9.4.1 Linear Filters

Several filtering algorithms will be presented together with the most useful supports.

• *Uniform filter* – The output image is based on a local averaging of the input filter where all of the values within the filter support have the same weight. In the continuous spatial domain (x,y) the *PSF* and transfer function are given in Table 4–T.1 for the rectangular case and in Table 4–T.3 for the circular (pill box) case. For the discrete spatial domain [m,n] the filter values are the samples of the continuous domain case. Examples for the rectangular case (J=K=5) and the circular case (R=2.5) are shown in Figure 26.

$$h_{rect}[j,k] = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad h_{circ}[j,k] = \frac{1}{21}\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

(a) Rectangular filter (J=K=5) **(b)** Circular filter (R=2.5)

Figure 26: Uniform filters for image smoothing

Note that in both cases the filter is normalized so that $\Sigma\ h[j,k] = 1$. This is done so that if the input $a[m,n]$ is a constant then the output image $c[m,n]$ is the same constant. The justification can be found in the Fourier transform property described in eq. (26). As can be seen from Table 4, both of these filters have transfer functions that have negative lobes and can, therefore, lead to phase reversal as seen in Figure 23. The square implementation of the filter is separable and incremental; the circular implementation is incremental [24, 25].

• *Triangular filter* – The output image is based on a local averaging of the input filter where the values within the filter support have differing weights. In general, the filter can be seen as the convolution of two (identical) uniform filters either rectangular or circular and this has direct consequences for the computational complexity [24, 25]. (See Table 13.) In the continuous spatial domain the *PSF* and transfer function are given in Table 4–T.2 for the rectangular support case and in Table 4–T.4 for the circular (pill box) support case. As seen in Table 4 the transfer functions of these filters do not have negative lobes and thus do not exhibit phase reversal.

Examples for the rectangular support case (J=K=5) and the circular support case (R=2.5) are shown in Figure 27. The filter is again normalized so that $\Sigma\ h[j,k]$=1.

$$h_{rect}[j,k] = \frac{1}{81}\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} \qquad h_{circ}[j,k] = \frac{1}{25}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

(a) Pyramidal filter (*J*=*K*=5) **(b)** Cone filter (*R*=2.5)

Figure 27: Triangular filters for image smoothing

• *Gaussian filter* – The use of the Gaussian kernel for smoothing has become extremely popular. This has to do with certain properties of the Gaussian (e.g. the central limit theorem, minimum space-bandwidth product) as well as several application areas such as edge finding and scale space analysis. The *PSF* and transfer function for the continuous space Gaussian are given in Table 4–T6. The Gaussian filter is separable:

$$\begin{aligned} h(x,y) &= g_{2D}(x,y) = \left(\frac{1}{\sqrt{2\pi}\sigma}e^{-\left(x^2/2\sigma^2\right)}\right)\bullet\left(\frac{1}{\sqrt{2\pi}\sigma}e^{-\left(y^2/2\sigma^2\right)}\right) \\ &= g_{1D}(x)\bullet g_{1D}(y) \end{aligned} \tag{93}$$

There are four distinct ways to implement the Gaussian:

– Convolution using a finite number of samples (N_o) of the Gaussian as the convolution kernel. It is common to choose $N_o = \lceil 3\sigma \rceil$ or $\lceil 5\sigma \rceil$.

$$g_{1D}[n] = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma}e^{-\left(n^2/2\sigma^2\right)} & |n| \le N_o \\ 0 & |n| > N_o \end{cases} \tag{94}$$

– Repetitive convolution using a uniform filter as the convolution kernel.

$$\begin{aligned} &g_{1D}[n] \approx u[n] \otimes u[n] \otimes u[n] \\ &u[n] = \begin{cases} 1/(2N_o+1) & |n| \le N_o \\ 0 & |n| > N_o \end{cases} \end{aligned} \tag{95}$$

The actual implementation (in each dimension) is usually of the form:

$$c[n] = ((a[n] \otimes u[n]) \otimes u[n]) \otimes u[n] \tag{96}$$