Input LUTs would be valuable also for nonlinear point operations if the 8-bit input values were mapped to higher precision output values, e.g., 16-bit integers or 32-bit floating point numbers. Then rounding errors could be avoided. At the same time, the gray levels could be converted into a calibrated signal, e.g., a temperature for an infrared camera. Unfortunately, such generalized LUTs are not yet implemented in hardware. However, it is easy to realize them in software.

In contrast to the input LUT, the output LUT is a much more widely used tool, since it does not change the stored image. With LUT operations, we can also convert a gray value image into a *pseudo-color image*. Again, this technique is common even with the simplest frame grabber boards (Fig. 8.1). Not much additional hardware is needed. Three digital-analog converters are used for the primary colors red, green, and blue. Each channel has its own LUT with 256 entries for an 8-bit display. In this way, we can map each individual gray value q to any color by assigning a color triple to the corresponding LUT addresses $r(q)$, $g(q)$, and $b(q)$. Formally, this is a *vector* point operation

$$\mathbf{P}(q) = \begin{bmatrix} r(q) \\ g(q) \\ b(q) \end{bmatrix}. \tag{8.8}$$

When all three point functions $r(q)$, $g(q)$, and $b(q)$ are identical, a gray tone will be displayed. If two of the point functions are zero, the image will appear in the remaining color.

8.2.3 Interactive Gray Value Evaluation

Homogeneous point operators implemented via look-up tables are a very useful tool for inspecting images. Since the look-up table operations work in real-time, images can be manipulated interactively. If only the output look-up table is changed, the original image content remains unchanged. Here, we demonstrate typical tasks.

Evaluating and Optimizing Illumination. With the naked eye, we can hardly estimate the homogeneity of an illuminated area as demonstrated in Fig. 8.4a,b. A histogram reveals the gray scale distribution but not its spatial variation (Fig. 8.4c,d). Therefore, a histogram is not of much help for optimizing the illumination interactively. We need to mark gray scales such that absolute gray levels become perceivable for the human eye. If the radiance distribution is continuous, it is sufficient to use equidensities. This technique uses a staircase type of homogeneous

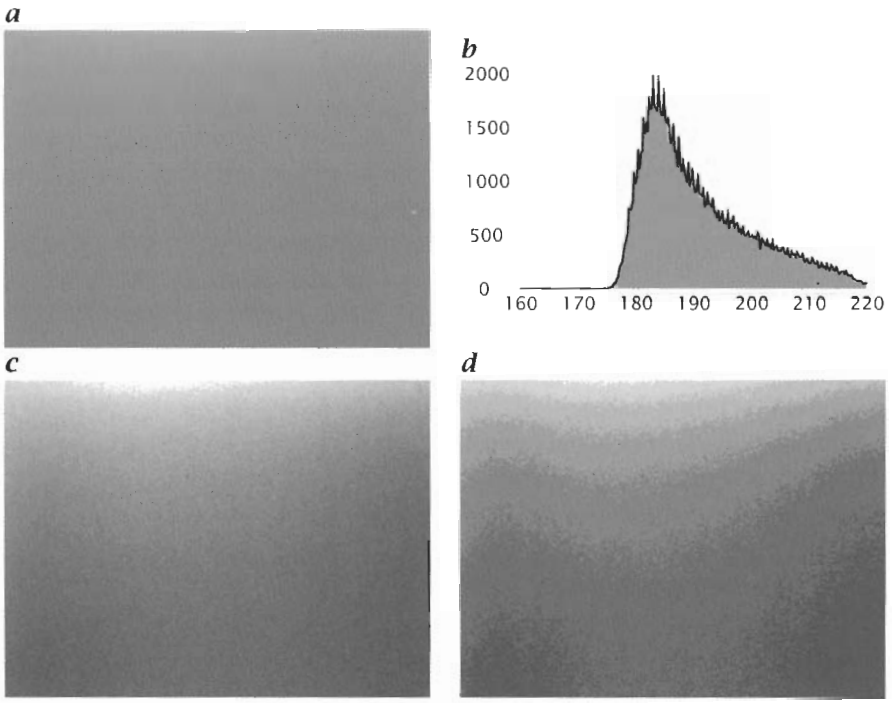

Figure 8.4: ***a*** *The irradiance is gradually decreasing from the top to the bottom, which is almost not recognized by the eye. The gray scale of this floating-point image computed by averaging over 100 images ranges from 160 to 200.* ***b*** *Histogram of* ***a****;* ***c*** *and* ***d*** *(contrast enhanced, gray scale 184.0–200.0): Edges artificially produced by a staircase LUT with a step height of 1.0 and 2.0 make contours of constant irradiance easily visible (**exercise 8.1**).*

point operation by mapping a certain range of gray scales onto one. This point operation is achieved by zeroing the p least significant bits with a logical *and* operation:

$$q' = P(q) = q \wedge \overline{(2^p - 1)}, \tag{8.9}$$

where $\wedge$ denotes the logical (bitwise) *and* and overlining denotes *negation*. This point operation limits the resolution to $Q - p$ bits and, thus, 2^{Q-p} quantization levels. Now, the jump between the remaining quantization levels is large enough to be perceived by the eye and we see contour lines of equal absolute gray scale in the image (Fig. 8.4). We can now try to homogenize the illumination by making the distance between the contour lines as large as possible.

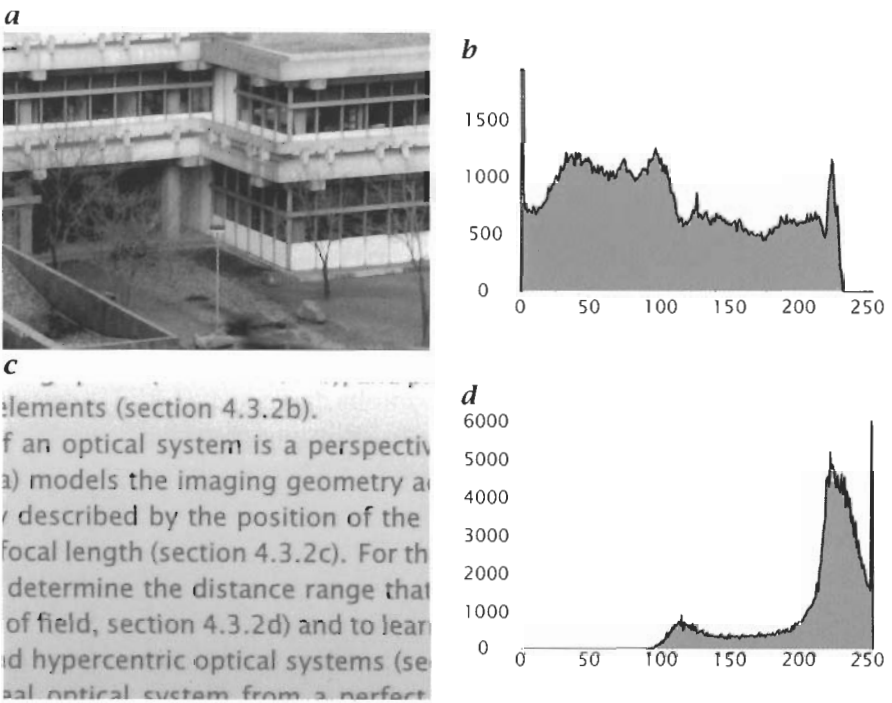

Figure 8.5: *Detection of underflow and overflow in digitized images by histograms (**exercise 8.2**); **a** image with underflow and **b** its histogram; **c** image with overflow and **d** its histogram.*

Another way to mark absolute gray values is the so-called *pseudocolor* display that has already be discussed in Sect. 8.2.2. With this technique, a gray level q is mapped onto an RGB triple for display. Since color is much better recognized by the eye, it helps reveal absolute gray levels. Try **exercise 8.1** to play with equidensities and pseudocolor display to optimize illumination.

Detection of Underflow and Overflow. Under- and overflows of the gray values of a digitized image often go unnoticed and cause a serious bias in further processing, for instance for mean gray values of objects or the center of gravity of an object. In most cases, such areas cannot be detected directly. They may only become apparent in textured areas when the texture is bleached out. Over and underflow are detected easily in histograms by strong peaks at the minimum and/or maximum gray values (Fig. 8.5). With pseudocolor mapping, the few lowest and highest gray values could be displayed, for example, in blue and red, respec-

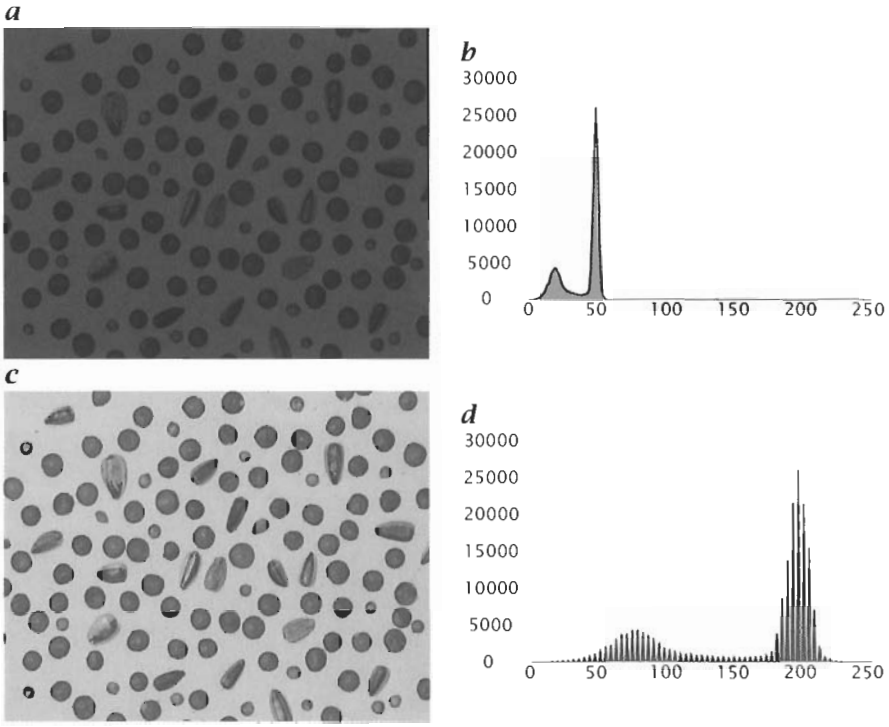

Figure 8.6: *Contrast enhancement (**exercise 8.3**); **a** underexposed image and **b** its histogram; **c** interactively contrast enhanced image and **d** its histogram.*

tively. Then, gray values dangerously close to the limits immediately pop out of the image and can be avoided by correcting the illumination lens aperture or gain of the video input circuit of the frame grabber.

Contrast enhancement. Because of poor illumination conditions, it often happens that images are underexposed. Then, the image is too dark and of low contrast (Fig. 8.6a). The histogram (Fig. 8.6b) shows that the image contains only a small range of gray values at low gray values. The appearance of the image improves considerably if we apply a point operation which maps a small grayscale range to the full contrast range (for example with this operation: $q' = 4q$ for $q < 64$, and $q' = 255$ for $q \geq 64$) (Fig. 8.6c). We only improve the appearance of the image but not the image *quality* itself. The histogram shows that the gray value resolution is still the same (Fig. 8.6d).

The image quality can be improved. The best way is to increase the object irradiance by using a more powerful light source or a better design of the illumination setup. If this is not possible, we can still increase the

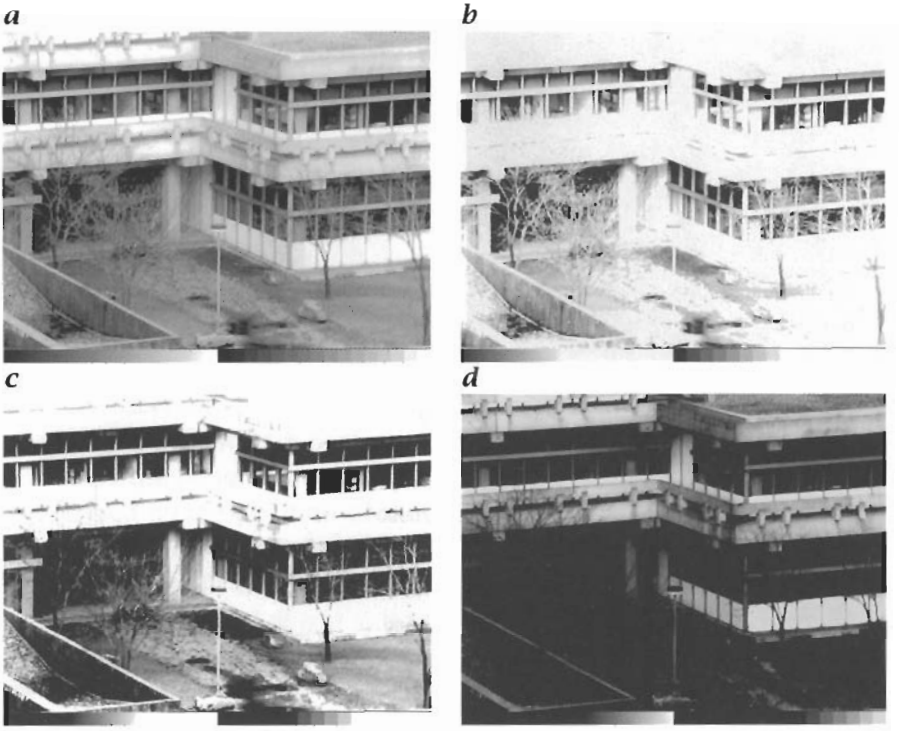

Figure 8.7: b–d *Contrast stretching of the image shown in* ***a****. The stretched range can be read from the transformation of the gray scale wedge at the bottom of the image (****exercise 8.3****).*

gain of the analog video amplifier. All modern image processing boards include an amplifier whose gain and offset can be set by software (see Figs. 8.1 and 8.2). By increasing the gain, the brightness and resolution of the image improve, but at the expense of an increased noise level.

Contrast Stretching. It is often of interest to analyze faint irradiance differences which are beyond the resolution of the human visual system or the display equipment used. This is especially the case if images are printed. In order to observe faint differences, we stretch a small gray scale range of interest to the full range available. All gray values outside this range are set to the minimum or maximum value. This operation requires that the gray values of the object of interest fall into the range selected for contrast stretching. An example of contrast stretching is shown in Fig. 8.7a,b. The wedge at the bottom of the images, ranging from 0 to 255, directly shows which part of the gray scale range is contrast enhanced.

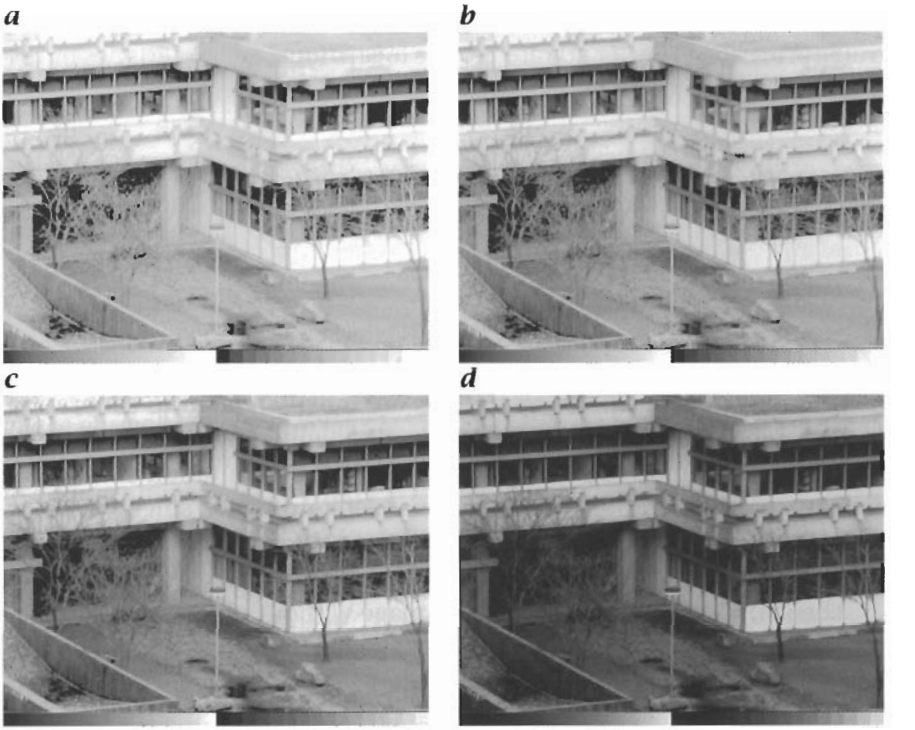

Figure 8.8: *Presentation of an image with different gamma values:* ***a*** *0.5,* ***b*** *0.7,* ***c*** *1.0, and* ***d*** *2.0 (**exercise 8.3**).*

Range Compression. In comparison to the human visual system, a digital image has a considerably smaller dynamical range. If a minimum resolution of 10% is demanded, the gray values must not be lower than 10. Therefore, the maximum dynamical range in an 8-bit image is only $255/10 \approx 25$. The low contrast range of digital images makes them appear of low quality when high-contrast scenes are encountered. Either the bright parts are bleached or no details can be recognized in the dark parts.

The dynamical range can be increased by a transform that was introduced in Sect. 2.2.6 as the *gamma transform*. This nonlinear homogeneous point operation has the form

$$q' = \frac{255}{255^{\gamma}} q^{\gamma}. \tag{8.10}$$

The factors in (8.10) are chosen such that a range of $[0, 255]$ is mapped onto itself. This transformation allows a larger dynamic range to be recognized at the cost of resolution in the bright parts of the image.

The dark parts become brighter and show more details. This contrast transformation is better adapted to the logarithmic characteristics of the human visual system. An image presented with different gamma factors is shown in Fig. 8.8.

8.3 Inhomogeneous Point Operations

Homogeneous point operations are only a subclass of point operators. In general, a point operation depends also on the position of the pixel in the image. Such an operation is called an *inhomogeneous point operation*. Inhomogeneous point operations are mostly related to calibration procedures. Generally, the computation of an inhomogeneous point operation is much more time consuming than the computation of a homogeneous point operation. We cannot use look-up tables since the point operation depends on the pixel position and we are forced to calculate the function for each pixel.

The subtraction of a background image without objects or illumination is a simple example of an inhomogeneous point operation which is written as:

$$G'_{mn} = P_{mn}(G_{mn}) = G_{mn} - B_{mn}, \tag{8.11}$$

where B_{mn} is the background image.

8.3.1 Image Averaging

One of the simplest inhomogeneous point operations is *image averaging*. There are a number of imaging sensors available which show a considerable noise level. Prominent examples include *thermal imaging* (Sect. 6.4.1) and all sensor types such as slow-scan CCD imagers or image amplifiers where only a limited number of photons are collected.

Figure 8.9a shows the temperature differences at the water surface of a wind-wave facility cooled at 1.8 m/s wind speed by evaporation. Because of a substantial noise level, the small temperature fluctuations can hardly be detected. Taking the mean over several images significantly reduces the noise level (Fig. 8.9b).

The error of the mean (Sect. 3.3.4) taken from N samples is given by

$$\sigma^2_{\langle g \rangle} \approx \frac{1}{(N-1)}\sigma^2_g = \frac{1}{N(N-1)}\sum_{n=0}^{N}(g - \langle g \rangle)^2. \tag{8.12}$$

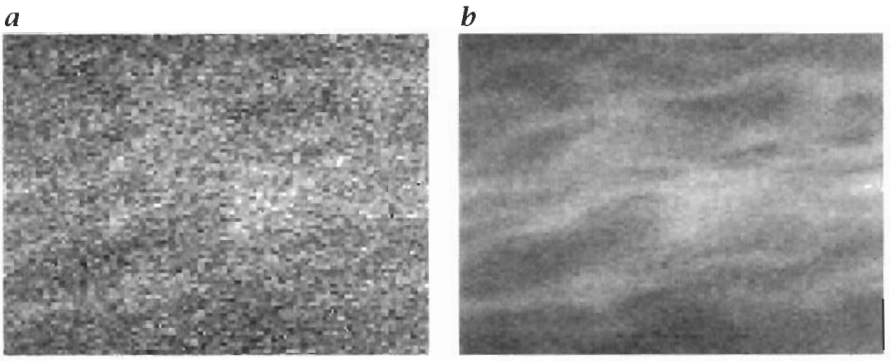

Figure 8.9: *Noise reduction by image averaging (**exercise 8.4**): **a** single thermal image of small temperature fluctuations on the water surface cooled by evaporation; **b** same, averaged over 16 images; the full gray value range corresponds to a temperature range of 1.1 K.*

If we take the average of N images, the noise level is reduced by $1/\sqrt{N}$ compared to a single image. Taking the mean over 16 images thus reduces the noise level by a factor of four. Equation (8.12) is only valid, however, if the standard deviation σ_g is significantly larger than the standard deviation related to the quantization (see Sect. 7.9).

8.3.2 Correction of Inhomogeneous Illumination

Every real-world application has to contend with *uneven illumination* of the observed scene. Even if we spend a lot of effort optimizing the illumination setup, it is still very hard to obtain a perfectly even object irradiance. A nasty problem is caused by small dust particles in the optical path, especially on the glass window close to the CCD sensor. Because of the distance of the window from the imager, these particles — if they are not too large — are blurred to such an extent that they are not directly visible. But they still absorb some light and thus cause a drop in the illumination level in a small area. These effects are not easily visible in a scene with high contrast and many details, but become very apparent in the case of a uniform background (Fig. 8.4a and b). CCD sensors also show an uneven sensitivity of the individual photoreceptors which adds to the nonuniformity of the image. These distortions severely limit the quality of the images. These effects make it more difficult to separate an object from the background, and introduce systematic errors for subsequent image processing steps.

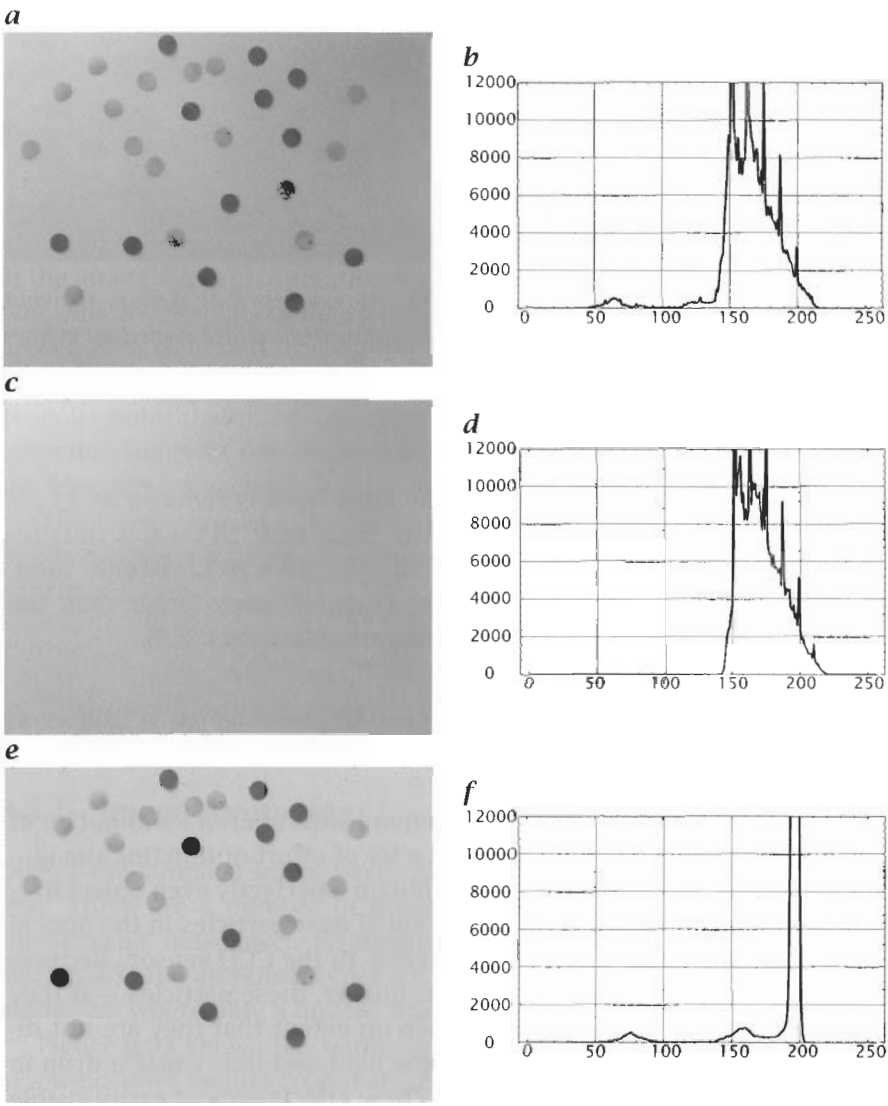

Figure 8.10: *Correction of uneven illumination with an inhomogeneous point operation (**exercise 8.5**): **a** original image and **b** its histogram; **c** background image and **d** its histogram; **e** division of the image by the background image and **f** its histogram.*

Nevertheless, it is possible to correct these effects if we know the nature of the distortion and can take suitable reference images. In the following, we study two cases. In the first, we assume that the gray value in the image is a product of the inhomogeneous irradiance and the reflectivity or transmissivity of the object. Furthermore, we assume that we can take a reference image without absorbing objects or with an object of constant reflectivity. A reference image can also be computed, when small objects are randomly distributed in the image. Then, it is sufficient to compute the average image from many images with the objects. The inhomogeneous illumination can then be corrected by dividing the image by the reference image:

$$G'_{mn} = c \cdot G_{mn} / R_{mn}. \tag{8.13}$$

The constant c is required to represent the normalized image with integer numbers again. If the objects absorb light, the constant c is normally chosen to be close to the maximum integer value. Figure 8.10e demonstrates that an effective suppression of inhomogeneous illumination is possible using this simple method.

The simple ratio imaging described above is not applicable if also a non-zero inhomogeneous background has to be corrected for, as caused, for instance, by the fixed pattern noise of a CCD sensor. In this case, two reference images are required. First, we take a background image $\boldsymbol{B}$ without any illumination. Second, we take a reference image $\boldsymbol{R}$ with full illumination but without absorbing objects as in the first case study. Then, a normalized image corrected for both the fixed pattern noise and inhomogeneous sensitivity is given by

$$\boldsymbol{G'} = c \frac{\boldsymbol{G} - \boldsymbol{B}}{\boldsymbol{R} - \boldsymbol{B}}. \tag{8.14}$$

This technique is known as a two-point calibration and requires, of course, a linear signal.

8.3.3 Radiometric Calibration

Many image measuring tasks require an absolute radiometric calibration of the measured irradiance at the image plane. Once such a calibration is obtained, we can infer the radiance of the objects from the irradiance in the image. One obvious example is thermography. Here, the temperature of the emitted object is determined from its radiance using Planck's equations (Section 6.4.1).

Here, we will give a practical calibration procedure for ambient temperatures. Because of the nonlinear relation between radiance and temperature, a simple two-point calibration with linear interpolation is not