:

$$y_{smoothpix}\left(x_{pix}\right) = \frac{1}{3}\sum_{i=-1}^{1} y_{subpix}\left(x_{pix}+i\right) \tag{5.4}$$

where y_{subpix} is the subpixel location of the laser curve in the appropriate column.

Our experience shows that about ten applications of this mask are required to obtain a sufficiently smooth result, but this method offers the advantage by its smoothness of not denaturing the original aspect of the laser curve.

Fig. 5.17 shows the laser curve of our reference image before and after smoothing. The improvement is obvious.

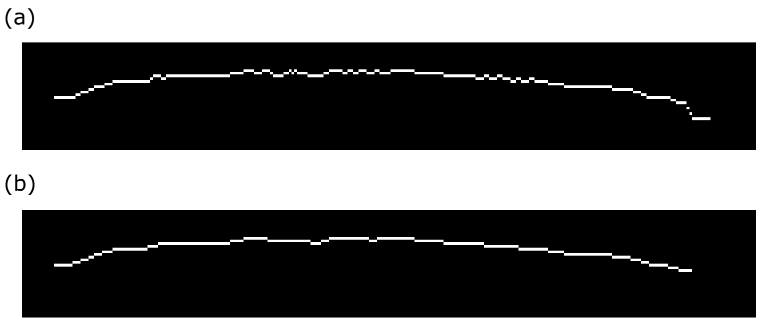

Fig. 5.17. Laser curve on the reference image (a) before and (b) after smoothing.

5.7.4 Correction of the distortion

The final operation to be performed on the extracted objects is the correction of the distortion induced by the camera lens. The procedure has been described in Section 2.6 and has to be applied on each pixel of each laser object.

After this last treatment, we thus have at our disposal a series of laser objects, and for each of them, a list of the undistorted ICS coordinates of their constitutive pixels.