Chapter 7

Preprocessing and Enhancement

Before features are extracted or images are presented to a human operator, their quality sometimes has to be improved. Such 'enhancement' may have dfferent goals, such as sharpening the image, increasing contrast, or reducting noise. Often one wants to achieve several such goals in combination. Enhancement often takes the form of an image normalisation, where one makes sure that the same levels of contrast and the same range of intensity levels result, even if this is not the case in the input images. This protects further processing against adverse effects of variable illumination levels, due e.g. to changing wheather conditions or different times of day.

7.1 I - Fourier power spectra of images

Usually, nearby pixels in an image have similar grey levels. This correlated nature of the data already suggests that the power spectra of such images will mainly contain low frequencies. Looking at the Fourier transform, its spectrum is seen to corroborate this suggestion. As an example we show such spectrum in fig. 7.1. The *dc* component (i.e. $u = v = 0$ or $k = l = 0$ in the notations of chapter 5) is found at the center of that image. Notice how most of the power is clustered around *dc*, indeed. If we only retain the lower frequencies, the reconstructed image looks like fig. 7.2. As can be seen, the result is very similar to the original image. Nevertheless, the object boundaries are quite unsharp now.

Since these boundaries – so-called *edges* – usually contain the essential information about object shape, higher frequencies cannot simply be discarded. The connection between edges and high frequencies is further illustrated by applying a high-pass filter, as shown in fig. 7.3. Unfortunately, it is not only edge information that tends to occupy the higher frequency regions of the spectrum. Whereas grey levels have a high degree of correlation with their neighbouring values, noise in neighbouring pixels often is uncorrelated. It can be shown that this implies that the noise has a flat spectrum. Consequently, the signal to noise ratio at higher frequencies can be rather low. Here lies the heart of the problem. The edge information, often regarded the most crucial aspect of the data, is found submerged in noise.

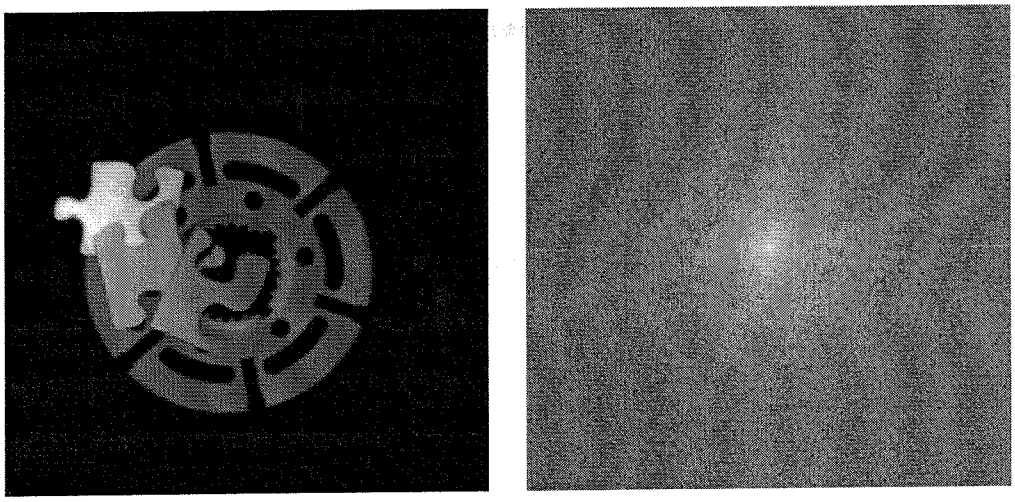

Figure 7.1: *Image (left) with its Fourier spectrum (on the right and rescaled for reasons of visibility).*

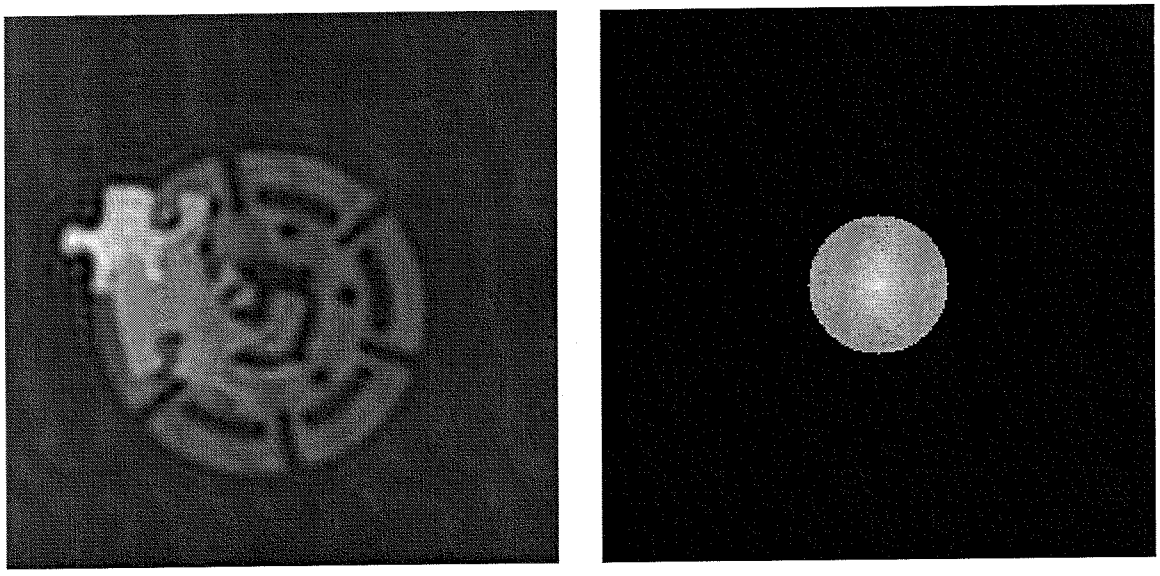

Figure 7.2: *Eliminating all higher frequencies in the Fourier spectrum (right image, black parts are removed) yields a reconstructed image which is blurred (left).*

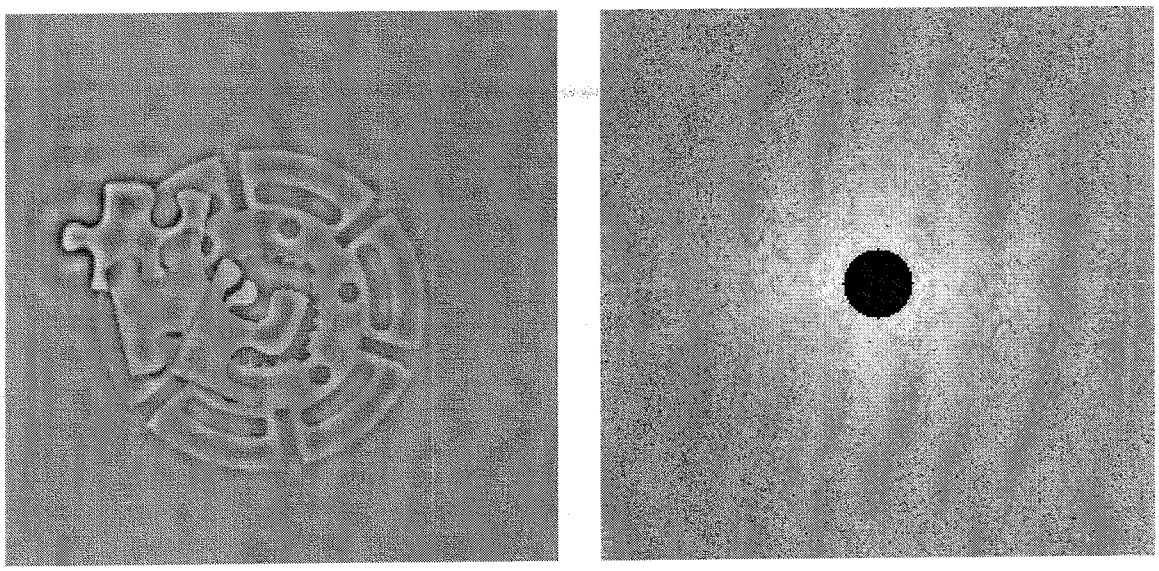

Figure 7.3: *Eliminating all lower frequencies in the Fourier spectrum (right image, black parts are removed), the edges stand out when the image is reconstructed (left).*

A first, very important class of filters are linear and shift-invariant. As shown in chapter 5 these filters amount to convolution. Their importance lies in a combination of a deep theoretical insight in their effect and the fact that they can be implemented rather efficiently. The practical use of such filter is as follows. Consider a mask, usually square and of small dimensions (e.g. 3×3 or 5×5), as shown in fig. 7.4. The mask is shifted over the image. At each position the mask parameters are multiplied with the grey level of the pixel they cover. The sum of these products is assigned to the central pixel as its new grey level. Although such masks are not necessarily square and the output is not always assigned to the central pixel, this is typically the case. Depending on the choice for the parameters c_{ij}, the effect can be very different. Several examples follow. Large masks, which would be computationally expensive, can often be implemented more efficiently as a multiplication in the Fourier domain (cfr. the convolution theorem in chapter 5).

However sophisticated, linear filters can never resolve the problem of edge information and noise being intertwined, since they cannot disentangle the contributions from edges and noise at a given frequency. Their effect on these contributions will be identical, i.e. a multiplication with the corresponding gain factor specified by their modulation transfer function. It should not come as a surprise then that several non-linear techniques are proposed in this chapter.

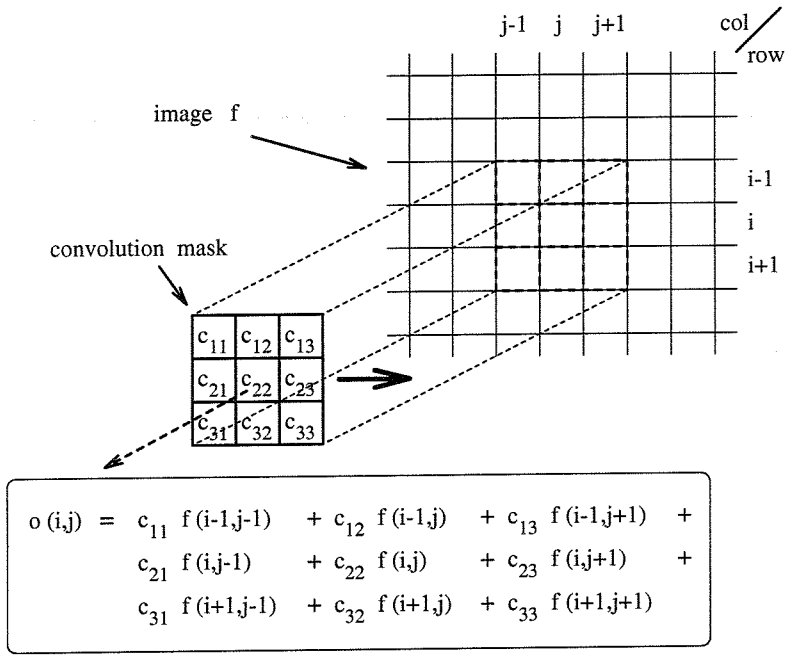

Figure 7.4: *Convolution masks are shifted over the image. At each position they yield a new value ($o(i,j)$), obtained as a linear combination of the underlying image intensities. The coefficients to be used are given by the mask.*

7.2 I - Contrast enhancement and deblurring

It might happen that the dynamic range of the camera (i.e. the range of possible intensities) is not effectively used for the representation of an image, e.g. when it has been under- or overexposed. Two techniques for increasing contrast will be considered. One is a global method (*histogram equalization*), the other works locally (*unsharp masking*).

The image may also be blurred, due to suboptimal camera parameter settings, atmospheric haze, motion, etc. Depending on the type of blur, different deblurring strategies can be followed. First, in the case of Gaussian blur (e.g. unsharp image acquisition), *unsharp masking* is useful as well. In the case of motion blur, dedicated approaches exist that take the pecularities of *motion blur* into account. Finally, *inverse filtering* is described as a general method, with *Wiener filters* as its optimal implementation.

Blurring typically causes contrast over edges to deteriorate. Hence the issues of deblurring and contrast enhancement are related.

7.2.1 Histogram equalisation

Suppose that the image intensities i fall in the interval $I = [0, i_{max}]$. Further suppose that the original histogram (i.e. the grey level probability density function) is a continuous function $p(i)$. For an inappropriately exposed picture, or for a picture where a large and interesting part contains intensities that are all very close to each other, the majority of

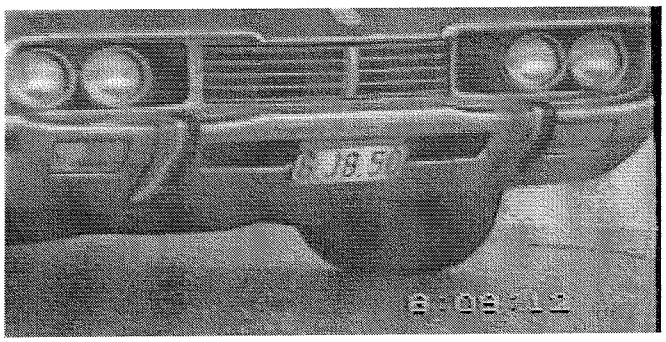

Figure 7.5: *Image with low contrast in regions of interest.*

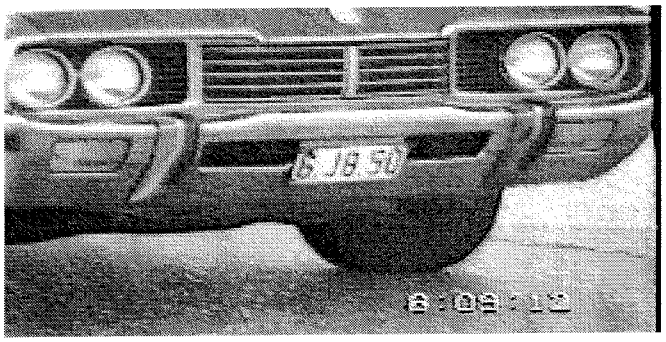

Figure 7.6: *Car image of fig. 7.5 after histogram equalisation. Contrast has markedly been increased in the license plate area.*

pixels will be found in a relatively small portion of the intensity range I. The histogram will have a peaked outlook. The idea underlying histogram equalisation is to produce a new histogram, with a more flat outlook instead. This explains the different names for this operation: *histogram flattening* or *histogram equalization*. For a flattened histogram, the whole range of possible intensities is used. As an example, figure 7.5 shows an image with rather low contrast in several regions of interest, such as the license plate. Figure 7.6 shows the result of histogram equalisation. Contrast has improved substantially at several places, including the license plate area.

Histogram equalisation carries out one of many possible maps from I onto itself. Pixels are given new intensities following some specific transition, as shown in fig. 7.7. Notice that the map is monotonously increasing. Unless the map is the identity, the new image will have a different probability density for its intensities, i.e. a different histogram. For one such map the outcome will be a flat histogram. The problem therefore is to find this particular map. This search is simplified by looking at the cumulative probability

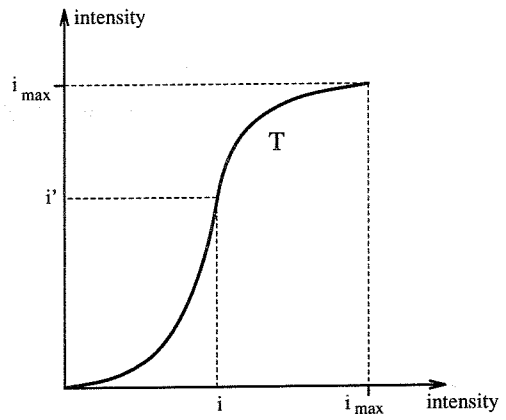

Figure 7.7: *A map with I, the intensity interval, both as its domain and range. A pixel with intensity i is given intensity $i' = T(i)$ instead.*

distributions, i.e.

$$C(i) = \int_0^i p(v)dv.$$

$C(i)$ maps the interval I onto the unit interval, such that $C(0) = 0$ and $C(i_{max}) = 1$. For a flat histogram (i.e. a probability density function which is constant), the cumulative probability distribution will be a straight line between these points. An image with a flat histogram therefore ought to have a cumulative probability distribution value of $1/i_{max}$ for each grey level i. The actual distribution will deviate from this target distribution in that a different proportion of pixels is found in the grey level intervals $[0, i]$. This difference can be eliminated by mapping a grey level i in the original image to a new grey level i' such that

$$\frac{i'}{i_{max}} = \int_0^i p(v)dv \ ,$$

and therefore

$$i' = i_{max}C(i) \ .$$

This grey level transformation forces the resulting histogram to be flat. Indeed, the new probability density function is found as

$$p' = p\frac{di}{di'} = p\frac{1}{p\ i_{max}} = \frac{1}{i_{max}},$$

which is the desired constant value.

As to the effect of histogram equalization, grey level intervals with many pixels packed together (high slope of the cumulative probability distribution) are expanded, and similarly, intervals with few pixels are compressed. The cumulative probability distribution for the car image is shown in fig. 7.8. The histogram of the original image and the flattened histogram are shown in fig. 7.9 at the left and right sides, resp. As can be seen, the resulting histogram is not really flat. This is due to discretisation effects. Discontinuous jumps

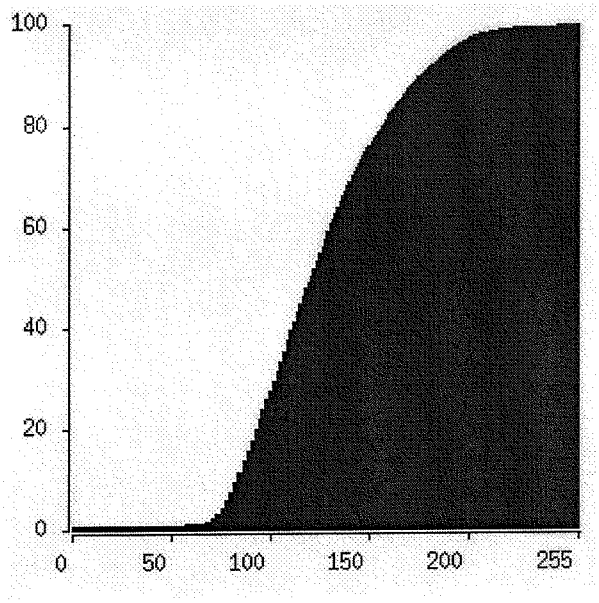

Figure 7.8: *Cumulative probability distribution (in %) for the car image.*

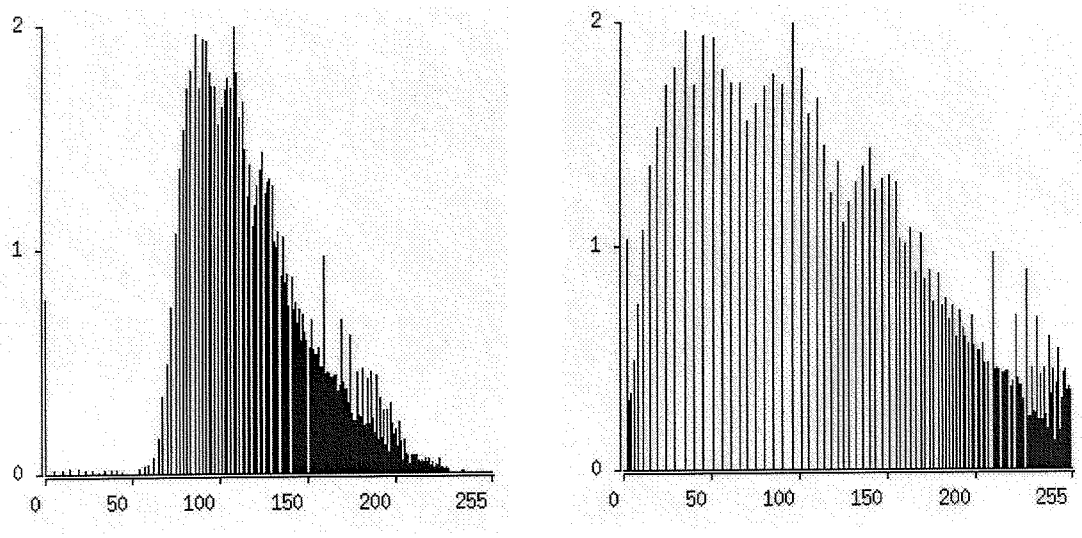

Figure 7.9: *Original histogram (left) and flattened result (right).*

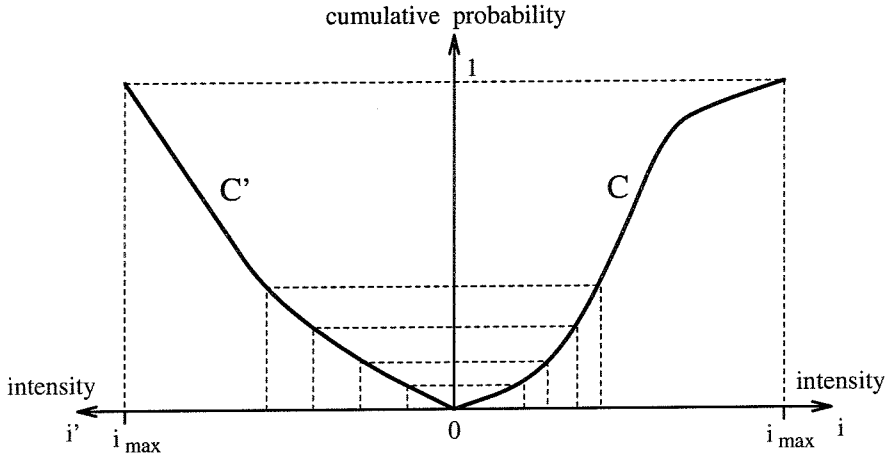

Figure 7.10: *Histogram transformations correspond to intensity maps that are found by equating the cumulative probability distributions.*

in the discretised cumulative probability distribution (inherently coupled to the use of a histogram as the discrete approximation to the probability density function), lead to similar jumps in the equalisation map. As a result some intensities will be skipped. In fact, such phenomenon can already be observed in the original histogram. The reason for this effect is an earlier logarithmic intensity transformation by the frame grabber that took the image. This step was carried out to improve contrast already, but further improvement is clearly possible via the histogram equalisation technique. This technique has the advantage of using image information, whereas the logarithmic transformation was carried out regardless of image content.

Other histogram transformations follow the same principle. The grey level transformation $i' = T(i)$ that yields a probability density p' instead of the original probability density p is obtained by making sure that i is mapped onto i' such that

$$C'(i') = \int_0^{i'} p'(w)dw = \int_0^{i} p(v)dv = C(i).$$

where $C'(i')$ and $C(i)$ are the prescribed and original cumulative probability distributions. Again, the prescribed cumulative probability distribution $C'(i')$ dictates how many pixels (in proportional terms) should obtain intensities up to i'. In order to reach that quota, pixels in the original image have to be combined up to some grey level i such that $C'(i') = C(i)$. The map is thus found to be

$$i' = C'^{-1}(C(i)).$$

A sketch of the situation can be found in fig. 7.10.

7.2.2 Unsharp masking

This technique enhances local contrast and can effectively undo Gaussian blur. Moreover, it is simple to implement. The principle is e.g. used in photocopiers. Suppose a sharp

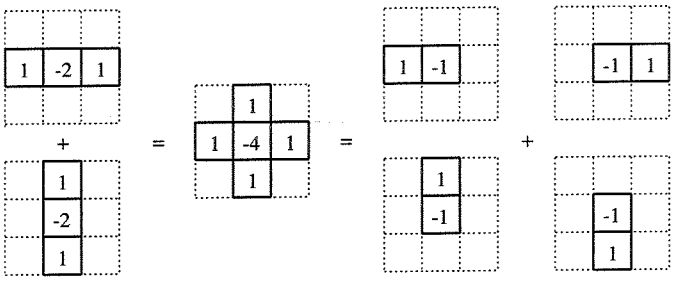

Figure 7.11: *The finite difference approximation of the Laplacian operator shows that it is equivalent to summing the difference in grey level between the central pixel and its surround.*

image $f_o(x, y)$ was blurred by a Gaussian filter, resulting in the blurred image $f_b(x, y)$. The process of gradual Gaussian blurring, governed by the so-called heat or diffusion equation

$$\frac{\partial f}{\partial t} = c\left(\nabla^2 f\right).$$

with the diffusion coefficient c a positive constant and

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

will start at $t = 0$ with f_o as initial condition. The finite difference approximation of the Laplacian operator ∇^2 (fig. 7.11) shows that the grey level f is changed by a portion of the average grey level difference with the surrounding pixels. The sign is such that the grey level of each pixel approaches the average value of its surround. Thus, variations decline gradually, starting at the high frequency end of the spectrum. After some time Δt, the diffusion process will have reached exactly the degree of blur observed in f_b. Hence, as a first approximation

$$\frac{f_b - f_o}{\Delta t} = c\nabla^2 f_b \ , \tag{7.1}$$

and thus

$$f_o = f_b - (c\Delta t)\nabla^2 f_b \ .$$

In practice, $c\Delta t$ will be replaced by an appropriately chosen constant. Note that this unsharp masking technique can be interpreted as running a diffusion process backwards. Indeed, upon swapping f_b and f_o in eq. (7.1), f_o can be interpreted as the outcome of a diffusion process with f_b as initial conditions and *negative* diffusion coefficient $-c$.

Fig. 7.12 shows a sketch of this deblurring technique. Part (a) shows a cross-section of the intensity profile of an input image, in this case taken at the position of an edge. In order to carry out the above deblurring scheme, the Laplacian can be approximated as the difference between the image and its smoothed version. The dashed line shows the cross-section of the smoothed input. A more detailed discussion on the Laplacian filter is to follow at the end of this chapter. Inlet (b) shows the difference approximation to the Laplacian (with the sign reversed, the actual Laplacian is found as the smoothed minus the

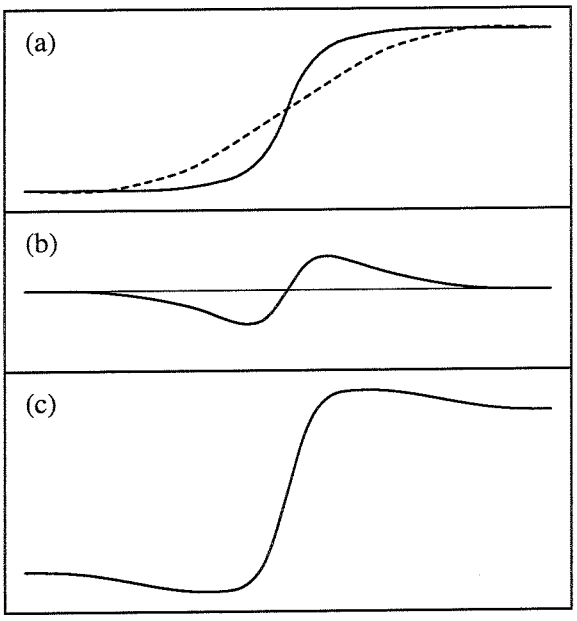

Figure 7.12: *Sketch of unsharp masking: (a) input (bold line) and its blurred version (dashed line), (b) approximation for output of the Laplacian operator, (c) result of summing the input in (a) and the Laplacian output in (b), the edge can be seen to be sharpened.*

original image). Unsharp masking then consists of summing the original profile and the curve in (b) multiplied by an appropriately chosen constant. Inlet (c) of the figure shows the sum if the factor is chosen to be 1. In that case, the result is found as twice the original minus the smoothed profile. As can be seen, the edge profile in (c) is steeper, giving a sharper impression. Also are there under- and overshoots flanking the edge. This should not be a problem, as they further increase the impression of image sharpness. In fact, the human visual system itself produces such effects (see the discussion on Mach bands given in chapter ??). In fact, the choice for the contant replacing $c\Delta t$ was such that the inverse diffusion process overcompensated the blur, i.e. the inverse process was allowed to evolve further than the process producing the blur. Finally, unsharp masking is illustrated on a real image. Fig. 7.13 shows an unsharp image. The result of unsharp masking in shown in fig. 7.14. As can be seen, the result is quite impressive considering the simplicity of the technique.

7.2.3 Inverse filtering

The effects of a linear blurring process with modulation transfer function $B(u, v)$ would be undone if the image could be passed through a linear filter with a modulation transfer function $B'(u, v)$ such that

$$B(u, v)\ B'(u, v) = 1.$$

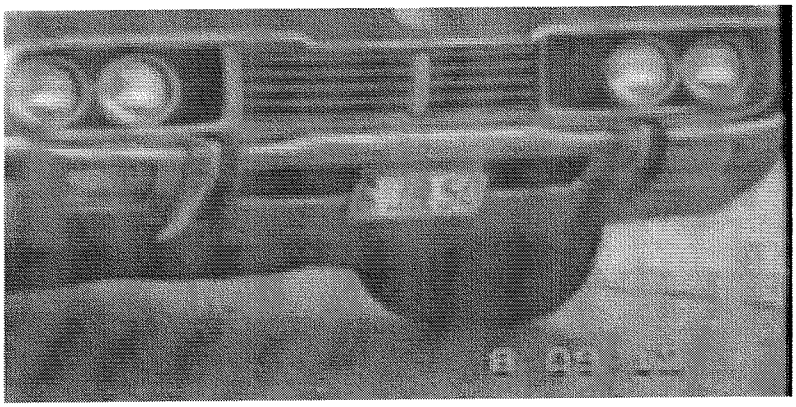

Figure 7.13: *Blurred car image.*

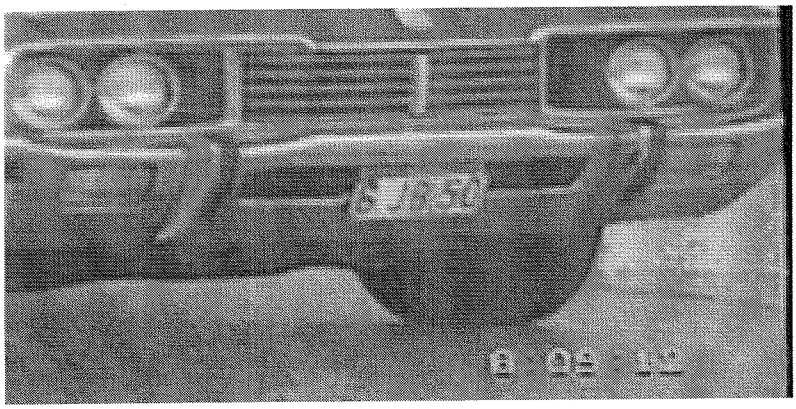

Figure 7.14: *Deblurred image for the blurred car image of fig. 7.14 using unsharp masking.*



Figure 7.15: *Blurred TV presenter image.*

An immediate problem is that frequencies that have been totally suppressed, i.e. for which $B(u,v) = 0$, cannot be recovered. Another serious problem is noise, certainly if it was introduced after the blurring. Strongly attenuated frequencies tend to become submerged in the noise then, and attempts to recover them by amplification boost the noise accordingly.

As an intuitive approach to the problem, consider an inverse filter with modulation transfer function

$$B'(u,v) = \frac{1}{B(u,v) + C}.$$

As long as $B(u,v) \gg C$ the modulation transfer function closely approximates the ideal MTF $1/B(u,v)$. For small values of $B(u,v)$ it shows a kind of saturation, i.e. it will tend to $1/C$ when $B(u,v) \to 0$. The effect is illustrated on the image of fig. 7.15. This image is the result of a blurring operation with a gaussian filter having a standard deviation of 16 pixels. Applying an inverse filter of the above type yields the result shown in fig. 7.16. This inverse filter will still let pass frequencies that have totally been filtered out by $B(u,v)$, i.e. frequencies with $B(u,v) = 0$. In that case we add pure noise! Indeed, the inverse filtered image in fig. 7.16 clearly shows spurious high frequencies. This caveat can be eliminated by making the following (equally intuitive) choice:

$$B'(u,v) = \frac{B(u,v)}{B^2(u,v) + C}$$

with C a constant. As long as $B(u,v)$ is sufficiently large to suppress the influence of C, we have $B'(u,v) \simeq 1/B(u,v)$. Once $B(u,v)$ is getting too small however, i.e. when the



Figure 7.16: *Result of inverse filtering using a modulation transfer function of the type* $1/(B + ct.)$.

corresponding frequency has been filtered out almost completely from the input image, then $B'(u, v)$ also decays to zero. In fact, this choice already comes close to the Wiener filter. The Wiener filter basically adds the optimal choice for C and makes it a function of spatial frequency.

7.2.4 * The Wiener filter

Wiener filtering is a theoretically optimal deblurring technique. It directly builds on the inverse filtering ideas developed in the previous section, but requires additional knowledge about the image statistics in exchange for this optimality.

First, we have to introduce the concept of *autocorrelation.* When images are processed, it is at times useful to correlate them. In this way we can tell, for example, how similar two brightness patterns are. The *cross correlation* of $a(x, y)$ and $b(x, y)$ is defined by

$$\phi_{ab}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(\xi - x, \eta - y) b(\xi, \eta) d\xi d\eta.$$

The difference with convolution lies in the direct use of $a(x, y)$, without prior reflection. If $a(x, y) = b(x, y)$, the result is called the *autocorrelation.* The autocorrelation of a function is symmetric, that is, $\phi_{aa}(-x, -y) = \phi_{aa}(x, y)$ and thus has an extremum at $(x, y) = (0, 0)$, that more particularly is always a global maximum. If $b(x, y) = a(x - x_0, y - y_0)$, then $\phi_{ab}(x, y)$ attains its maximum at (x_0, y_0). There can of course be other local extrema, but also other maxima with the same value if at least one of the functions is periodic. Even if $b(x, y)$ would not be an exact, but a slightly different copy of $a(x, y)$, then still could



Figure 7.17: *Result of inverse filtering using a modulation transfer function of the type* $B/(B^2 + ct.)$.

we most probably estimate the shift by looking for the maximum of ϕ_{ab}. The Fourier transforms of the cross correlations and autocorrelations are often informative. They are called *power spectra* for reasons that will become apparent immediately below, and they are denoted $\Phi_{ab}(u, v)$ and $\Phi_{aa}(u, v)$, respectively. If the Fourier transform of $a(x, y)$ is $A(u, v)$, then

$$\Phi_{aa}(u, v) = |A(u, v)|^2 = A^*(u, v)A(u, v),$$

where $A^*(u, v)$ is the complex conjugate of $A(u, v)$. Thus Φ_{aa} is always real, a property that can also be deduced from the symmetry of ϕ_{aa} or the fact that the transform of $a(-x, -y)$ is $A^*(u, v)$ as was used here. The power spectrum is shift-invariant, i.e. translating the spatial pattern doesn't alter the power spectrum, it would only alter the phase.

Now that we have looked at the cross and autocorrelation and their Fourier transforms, we can carry on with our discussion on optimal deblurring. Suppose that we are given the sum of the signal $b(x, y)$ and the noise $n(x, y)$. Our task is to recover, as best as we possibly can, the signal $b(x, y)$. The measure of how well we succeed will be the integral of the square of the difference between the output $o(x, y)$ and the desired signal $d(x, y)$. Usually $d(x, y)$ is just $b(x, y)$. We choose to minimize the integral of the square of the error because it leads to tractable mathematics. We then have to minimize

$$E = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (o(x, y) - d(x, y))^2 dxdy.$$

If we are to use a linear system for the filtering operation, we can characterize the system by means of its point-spread function $h(x, y)$. The input to the system is

$$i(x, y) = b(x, y) + n(x, y),$$

and the output is,

$$o(x,y) = i(x,y) * h(x,y).$$

So

$$E = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(o^2(x,y) - 2o(x,y)d(x,y) + d^2(x,y))dxdy.$$

Since $o^2 = (i*h)^2$,

$$\begin{aligned} o^2(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(x-\xi,y-\eta)h(\xi,\eta)d\xi d\eta \times \\ &\quad \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(x-\alpha,y-\beta)h(\alpha,\beta)d\alpha d\beta, \end{aligned}$$

and so

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} o^2(x,y)dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_{ii}(\xi-\alpha,\eta-\beta)h(\xi,\eta)h(\alpha,\beta)d\xi d\eta d\alpha d\beta,$$

where $\phi_{ii}(x,y)$ is the autocorrelation of $i(x,y)$. Moreover,

$$o(x,y)d(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} i(x-\xi,y-\eta)h(\xi,\eta)d(x,y)d\xi d\eta,$$

and so

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} o(x,y)d(x,y)dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_{id}(\xi,\eta)h(\xi,\eta)d\xi d\eta.$$

Finally we need

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d^2(x,y)dxdy = \phi_{dd}(0,0).$$

We can now rewrite the expression for the error term to be minimized in the form

$$\begin{aligned} E &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_{ii}(\xi-\alpha,\eta-\beta)h(\xi,\eta)h(\alpha,\beta)d\xi d\eta d\alpha d\beta \\ &\quad -2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_{id}(\xi,\eta)h(\xi,\eta)d\xi d\eta + \phi_{dd}(0,0). \end{aligned}$$

This expression is to be minimized by finding the appropriate point-spread function $h(x,y)$. This is again a problem in the *calculus of variations*. We have a functional in h only, none of its derivatives are used. Using the symmetry of the autocorrelation function ϕ_{ii}, we find the Euler-Lagrange equation

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_{ii}(\xi-\alpha,\eta-\beta)h(\alpha,\beta)d\alpha d\beta - \phi_{id}(\xi,\eta) = 0,$$

that is, perhaps surprisingly,

$$\phi_{id} = \phi_{ii} * h.$$

This equation for $h(x,y)$ can be solved by taking the Fourier transform,

$$\Phi_{id} = H\Phi_{ii},$$

where Φ_{ii} and Φ_{id} are the power spectra. The power spectra are thus all we need to know to design the image-restoring system under the given assumptions. Because power spectra do

not fully specify the original image and many images may have the same power spectrum, the same system will be optimal for a large class of images.

As an example, consider a system designed to suppress noise, that is, a system that takes the sum of the image $b(x, y)$ and the noise $n(x, y)$ and produces an output $o(x, y)$ that is as close as possible, in the least-squares sense, to the original image $b(x, y)$. Here $d(x, y) = b(x, y)$ and

$$i(x, y) = b(x, y) + n(x, y).$$

So,

$$\Phi_{id} = \Phi_{bb} + \Phi_{nb},$$

and

$$\Phi_{ii} = \Phi_{bb} + \Phi_{bn} + \Phi_{nb} + \Phi_{nn},$$

as can be seen by noting the definitions of Φ_{ii} and Φ_{id}. We now assume that the noise is not correlated to the signal, so that $\Phi_{bn} = \Phi_{nb} = 0$. Then

$$H = \frac{\Phi_{id}}{\Phi_{ii}} = \frac{\Phi_{bb}}{\Phi_{bb} + \Phi_{nn}} = \frac{1}{1 + \Phi_{nn}/\Phi_{bb}}.$$

It is clear what the optimal system is doing. In parts of the spectrum where the signal-to-noise ratio Φ_{bb}/Φ_{nn} is high, the gain is almost unity; in parts where the noise dominates, the gain is very low, approximately Φ_{bb}/Φ_{nn}, which is just the signal-to-noise ratio.
Now consider the case where the signal $b(x, y)$ is passed through a system with point-spread function $h(x, y)$ before the noise $n(x, y)$ is added. The result,

$$i = b * h + n,$$

is to be passed through a system with point-spread function $h'(x, y)$. The output

$$o = i * h'$$

should be as close as possible to the original image $b(x, y)$, in the least-squares sense. Here $d(x, y) = b(x, y)$, so that

$$\Phi_{id} = H\Phi_{bb} + \Phi_{nb}$$

and

$$\Phi_{ii} = H^2\Phi_{bb} + H(\Phi_{nb} + \Phi_{bn}) + \Phi_{nn}.$$

Assuming that the noise is not correlated with the signal, we have

$$H' = \frac{\Phi_{id}}{\Phi_{ii}} = \frac{H\Phi_{bb}}{H^2\Phi_{bb} + \Phi_{nn}}.$$

If the signal-to-noise ratio is high in a particular part of the spectrum, then $H' \sim 1/H$ there, while the gain is limited to about $H(\Phi_{bb}/\Phi_{nn})$ in parts where $\Phi_{nn} > |H|^2\Phi_{bb}$.

1	1	1
1	1	1
1	1	1

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Figure 7.18: *Two low-pass convolution masks.*

7.3 I - Noise suppression

There are many kinds of noise (white, Gaussian, salt and pepper, etc.) and as a consequence there is a wide variety of algorithms dedicated to the suppression of each type. Since a detailed discussion on this subject is outside the scope of this introductory text, three representative techniques have been selected: *convolution with low-pass filters*, *median filtering*, and *anisotropic diffusion*.

7.3.1 Low-pass convolution filters

As noted in section 7.1, noise often introduces high frequencies, whereas the image signal itself mainly consists of lower frequencies. Therefore, at the higher frequency part of the spectrum the signal to noise ratio is typically low and noise can be reduced by suppressing the higher frequencies.

Considering linear and shift-invariant filters, such low-pass filtering or "smoothing" can be obtained by calculating local averages of the grey levels. Examples of 3×3 and 5×5 masks are shown in fig. 7.18. The outputs have to be divided by 9 and 25, resp. Taking fig. 7.19 as an input, the results for these filters are shown in fig. 7.20.

Although high frequencies have been suppressed more than low frequencies, hence the blurred effect, these filters are not really low-pass: Both low and high frequencies survive. This can be understood by looking at the modulation transfer function for these filters (i.e. their Fourier transform). Since both filters are real and even, so is their Fourier transform. It therefore suffices to specify the real component of the MTF. Moreover, the MTFs can easily be found because these masks can be generated as convolutions of simpler masks, operating on the rows and columns. For instance the 3×3 mask is found as the convolution of a 3×1 mask $(1, 1, 1)^T$ working on the columns and a 1×3 mask $(1, 1, 1)$ working on the rows. Hence, the Fourier transform for the resulting 3×3 mask simply is the product of the Fourier transforms of these two masks. Thus, the MTF for the 3×3 filter is

$$\frac{1}{9}(1 + 2\cos(2\pi u))(1 + 2\cos(2\pi v)) \ ,$$

and for the 5×5 filter

$$\frac{1}{25}(1 + 2\cos(2\pi u) + 2\cos(4\pi u))(1 + 2\cos(2\pi v) + 2\cos(4\pi v)) \ .$$

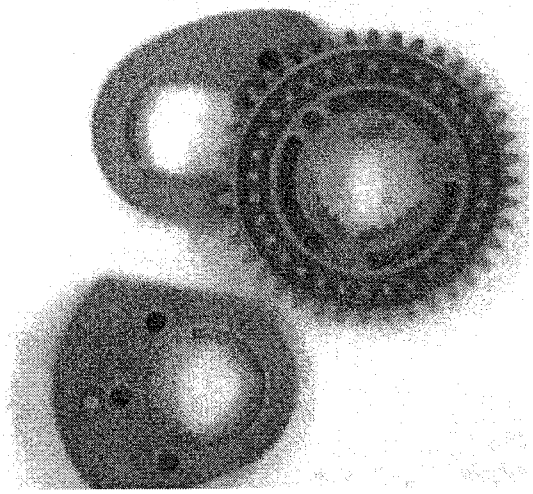

Figure 7.19: *Cogwheel scene.*

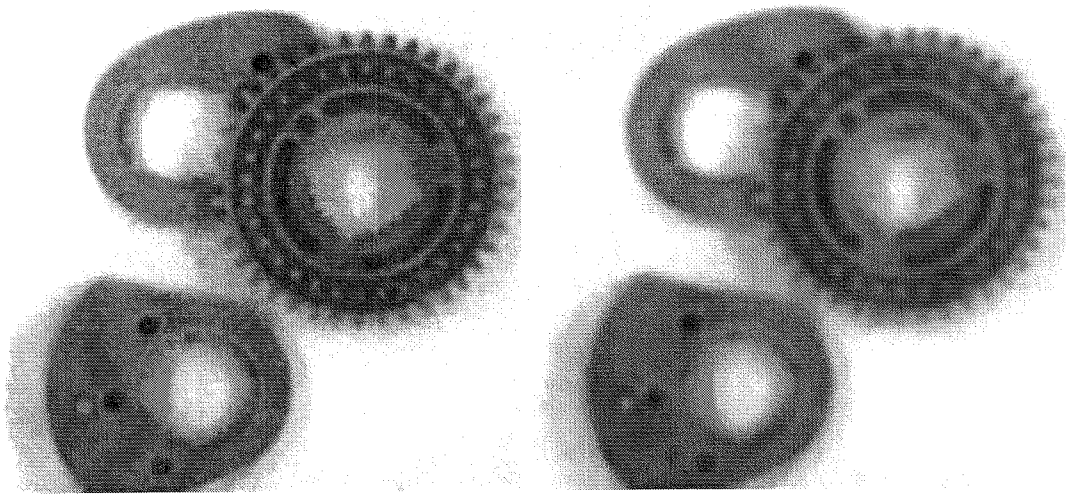

Figure 7.20: *Results obtained by processing fig. 7.19 with the* 3×3 *(left) and* 5×5 *convolution masks of fig. 7.18.*

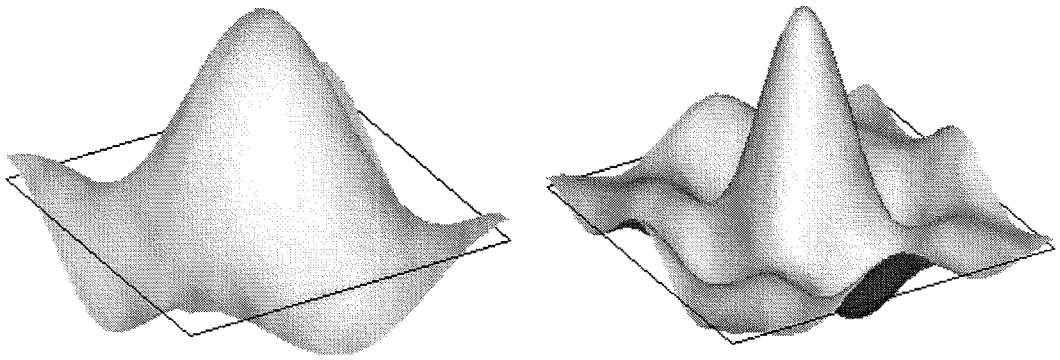

Figure 7.21: *Real components of the modulation transfer functions for the* 3×3 *(left) and* 5×5 *convolution masks of fig. 7.18.*

These real Fourier components are shown in fig. 7.21. The fact that the filters can be implemented as subsequent operations on the rows and the columns is also important from a practical point of view. The number of basic operations needed to implement such *separable filters* is lower than what is needed when implementing the two-dimensional filters directly. The MTFs being real function, there are no phase shifts to be expected, except for complete phase reversals (negative values). As can be observed, the first cut-off frequency is lower for the larger filter. However, the filters cannot be made to better approximate a purely low-pass filter by making them wider. The 5×5 filter only adds additional lobes. Making the filter wider would in the limit yield a sinc-shaped MTF.

Speaking of the sinc function, perfectly low-pass filters have already been introduced. An example of such operation was given in fig. 7.2, where an isotropic rectangular window type of filter (i.e. with sharp cutoff) was used. As shown in chapter 5, a rectangular cut-off filter has a sinc function as its inverse Fourier transform. The operation is equivalent to convolution with such a filter. The infinite extent of the filter in the spatial domain implies that this operation should preferably be carried out in the spatial frequency domain. Indeed, filters in the spatial domain that are to approximate this ideal cut-off operation will necessarily be large. However, its sign reversals cause a "rippling effect", as can be seen in fig. 7.2. In order to avoid this effect from occurring, it might be better to use a filter without sign reversals in the spatial domain.

Recapitulating, if a filter is to implement a genuine low-pass operation without causing ripples in the image, it must not have zero-crossings in either the spatial or the spatial frequency domain. The class of *binomial filters* has this property. One-dimensional binomial filters are generated by convolving $(1, 1)$ a number of times with itself. For image processing applications filters with odd dimensions are to be preferred, since the new value can then be assigned to the central position, which coincides with a pixel. Hence, this

1	2	1
2	4	2
1	2	1

Figure 7.22: 3×3 *binomial filter mask.*

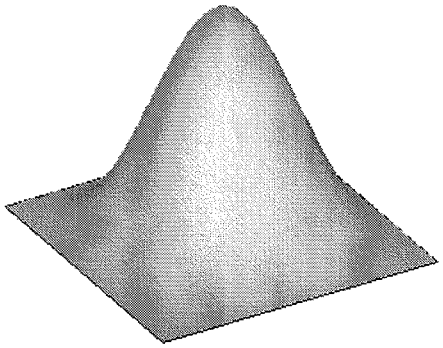

Figure 7.23: *MTF of the* 3×3 *binomial filter mask.*

restriction amounts to using $(1, 2, 1)$ and this filter convolved with itself. Thus, the next smallest binomial filter is $(1, 4, 6, 4, 1)$. Two-dimensional extensions are generated as separable filters with the same mask operating on the rows and columns. The two-dimensional filter generated with $(1, 2, 1)$ is shown in fig. 7.22. The output of this filter has to be divided by 16. The MTF of this filter is

$$(2 + 2\cos(2\pi u))(2 + 2\cos(2\pi v)) \ .$$

This is the MTF of a genuine low-pass filter, as shown in fig. 7.23. It's effect on fig. 7.19 is shown in fig. 7.24. Selecting larger binomial filters will render the resulting filters more isotropic. In the limit, the filters will approach a Gaussian. The Fourier transform of a Gaussian is a Gaussian, and consequently no zero-crossings occur in either the spatial and the spatial frequency domain. The above 3×3 filter can be considered an approximation to the Gaussian with minimal spatial extent. Fig. 7.25 shows an example of Gaussian smoothing. As can be seen, the rippling effect problem has successfully been removed. Yet, the important problem of unsharp edges remains and the result is still quite unpleasant to the eye.

High frequency noise suppression and edge degradation cannot be disentangled as long as linear filters are applied. The next section introduces a non-linear technique.

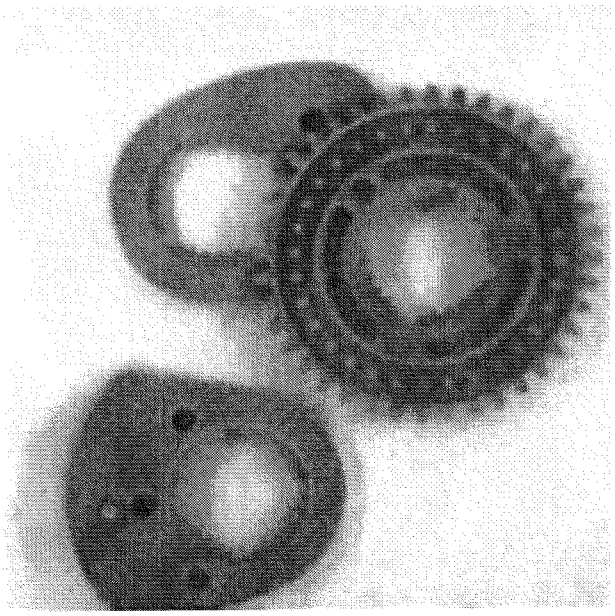

Figure 7.24: *Result for the* 3×3 *binomial filter mask.*



Figure 7.25: *TV presenter image with noise added (left) and result of smoothing with a Gaussian filter (right).*

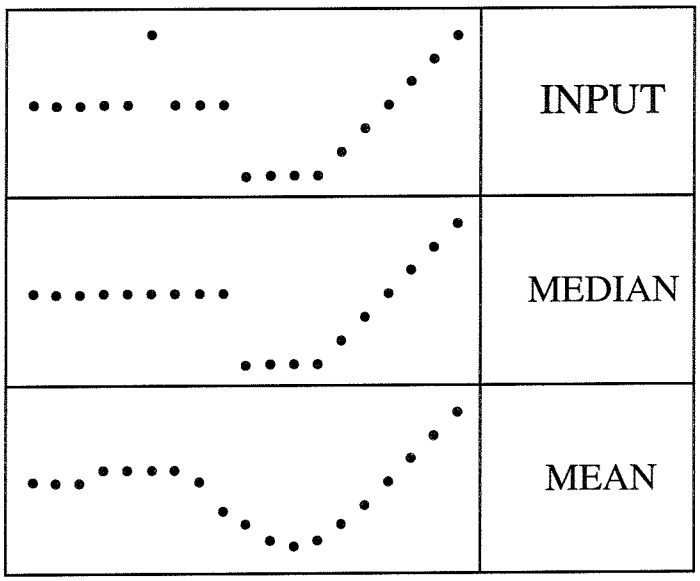

Figure 7.26: *Comparison of median and linear filtering for the top row sequence.*

7.3.2 Median filters

The low-pass method of the previous section replaces each grey level value by a linear combination of grey levels in a neighbourhood of the pixel under scrutiny. *Median filtering*, on the other hand, is based on the rank-order of these neighbourhood values. The grey levels are rearranged in order of increasing values. The pixel's grey level is then replaced by the value that is such that there are as many higher as lower values in the rank order. Thus, the new grey level is always already present in the image. No new grey levels emerge. The main advantage of this type of filter is its "odd-man-out" effect, i.e. abnormal values are completely discarded. Suppose e.g. we use a one-dimensional median filter on the string

$$1, 1, 1, 7, 1, 1, 1, 1.$$

Let us use a median filter of width 3, i.e. we use 3 subsequent values at a time. A string $1, 1, 1$ is replaced by the new value 1, because ranking this string would simply return it unchanged, with 1 as the value at the middle. Arriving at a string like $1, 7, 1$, however, the rank-ordered string will be $1, 1, 7$ and 1 will still be the middle value. We see that the filtered values for the above string will be

$$?, 1, 1, 1, 1, 1, 1, ?,$$

question marks indicating that these values cannot be calculated without information about the neighbouring values that are not shown. As can be seen the 7 has been removed from the sequence. Using a linear filter, such exceptional values will have a strong influence, however. These observations are further illustrated in fig. 7.26. For the linear filter the mean was selected. Both filters have been given width 5. As can be seen, the spike is



Figure 7.27: *TV presenter image output for* 3×3 *median filter.*

effectively filtered out by both methods. Nevertheless, the mean filter is influenced by its presence. Furthermore, this filter disrupts the structure of the pattern by rounding off all discontinuities. The output of the median filter is seen to only lift out the spike, leaving all discontinuities intact. One should not become over-optimistic, however. Fig. 7.27 shows the output of a (3×3) median filter for the noisy image in fig. 7.25. Some edges in this image are sharper than in the linearly filtered image of fig. 7.25 (e.g. the upper lip, the shoulders). On the other hand, the eyes have been deformed more seriously by the median filter. Median filters destroy cusps and protrusions in object outlines more radically than linear filters do. Also notice the disappearance of the presenter's ear-ring. Moreover, if the reduction of noise takes several iterations, the median filter tends to generate a patchy outlook, i.e. regions of constant grey level are separated by step discontinuities. This effect is quite salient, as illustrated in fig. 7.28, where the noisy original and the result of 10 iterations with a 3×3 median filter are shown side by side.

7.4 II - Combined deblurring and noise suppression

Unsharp masking – an approximation of a diffusion process running backwards – was shown to successfully sharpen images. Since it is a linear method, however, it will also boost the noise. Hence, a method is called for that is more selective in that it only sharpens edges and detail that is sufficiently outspoken. The idea is then to devise a process that behaves as usual diffusion on parts of the image without such edges and thus smooths out the noise, whereas it turns into a diffusion run backwards at places with outspoken edges.

Such scheme is offered by the *anisotropic diffusion* method: The envisaged effect is



Figure 7.28: *Right: result of 10 iterations with a* 3×3 *median filter for the image on the left.*

achieved by making the diffusion coefficient in

$$\frac{\partial f}{\partial t} = c\, \nabla^2 f$$

anisotropic. This results in the equation

$$\frac{\partial f}{\partial t} = div(c(f)\nabla f) \tag{7.2}$$

This operation is similar to diffusion (cfr. section 7.2.2), but now the grey level of a pixel is modified by a weighted sum of the grey level differences with the surrounding pixels, as shown in fig. 7.29. Choosing $c = 0$ at the boundaries and $c = 1$ elsewhere, encourages smoothing within the region in preference to smoothing across the boundaries. Of course, the boundaries are not known in advance. As will be clarified in section **??**, the magnitude of the grey level gradient gives a good indication for the presence of edges. Hence, a solution is to choose c locally as a function of the gradient: $c(f) \equiv c(||\nabla f||)$. Possible functions are

$$c(f) = e^{-(||\nabla f||/K)^2} \quad \text{and} \quad c(f) = \frac{1}{1 + (||\nabla f||/K)^2}$$

with K some constant.

This algorithm does succeed in sharpening edges. This can rather easily be demonstrated on a one-dimensional grey level profile. To avoid unnecessary complications the

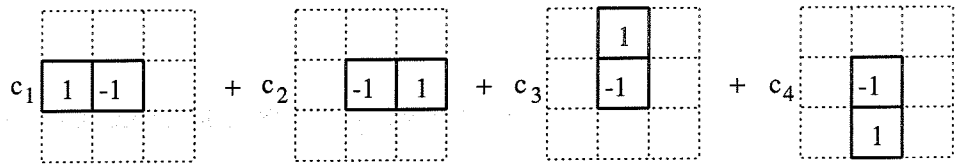

Figure 7.29: *With anisotropic diffusion, the grey level of the central pixel is modified with an amount which is calculated as a weighted sum of the local differences in grey level. The weights c_i are given a value that depends on the difference they are to be multiplied with (see text).*

conductivity function

$$c(p) = e^{-p^2} \qquad \text{with} \qquad p(x,t) = \frac{\partial f}{\partial x}(x,t),$$

is used. This choice is by no means essential and any other conductivity function of a similar qualitative form would yield the same result. In one dimension, eq.(7.2) becomes

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial x}\left(c(p)\frac{\partial f}{\partial x}\right) = c(p)\frac{\partial^2 f}{\partial x^2} + \left(\frac{\partial c}{\partial p}\right)\left(\frac{\partial p}{\partial x}\right)\frac{\partial f}{\partial x} \ .$$

Using the proposed form for $c(p)$ and the fact that

$$\frac{\partial f}{\partial x} = p \qquad \text{and} \qquad \frac{\partial p}{\partial x} = \frac{\partial^2 f}{\partial x^2} \ ,$$

it follows that we can cast the above PDE in the form of a diffusion equation

$$\frac{\partial f}{\partial t} = a(p)\frac{\partial^2 f}{\partial x^2} \qquad \text{where} \qquad a(p) \equiv c(p) + p\frac{dc}{dp} = e^{-p^2}(1-2p^2) \ . \tag{7.3}$$

Notice how the sign of the diffusion coefficient $a(p)$ in eq. (7.3) depends on p, the local grey level contrast (see fig. 7.30). If $|p|$ is small, the diffusion coefficient is positive, resulting in the usual smoothing process; for values of $|p|$ beyond a certain critical threshold however, $a(p)$ becomes negative causing the diffusion to run in reversed time.

However, discretization effects and "gaps" in the edges surrounding regions cause the contrast over the edges to diminish gradually, until they are no longer visible. As a consequence, the process should be halted at a stage where an acceptable compromise between edge sharpness and contrast is established. Stopping after too many iterations, results in a considerable loss of image quality. This is illustrated by the left image in fig. 7.31, which was generated using 500 iterations. This evolution towards a homogeneous solution can be halted by adding a restraining force. This force keeps the solution closer to the initial image f_o:

$$\frac{\partial f}{\partial t} = div(c(f)\nabla f) - \frac{1}{\sigma^2}(f - f_o) \ . \tag{7.4}$$

The result is shown as the right image in fig. 7.31. Notice how this non-linear scheme succeeds in suppressing the noise ánd keeping the edges and meaningful details such as the ear-rings intact (and even sharpens them!). It should be noted that typically 50 to 100 rather than 500 iterations have to be applied.

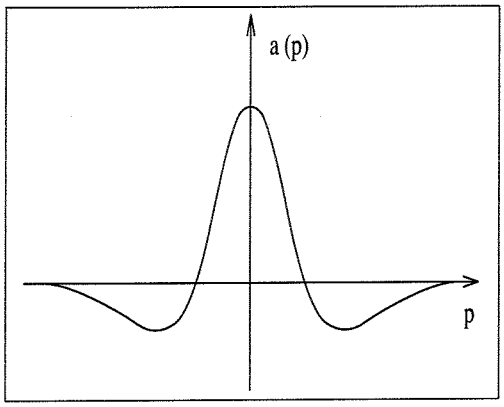

Figure 7.30: *The qualitative behaviour of the flux-coefficient $a(p)$.*

7.5 II - Reverting motion blur

If objects move fast or if the camera is e.g. panned fast, the images will show motion blur. Special techniques for removing such blur have been developed. Here we discuss one.

Suppose blur finds its origin in a translational motion of constant speed of a grey level pattern $i(x, y)$. For reasons of computational simplicity, the motion is assumed to be purely horizontal. Putting $t = 0$ (with t time) at the beginning of the image acquisition step and T the acquisition duration, and assuming that the (horizontal) velocity is given by u, the resulting (blurred) image intensity $b(x, y)$ is given by

$$b(x, y) = \frac{1}{T} \int_0^T i(x - ut, y)\ dt \ .$$

Due to the purely one-dimensional character of the motion, one can in fact consider one image row at a time and write:

$$b(x) = \frac{1}{T} \int_0^T i(x - ut)\ dt \ .$$

or using the substitution $\tau = x - ut$,

$$b(x) = \frac{1}{uT} \int_{x-uT}^{x} i(\tau)\ d\tau \ .$$

Thus,

$$b'(x) = \frac{db(x)}{dx} = \frac{1}{uT} \left(i(x) - i(x - uT)\right) \ ,$$

or,

$$i(x) = uTb'(x) + i(x - uT) \ . \tag{7.5}$$

Hence, the unblurred pattern $i(x)$ could be reconstructed recursively, one vertical strip of width uT at a time. Of course, the problem lies with the first strip. Suppose there are S such complete strips, the first starting at the first column for its left border. At the right



Figure 7.31: *Left: anisotropic diffusion after 500 iterations for the original image in fig. 7.28. Right: result for the same process with the restraining force added.*

border of the image, we might have part of an additional strip that is discarded in order not to complicate the discussion. Writing

$$\begin{aligned} i_s(x^*) &= i(x^* + (s-1)uT) \ , \qquad 0 \leq x^* < uT \\ b'_s(x^*) &= b'(x^* + (s-1)uT) \ , \end{aligned}$$

for the points in the sth strip, one finds for strip $\sigma > 1$ that

$$i_\sigma(x^*) = i_1(x^*) + uT \sum_{s=2}^{\sigma} b'_s(x^*) \ .$$

Taking the sum over the strips

$$\sum_{\sigma=2}^{S} i_\sigma(x^*) = (S-1)i_1(x^*) + uT \sum_{\sigma=2}^{S} \sum_{s=2}^{\sigma} b'_s(x^*) \ .$$

If S is sufficiently large, the left hand side can be approximated by $(S-1)\bar{i}$, with $\bar{i}$ the mean grey level. Since blurring typically doesn't change the average grey level, this value can be approximated by $\bar{b}$, the average of the intensity image $b(x)$. Thus,

$$i_1(x^*) = \bar{b} - \frac{uT}{S-1} \sum_{\sigma=2}^{S} \sum_{s=2}^{\sigma} b'_s(x^*) \ .$$

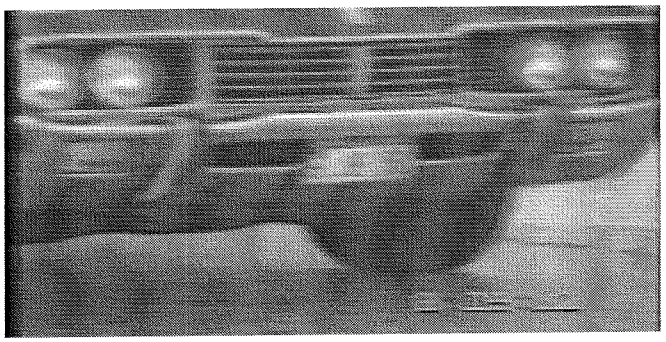

Figure 7.32: *Car image with motion blur.*

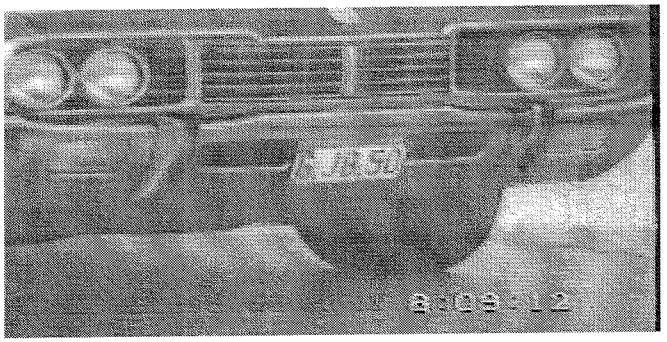

Figure 7.33: *Reconstructed image with motion blur removed.*

Since the problem with the recursion formula 7.5 was the need to first reconstruct the first strip, all necessary information for a complete image reconstruction is available now.

An example of this technique is shown for the blurred image in fig. 7.32. The scene has undergone a pure horizontal motion of 16 pixels during image formation. Fig. 7.33 shows the deblurred image. As can be seen, spurious horizontal stripes have been created. These emerge at places where the true intensity sums insufficiently agree with the average-based approximations.

7.6 II - Undoing radial distortion

not available yet

7.7 Further reading

Not available yet