计算视觉与模式识别

线性滤波器

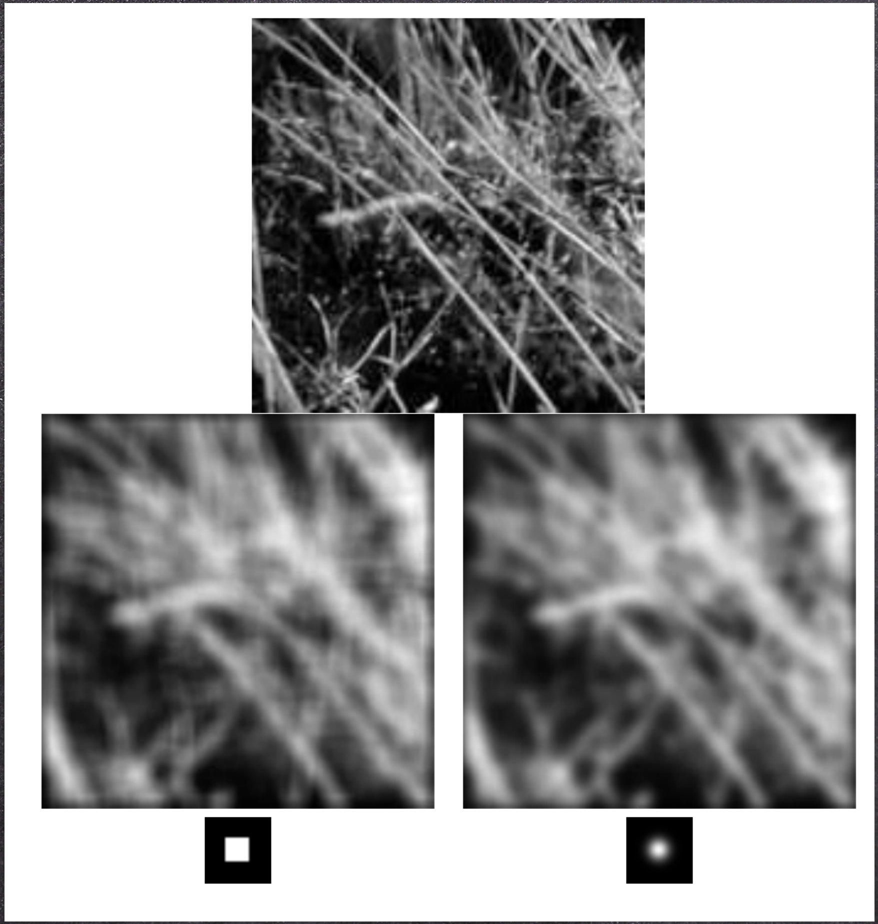
平均值：

$$\mathcal{R}_{ij} = \frac{1}{(2k+1)^2} \sum_{u=i-k}^{u=i+k} \sum_{v=j-k}^{v=j+k} \mathcal{F}_{uv}$$

卷积：

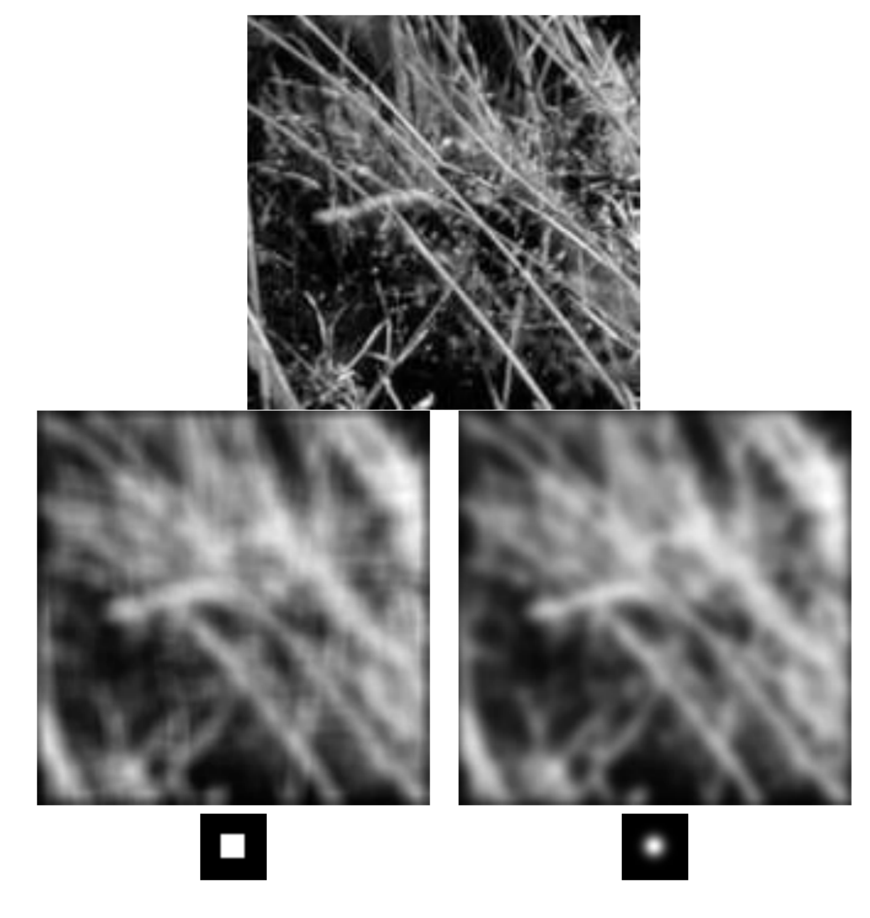
$$\mathcal{R}_{ij} = \sum_{u,v} H_{i-u,j-v} F_{u,v}$$

均值滤波示例



- Figure 8.1.

均值滤波示例



- Figure 8.1.

Although a uniform local average may seem to give a good blurring model, it generates effects that are not usually seen in defocussing a lens. The images above compare the effects of a uniform local average with weighted average. The image at the top shows a view of grass. On the left in the second row, the result of blurring this image using a uniform local model and on the right, the result of blurring this image using a set of Gaussian weights. The degree of blurring in each case is about the same, but the uniform average produces a set of narrow vertical and horizontal bars — an effect often known as ringing. The bottom row shows the weights used to blur the image, themselves rendered as an image; bright points represent large values and dark points represent small values (in this example the smallest values are zero).

对称高斯核：

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

离散形式，$2k+1\times 2k+1$矩阵：

$$H_{ij} = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{((i-k-1)^2+(j-k-1)^2)}{2\sigma^2}\right)$$

高斯核

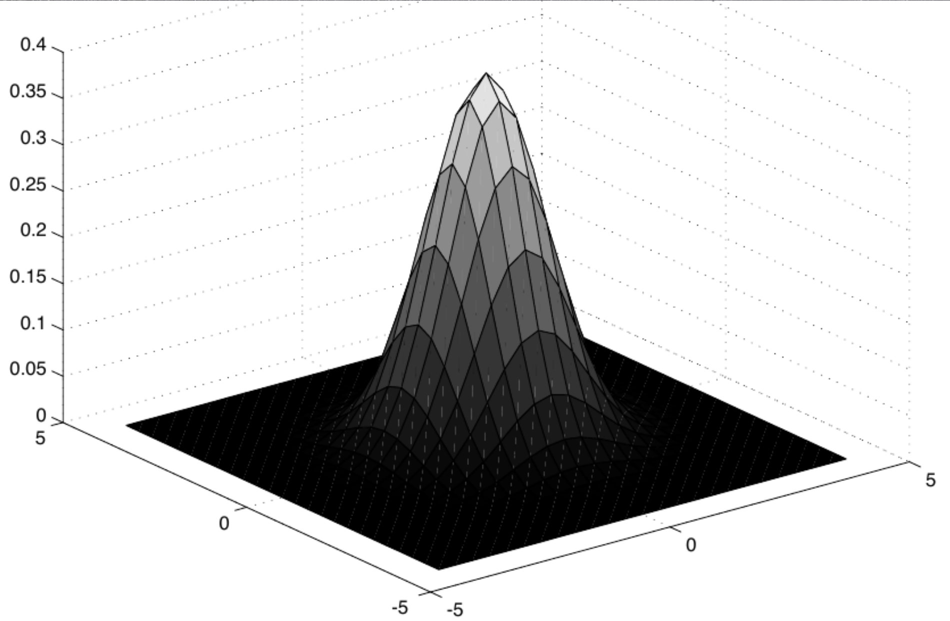
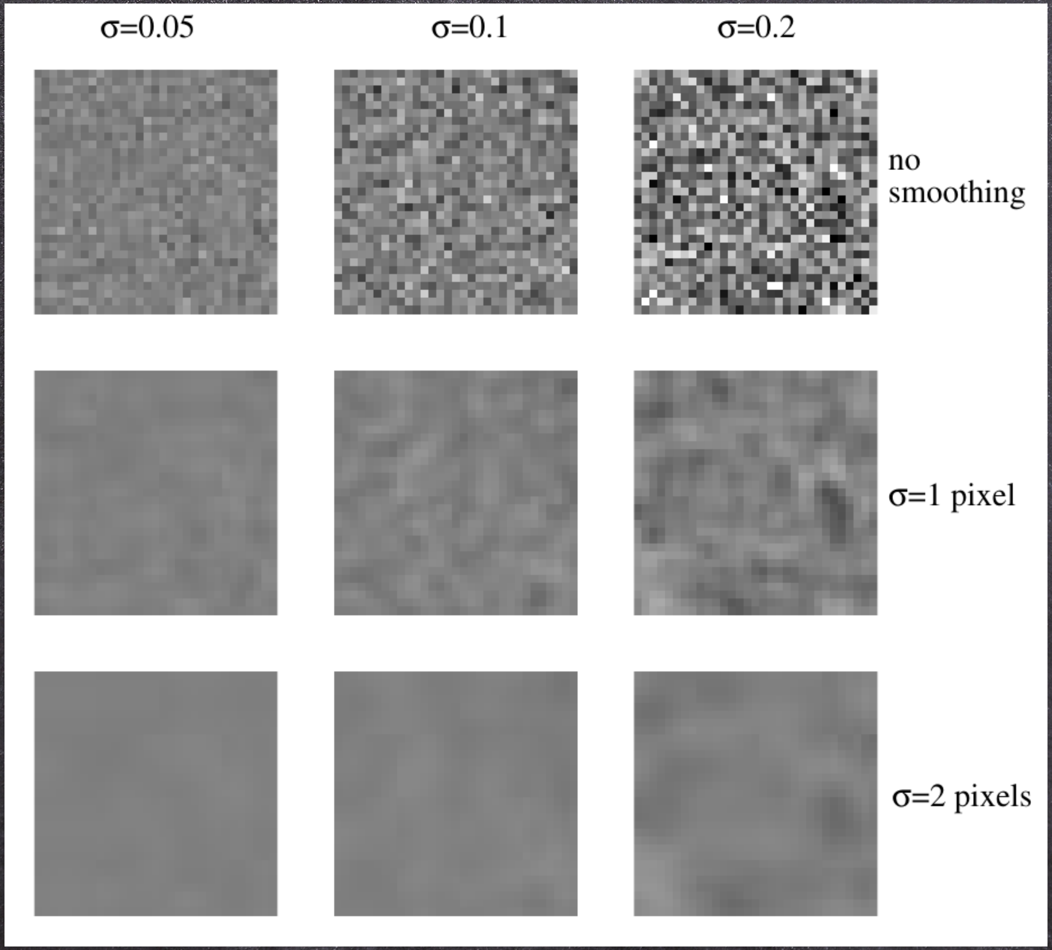


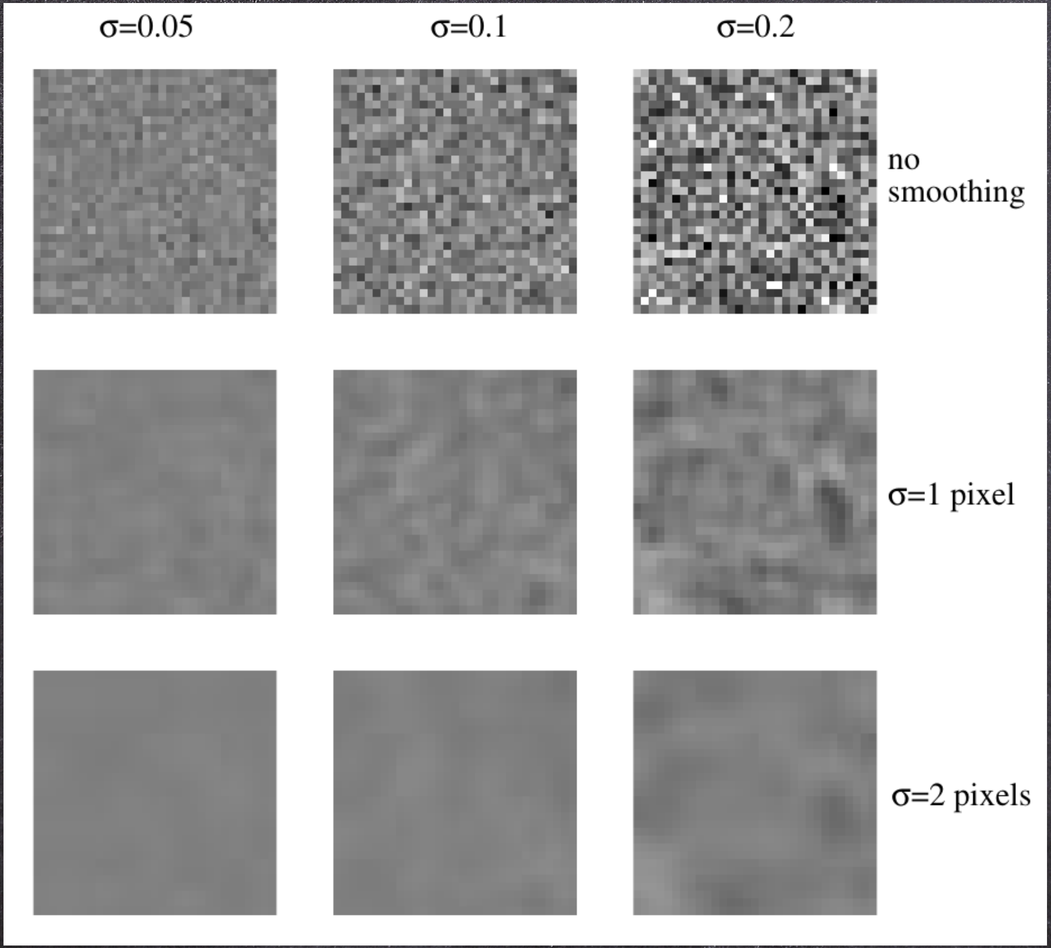
Figure 8.2. The symmetric Gaussian kernel in 2D. This view shows a kernel scaled so that its sum is equal to one; this scaling is quite often omitted. The kernel shown has $\sigma = 1$. Convolution with this kernel forms a weighted average which stresses the point at the center of the convolution window, and incorporates little contribution from those at the boundary. Notice how the Gaussian is qualitatively similar to our description of the point spread function of image blur; it is circularly symmetric, has strongest response in the center, and dies away near the boundaries.

高斯噪声与高斯滤波器



- FIGURE 4.3:

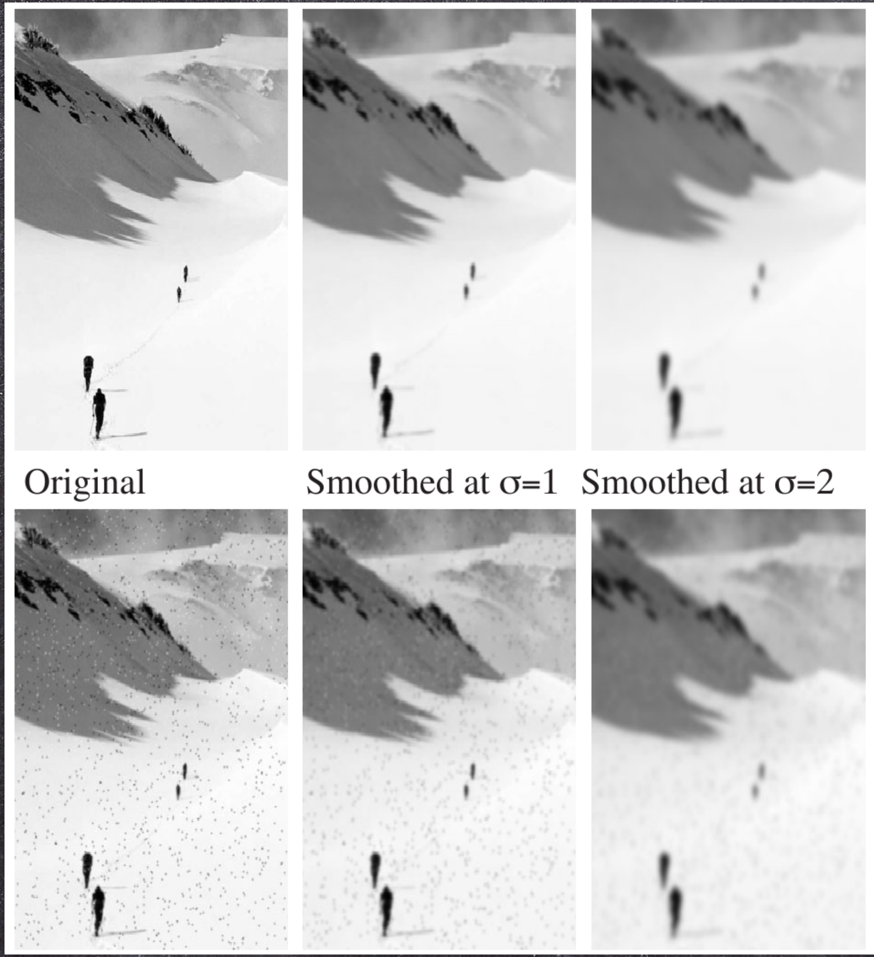
高斯噪声与高斯滤波器



- FIGURE 4.3:

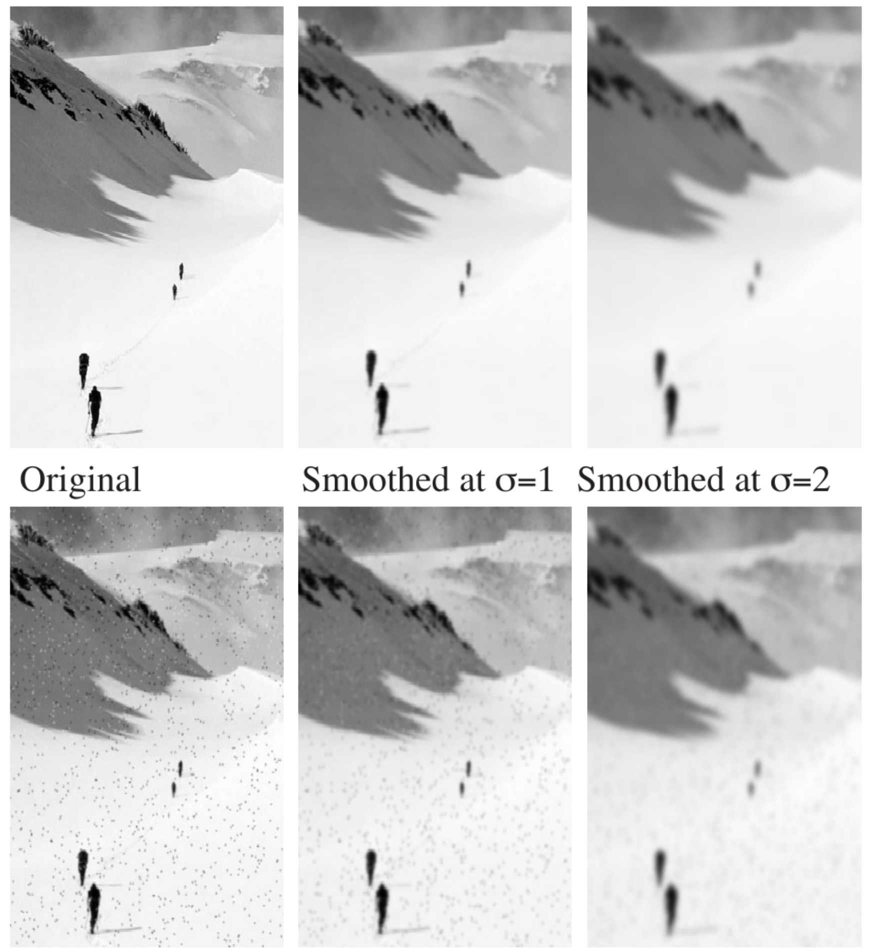
The top row shows images of a constant mid-gray level corrupted by additive Gaussian noise. In this noise model, each pixel has a zero-mean normal random variable added to it. The range of pixel values is from zero to one, so that the standard deviation of the noise in the first column is about $1/20$ of full range. The center row shows the effect of smoothing the corresponding image in the top row with a Gaussian filter of σ one pixel. Notice the annoying overloading of notation here; there is Gaussian noise and Gaussian filters, and both have σ's. One uses context to keep these two straight, although this is not always as helpful as it could be, because Gaussian filters are particularly good at suppressing Gaussian noise. This is because the noise values at each pixel are independent, meaning that the expected value of their average is going to be the noise mean. The bottom row shows the effect of smoothing the corresponding image in the top row with a Gaussian filter of σ two pixels.

椒盐噪声与高斯滤波



- Figure 8.3.

椒盐噪声与高斯滤波



- Figure 8.3.

In salt-and-pepper noise, we choose pixels uniformly at random, and uniformly at random make them either black or white. Gaussian smoothing is particularly effective at suppressing the effects of salt-and-pepper noise. The top row shows an image, and versions smoothed by a symmetric Gaussian with σ two pixels and four pixels. The images in the second row are obtained by corrupting the images in the top row by this noise model and then smoothing the result. Notice that, as the smoothing increases, detail is lost, but the effects of the noise diminish, too — the smoothed versions of the noisy images look very much like the smoothed version of the noise-free images.

导数与有限差分

偏导数

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x, y)}{\varepsilon}$$

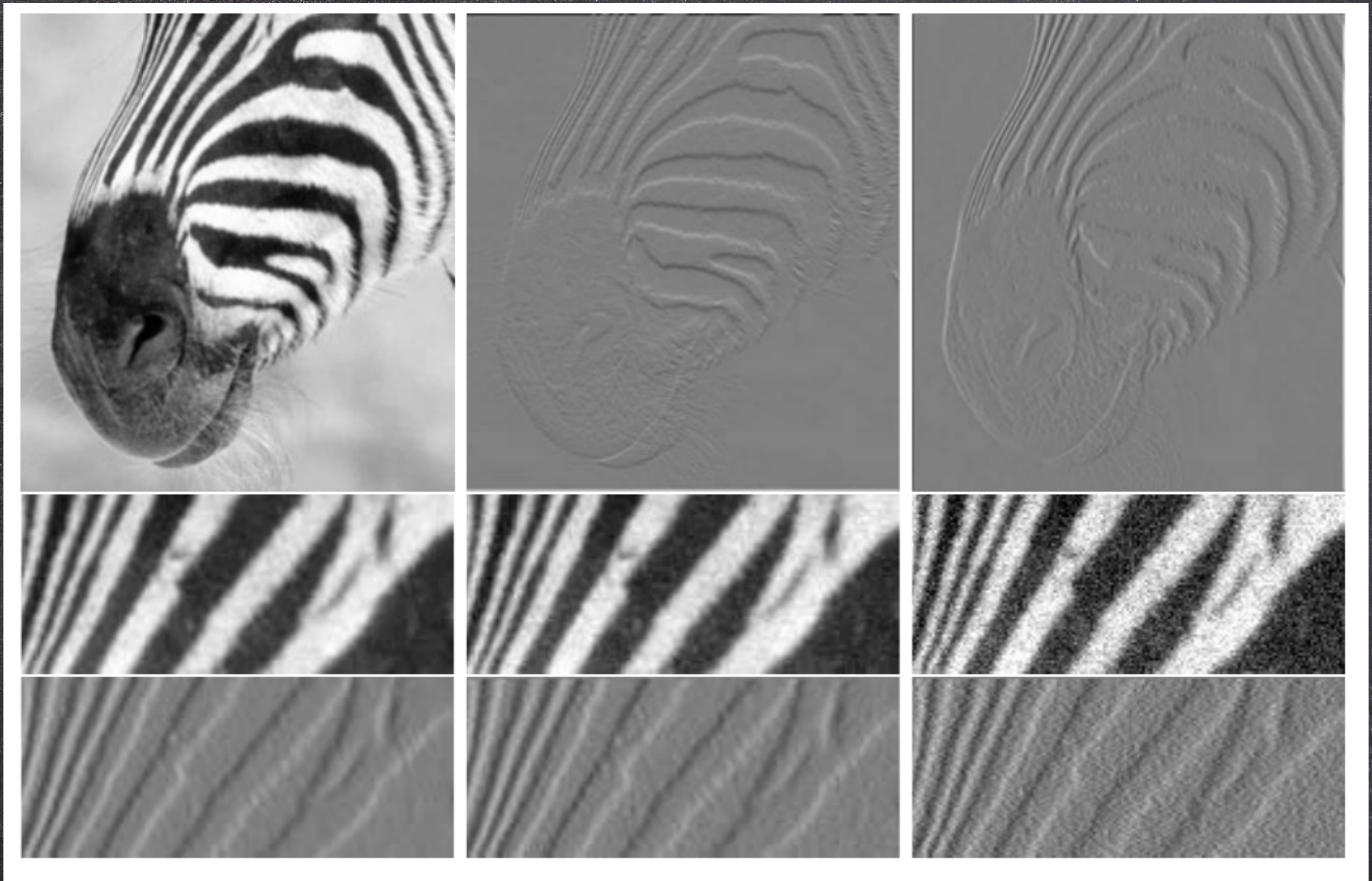
有限差分

$$\frac{\partial h}{\partial x} \approx h_{i+1,j} - h_{i-1,j}$$

卷积核

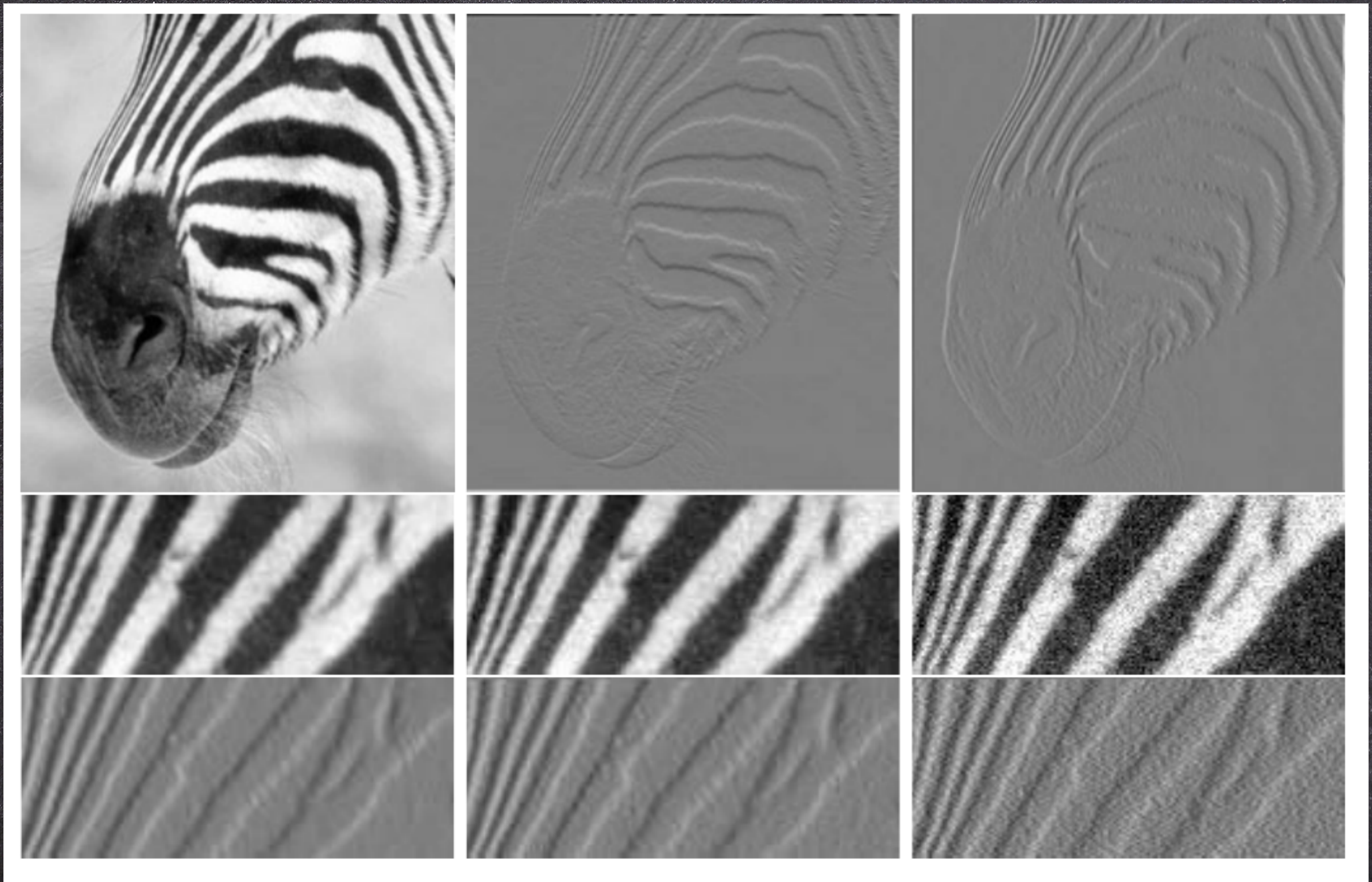
$$\mathcal{H} = \left\{ \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{matrix} \right\}$$

差分图像



- FIGURE 4.4:

差分图像



- FIGURE 4.4:

The top row shows estimates of derivatives obtained by finite differences. The image at the left shows a detail from a picture of a zebra. The center image shows the partial derivative in the y-direction—which responds strongly to horizontal stripes and weakly to vertical stripes—and the right image shows the partial derivative in the x-direction—which responds strongly to vertical stripes and weakly to horizontal stripes. However, finite differences respond strongly to noise. The image at center left shows a detail from a picture of a zebra; the next image in the row is obtained by adding a random number with zero mean and normal distribution ($\sigma = 0.03$; the darkest value in the image is 0, and the lightest 1) to each pixel; and the third image is obtained by adding a random number with zero mean and normal distribution ($\sigma = 0.09$) to each pixel. The bottom row shows the partial derivative in the x-direction of the image at the head of the row. Notice how strongly the differentiation process emphasizes image noise; the derivative figures look increasingly grainy. In the derivative figures, a mid-gray level is a zero value, a dark gray level is a negative value, and a light gray level is a positive value.

平移不变线性系统

线性

$$\begin{aligned} R(f+g) &= R(f)+R(g) \\ R(kf) &= kR(f) \end{aligned}$$

平移不变

$$\begin{aligned} R(f(x)) &= g(x) \\ R(f(x-y)) &= g(x-y) \end{aligned}$$

离散卷积

$$
\begin{aligned}
\boldsymbol{e}_0 &= \ldots.0,0,0,1,0,0,0.\ldots \\
\boldsymbol{f} &= \sum_i f_i \mathrm{Shift}(\boldsymbol{e}_0, i) \\
R(\mathrm{Shift}(\boldsymbol{f}, k)) &= \mathrm{Shift}(R(\boldsymbol{f}), k) \\
R(k\boldsymbol{f}) &= kR(\boldsymbol{f})
\end{aligned}
$$

$$
\begin{aligned}
R(\boldsymbol{f}) &= R\left(\sum_i f_i \mathrm{Shift}(\boldsymbol{e}_0, i)\right) \\
&= \sum_i R(f_i \mathrm{Shift}(\boldsymbol{e}_0, i)) \\
&= \sum_i f_i R(\mathrm{Shift}(\boldsymbol{e}_0, i)) \\
&= \sum_i f_i \mathrm{Shift}(R(\boldsymbol{e}_0), i)
\end{aligned}
$$

离散卷积（续）

$$
\begin{aligned}
\boldsymbol{g} &= R(\boldsymbol{e}_0) \\
R(\boldsymbol{f}) &= \sum_i f_i \mathrm{Shift}(\boldsymbol{g}, i) \\
&= \boldsymbol{g} * \boldsymbol{f} \\
R_j &= \sum_i g_{j-i} f_i
\end{aligned}
$$

二维卷积

$$
\begin{aligned}
\mathcal{E}_{00} &= \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} \\
R_{ij} &= \sum_{u,v} G_{i-u,j-v} F_{uv} \\
\mathcal{R} &= \mathcal{G} ** \mathcal{H}
\end{aligned}
$$

连续卷积

$$\begin{aligned}
R(kf) &= kR(f) \\
\mathrm{Shift}(f,c) &= f(u-c) \\
R(\mathrm{Shift}(f,c)) &= \mathrm{Shift}(R(f),c) \\
\mathrm{box}_\varepsilon(x) &= \begin{cases} 0 & \mathrm{abs}(x) > \frac{\varepsilon}{2} \\ 1 & \mathrm{abs}(x) < \frac{\varepsilon}{2} \end{cases} \\
R\left(\sum_i f_i \mathrm{Shift}(\mathrm{box}_\varepsilon, x_i)\right) &= \sum_i R(f_i \mathrm{Shift}(\mathrm{box}_\varepsilon, x_i)) \\
&= \sum_i f_i R(\mathrm{Shift}(\mathrm{box}_\varepsilon, x_i)) \\
&= \sum_i f_i \mathrm{Shift}\left(R\left(\frac{\mathrm{box}_\varepsilon}{\varepsilon}\varepsilon\right), x_i\right) \\
&= \sum_i f_i \mathrm{Shift}\left(R\left(\frac{\mathrm{box}_\varepsilon}{\varepsilon}\right), x_i\right)\varepsilon
\end{aligned}$$

连续卷积（续）

δ函数

$$
\begin{aligned}
d_\varepsilon(x) &= \frac{\mathrm{box}_\varepsilon(x)}{\varepsilon} \\
\delta(x) &= \lim_{\varepsilon\to 0} d_\varepsilon(x)
\end{aligned}
$$

卷积

$$
\begin{aligned}
R(f) &= \int \{R(\delta(u-x'))\} f(x')\mathrm{d}x' \\
&= \int g(u-x') f(x')\mathrm{d}x' \\
&= (g*f) \\
(g*f) &= (h*g) \\
(f*(g*h)) &= ((f*g)*h)
\end{aligned}
$$

二维卷积

$$\begin{aligned} d_\varepsilon(x,y) &= \frac{\mathrm{box}_{\varepsilon^2}(x,y)}{\varepsilon^2} \\ R(h) &= \iint g(x-x', y-y')h(x', y')\mathrm{d}x\mathrm{d}y \\ &= (g**h) \\ (g**h) &= (h**g) \end{aligned}$$

傅里叶变换

$$\begin{aligned}\mathcal{F}(g(x,y)) &= \iint g(x,y)e^{-i2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y \\ e^{-i2\pi(ux+vy)} &= \cos(2\pi(ux+vy))+i\sin(2\pi(ux+vy))\end{aligned}$$

线性

$$\begin{aligned}\mathcal{F}(g+h) &= \mathcal{F}(g)+\mathcal{F}(h) \\ \mathcal{F}(kg(x,y)) &= k\mathcal{F}(g(x,y))\end{aligned}$$

反变换

$$g(x,y) = \iint \mathcal{F}(g)e^{i2\pi(ux+vy)}\mathrm{d}u\mathrm{d}v$$

傅里叶基实部示例

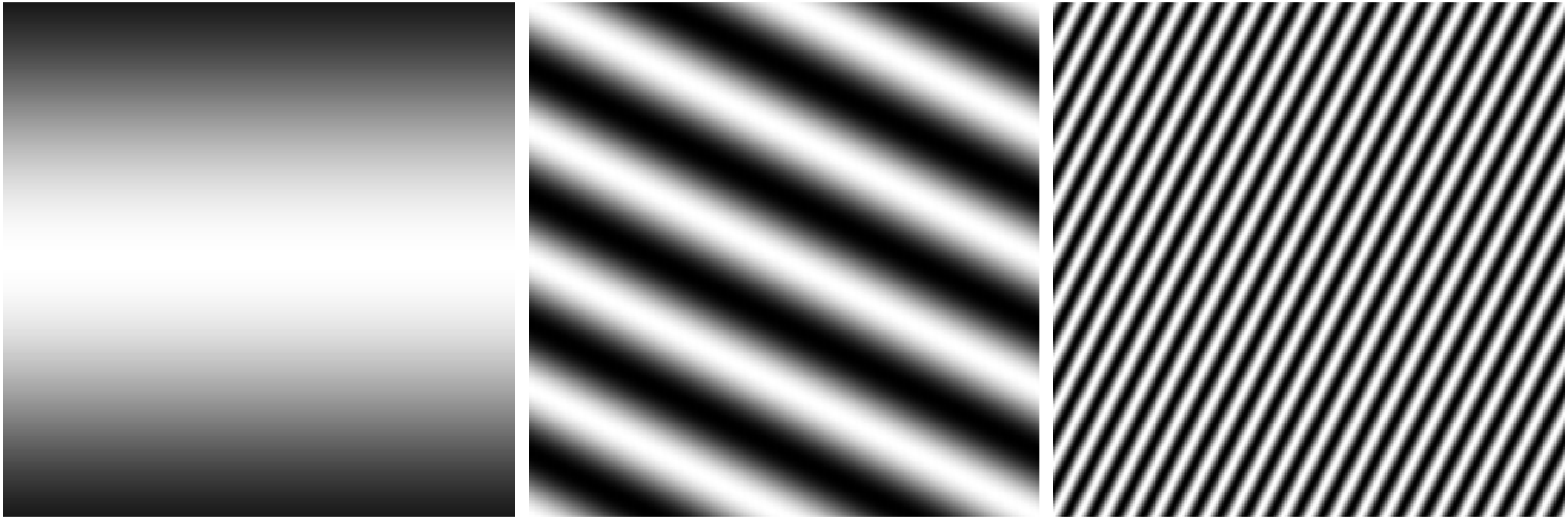


Figure 8.6. The real component of Fourier basis elements, shown as intensity images. The brightest point has value one, and the darkest point has value zero. The domain is $[-1,1]x[-1,1]$, with the origin at the center of the image. On the left, $(u,v) = (0, .4)$; in the center, $(u,v) = (1,2)$ and on the right $(u,v) = (10,-5)$. These are sinusoids of various frequencies and orientations, described in the text.

幅值与相位

$$\begin{aligned}
\mathcal{F}(g(x,y)) &= \iint g(x,y)\cos(2\pi ux + vy)\mathrm{d}x\mathrm{d}y \\
&\quad + i\iint g(x,y)\sin(2\pi ux + vy)\mathrm{d}x\mathrm{d}y \\
&= \Re(\mathcal{F}(g)) + i * \Im(\mathcal{F}(g)) \\
&= \mathcal{F}_R(g) + i * \mathcal{F}_I(g)
\end{aligned}$$

傅里叶变换示例

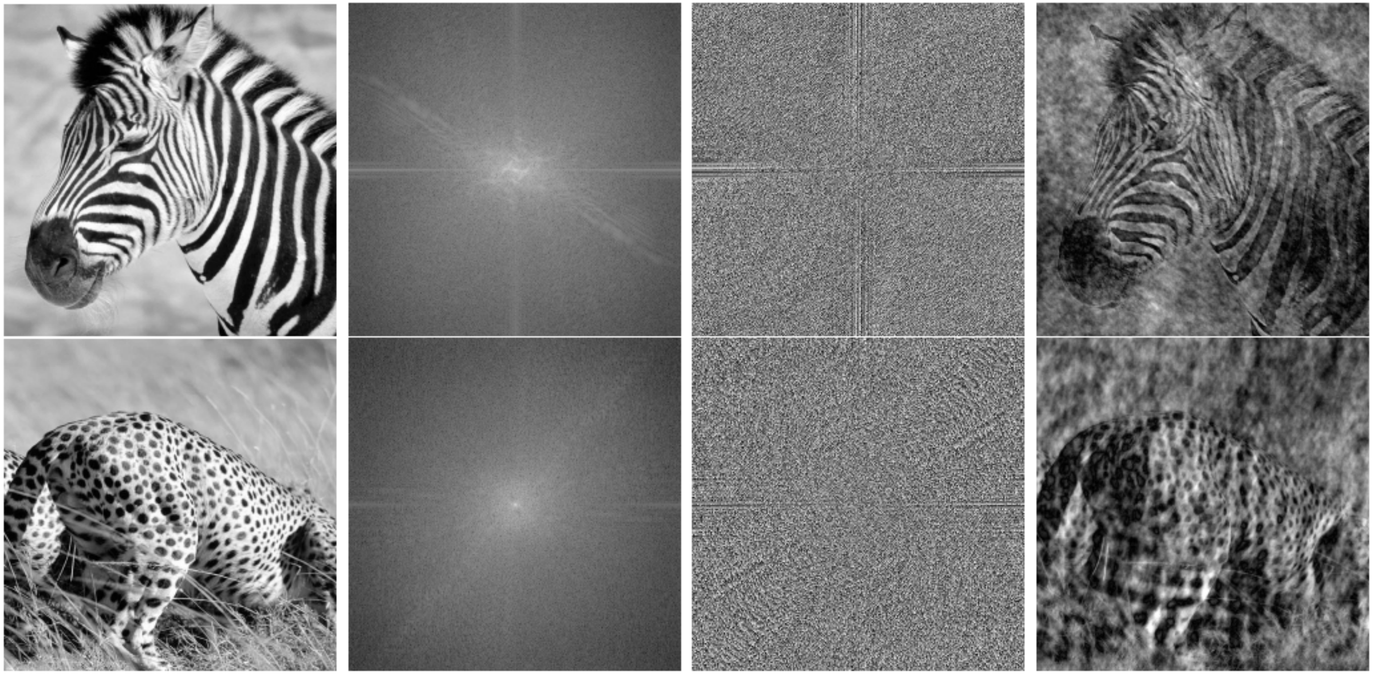
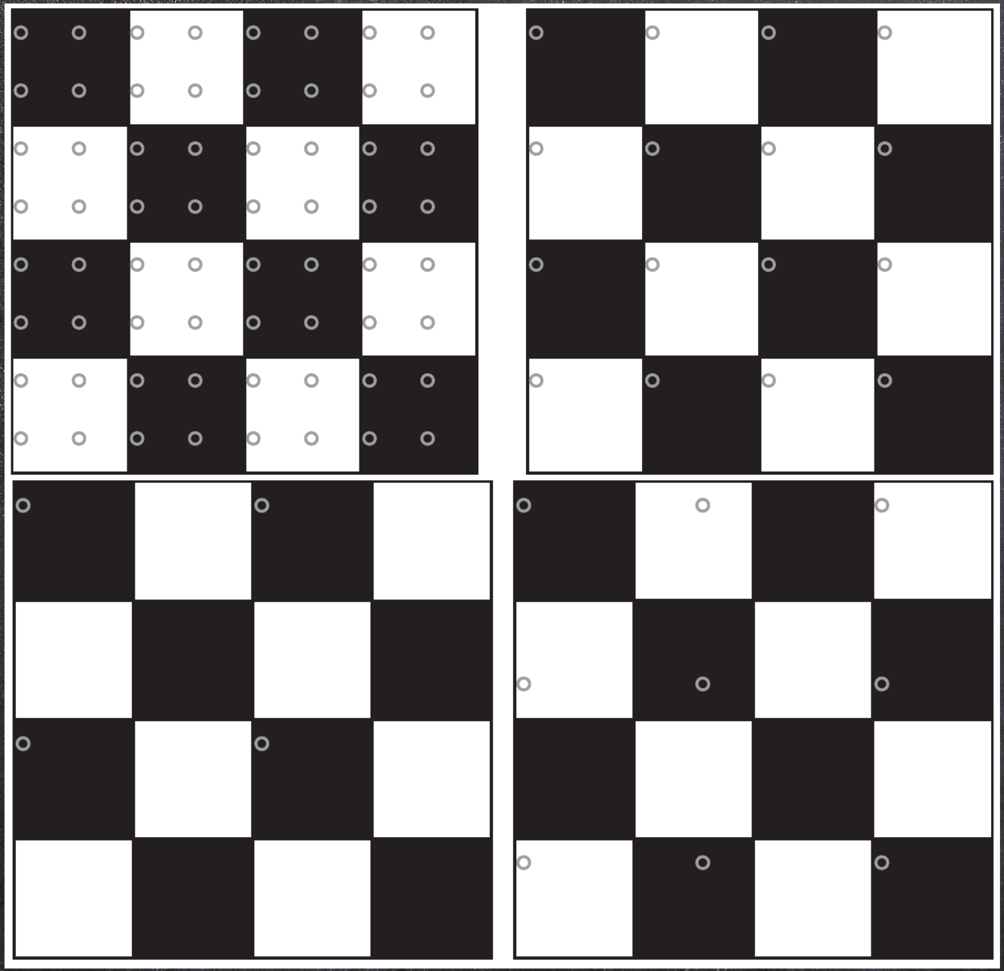


Figure 8.7. The second image in each row shows the log of the magnitude spectrum for the first image in the row; the third image shows the phase spectrum, scaled so that $-\pi$ is dark and π is light. The final images are obtained by swapping the magnitude spectra. While this swap leads to substantial image noise, it doesn't substantially affect the interpretation of the image, suggesting that the phase spectrum is more important for perception than the magnitude spectrum.

采样与失真



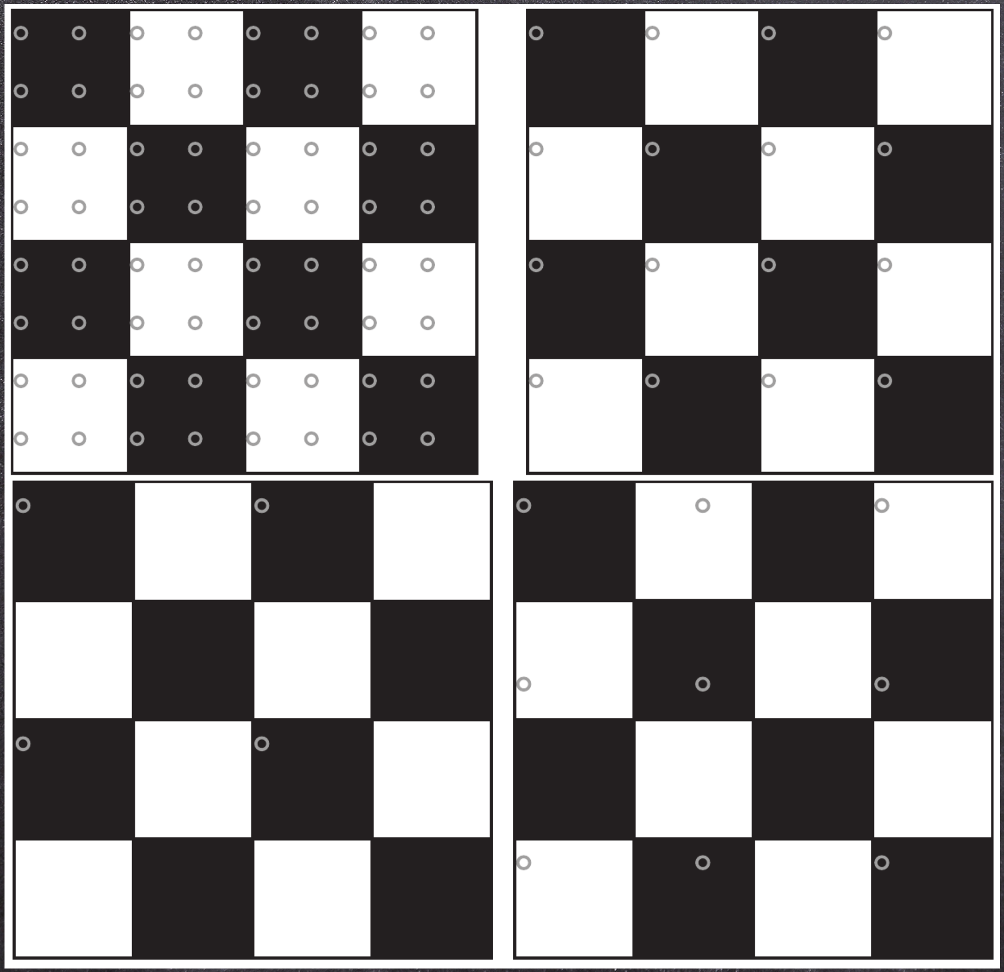


Figure 8.8. The two checkerboards on the top illustrate a sampling pro-

cedure which appears to be successful (whether it is or not depends on some details that we will deal with later). The grey circles represent the samples; if there are sufficient samples, then the samples represent the detail in the underlying function. The sampling procedure shown on the bottom is unequivocally unsuccessful; the samples suggest that there are fewer checks than there are. This illustrates two important phenomena: firstly, successful sampling schemes sample data "often enough"; and, secondly, unsuccessful sampling schemes cause high frequency information to appear as lower frequency information.

一维采样

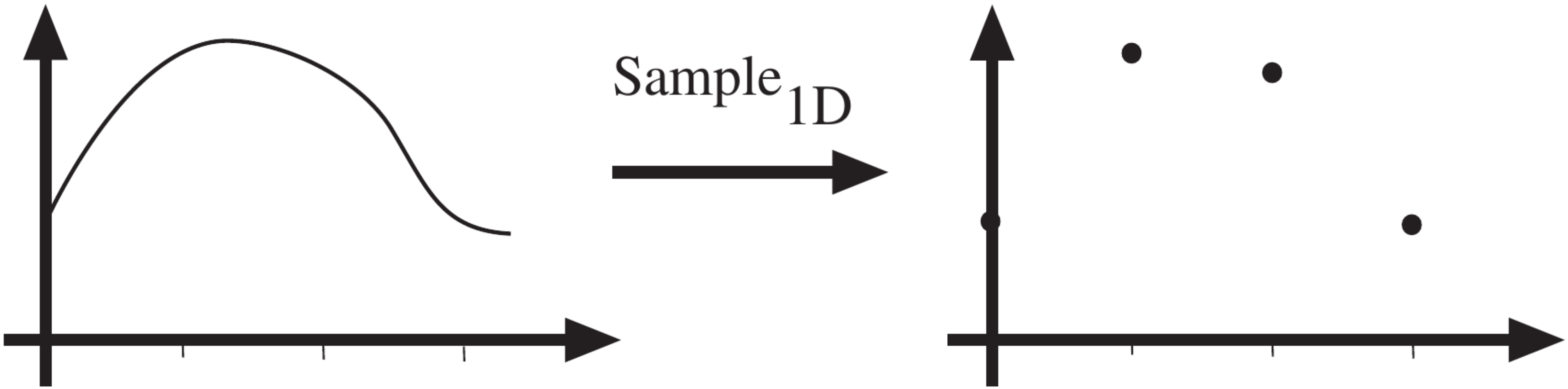


Figure 8.9. Sampling in 1D takes a function, and returns a vector whose elements are values of that function at the sample points, as the top figures show. For our purposes, it is enough that the sample points be integer values of the argument. We allow the vector to be infinite dimensional, and have negative as well as positive indices.

二维采样

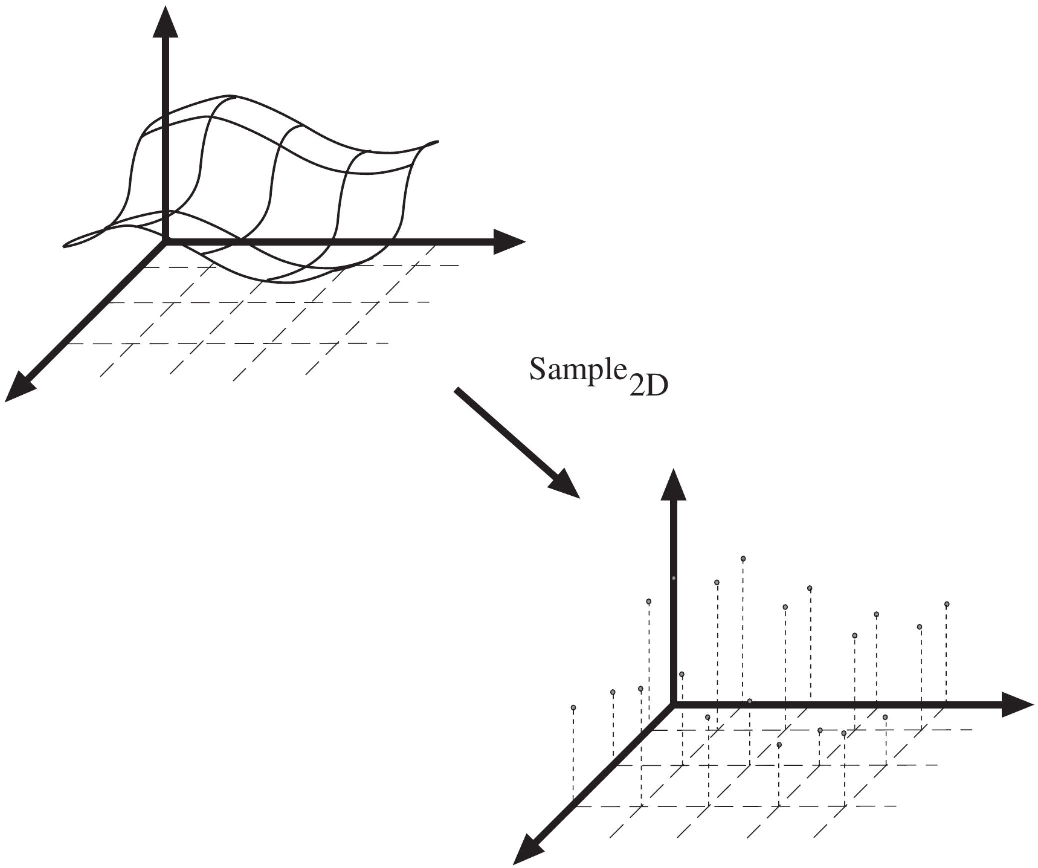


Figure 8.10. Sampling in 2D takes a function and returns an array; again, we allow the array to be infinite dimensional and to have negative as well as positive indices.

采样信号模型

$$
\begin{aligned}
\int_{-\infty}^{\infty} a\delta(x) f(x)\mathrm{d}x &= a\lim_{\varepsilon\to 0}\int_{-\infty}^{+\infty} d_\varepsilon(x) f(x)\mathrm{d}x \\
&= a\lim_{\varepsilon\to 0}\int_{-\infty}^{+\infty} \frac{\mathrm{bar}_\varepsilon(x)}{\varepsilon} f(x)\mathrm{d}x \\
&= a\lim_{\varepsilon\to 0}\sum_{i=-\infty}^{+\infty} \frac{\mathrm{bar}_\varepsilon(x)}{\varepsilon} f(i\varepsilon)\mathrm{bar}_\varepsilon(x-i\varepsilon)\varepsilon \\
&= af(0) \\
\mathrm{sample}_{2D}(f) &= \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)\delta(x-i,y-j) \\
&= \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(x,y)\delta(x-i,y-j)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{F}(\mathrm{sample}_{2D}(f(x,y))) &= \mathcal{F}\left(f(x,y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j) \right) \\
&= \mathcal{F}(f(x,y)) ** \mathcal{F}\left(\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j) \right) \\
&= F(u,v) ** \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j) \\
&= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F(u-i, v-j)
\end{aligned}
$$

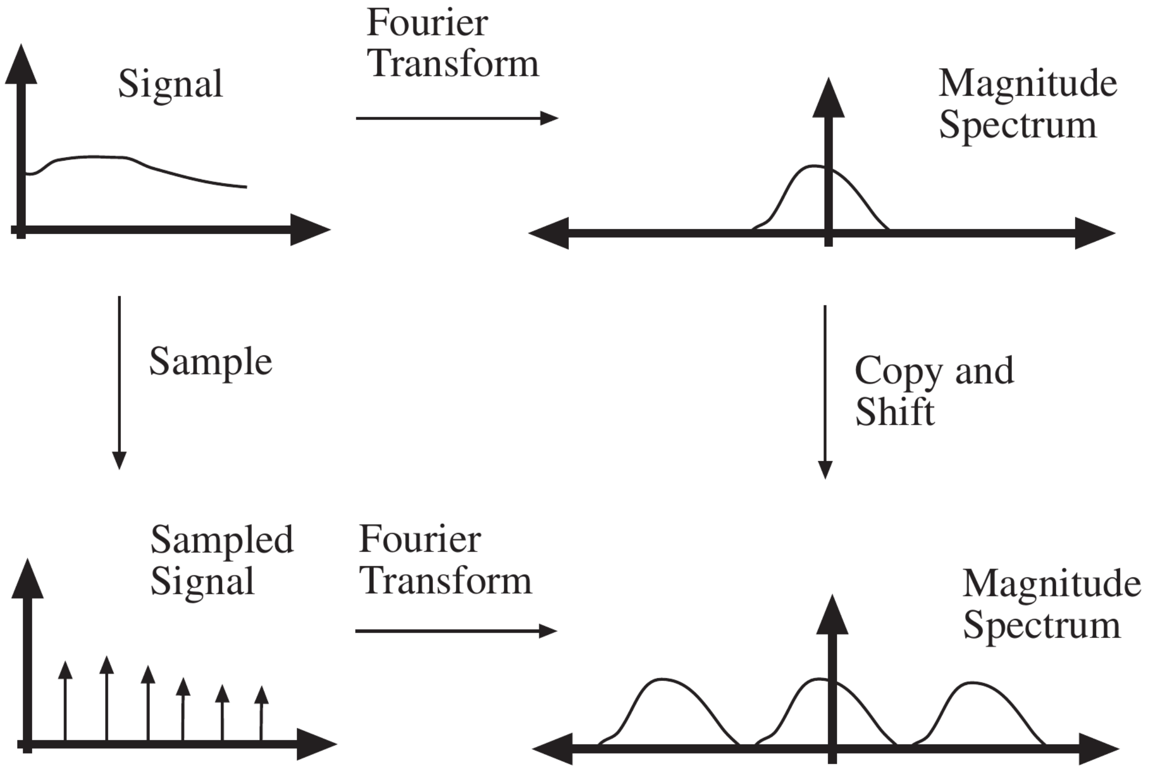


Figure 8.11. The Fourier transform of the sampled signal consists of a sum of copies of the Fourier transform of the original signal, shifted with respect to each other by the sampling frequency. Two possibilities occur. If the shifted copies do not intersect with each other (as in this case), the original signal can be reconstructed from the sampled signal (we just cut out one copy of the Fourier transform, and inverse transform it). If they do intersect (as figure 8.12) the intersection region is added, and so we cannot obtain a separate copy of the Fourier transform, and the signal has aliased.

频域混叠

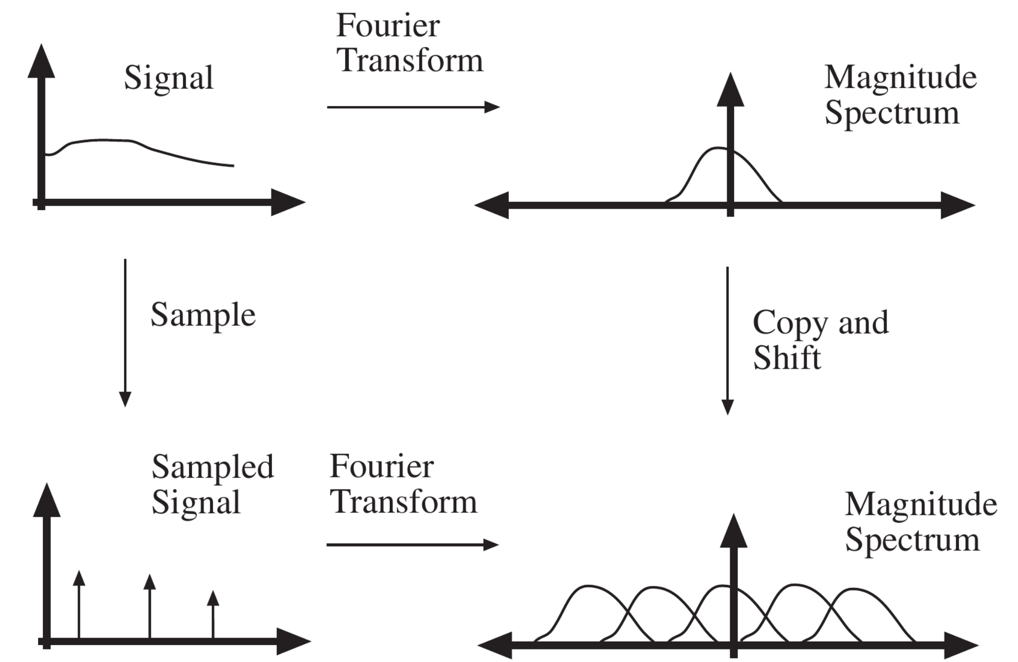
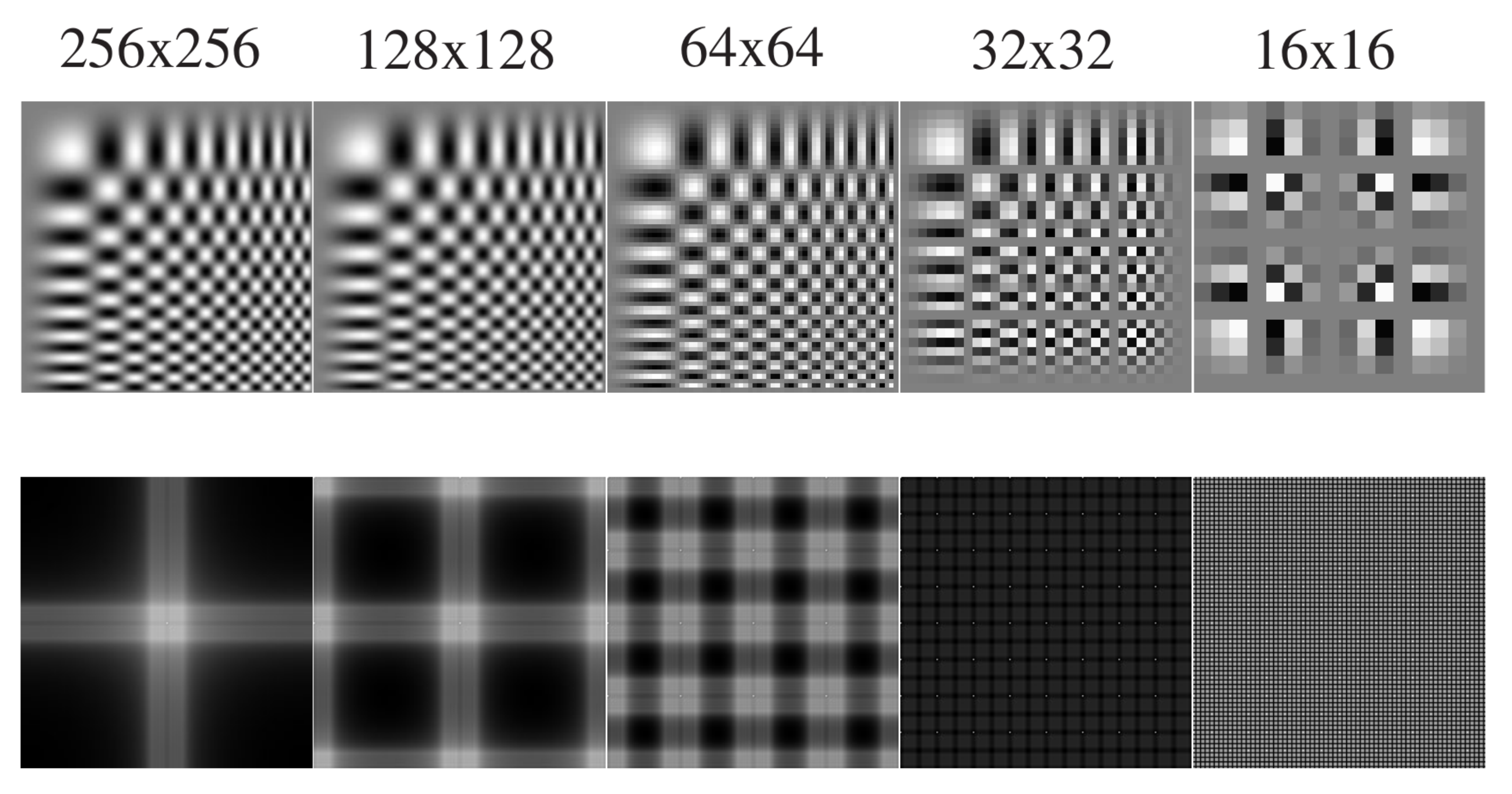


Figure 8.12. The Fourier transform of the sampled signal consists of a sum of copies of the Fourier transform of the original signal, shifted with respect to each other by the sampling frequency. Two possibilities occur. If the shifted copies do not intersect with each other (as in figure 8.11), the original signal can be reconstructed from the sampled signal (we just cut out one copy of the Fourier transform, and inverse transform it). If they do intersect (as in this figure) the intersection region is added, and so we cannot obtain a separate copy of the Fourier transform, and the signal has aliased. This also explains the tendency of high spatial frequencies to alias to lower spatial frequencies.

平采与重采样



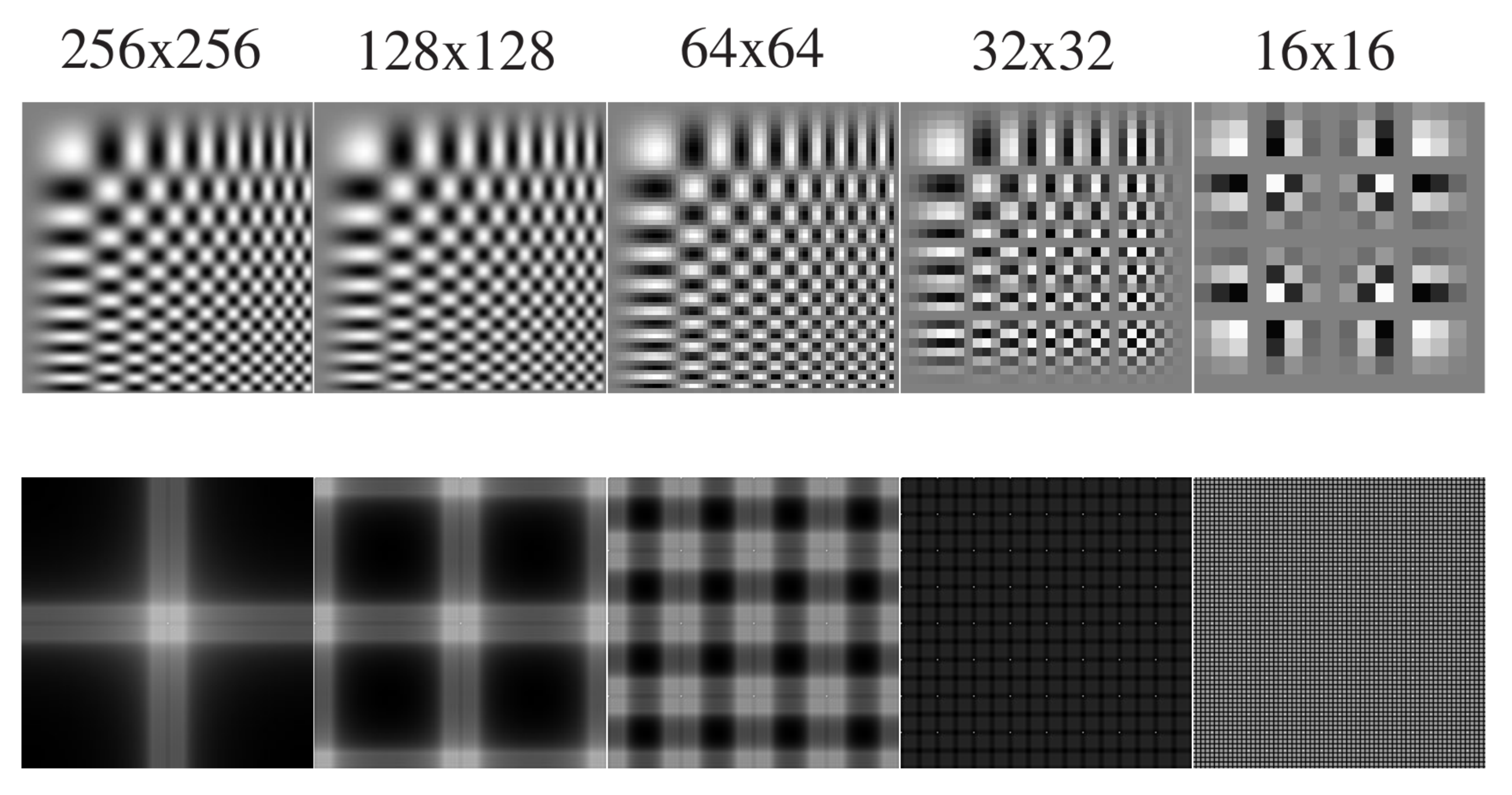


Figure 8.13. The top row shows sampled versions of an image of a grid obtained by multiplying two sinusoids with linearly increasing frequency

— one in x and one in y. The other images in the series are obtained by resampling by factors of two, without smoothing (i.e. the next is a 128x128, then a 64x64, etc., all scaled to the same size). Note the substantial aliasing; high spatial frequencies alias down to low spatial frequencies, and the smallest image is an extremely poor representation of the large image. The bottom row shows the magnitude of the Fourier transform of each image — displayed as a log, to compress the intensity scale. The constant component is at the center. Notice that the Fourier transform of a resampled image is obtained by scaling the Fourier transform of the original image and then tiling the plane. Interference between copies of the original Fourier transform means that we cannot recover its value at some points — this is the mechanism underlying aliasing.

平滑与重采样(高斯平滑)

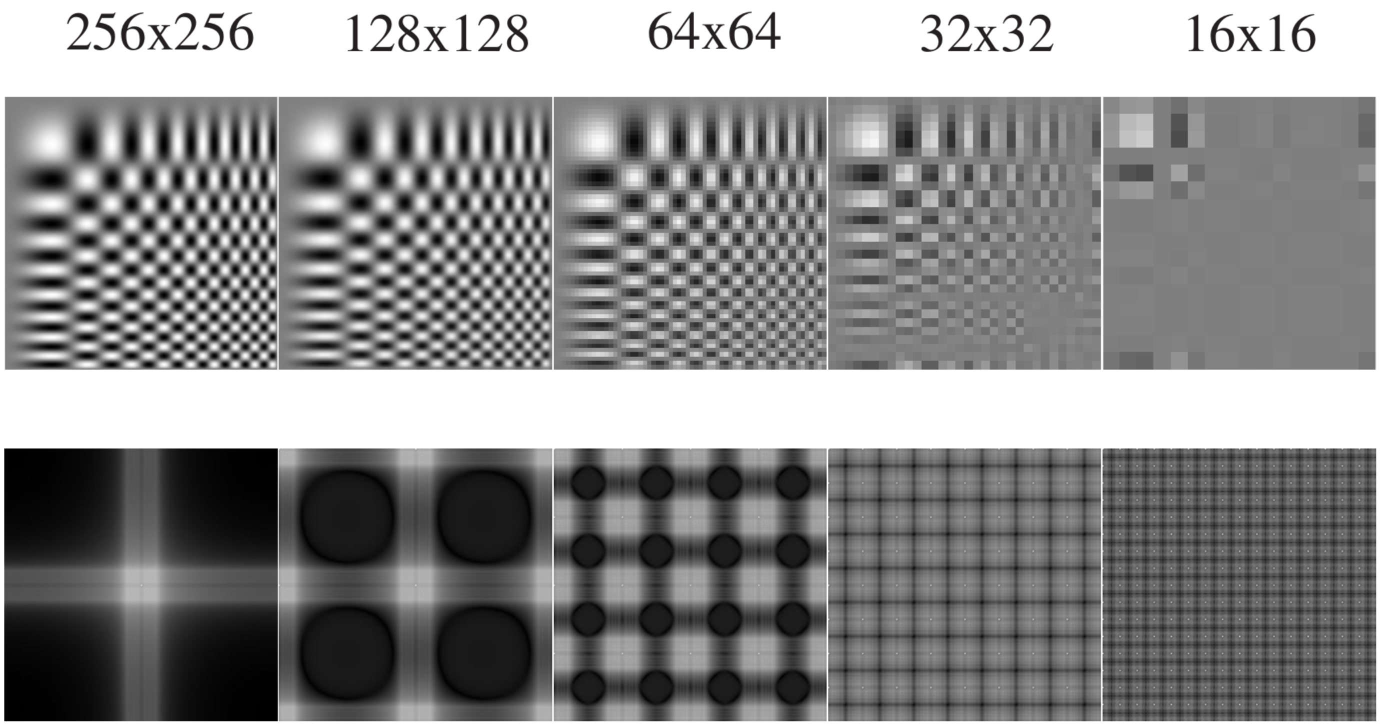


Figure 8.14. Left: Resampled versions of the image of figure 8.13, again by factors of two, but this time each image is smoothed with a Gaussian of σ one pixel before resampling. This filter is a low-pass filter, and so suppresses high spatial frequency components, reducing aliasing. **Right:** The effect of the low-pass filter is easily seen in these log-magnitude images; the low pass filter suppresses the high spatial frequency components so that components interfere less, to reduce aliasing.

平滑与重采样(高斯平滑)

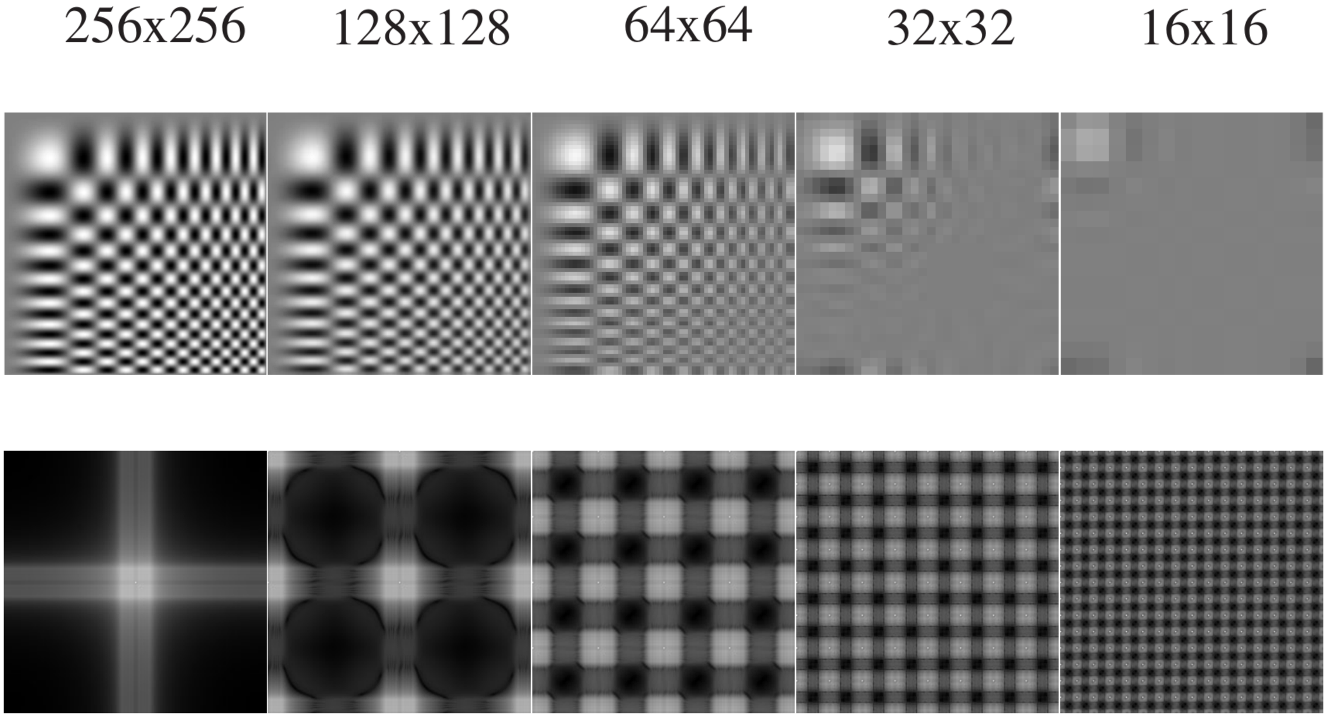


Figure 8.15. Left: Resampled versions of the image of figure 8.13, again by factors of two, but this time each image is smoothed with a Gaussian of σ two pixels before resampling. This filter suppresses high spatial frequency components more aggressively than that of figure 8.14. **Right:** The effect of the low-pass filter is easily seen in these log-magnitude images; the low pass filter suppresses the high spatial frequency components so that components interfere less, to reduce aliasing.

高斯金字塔

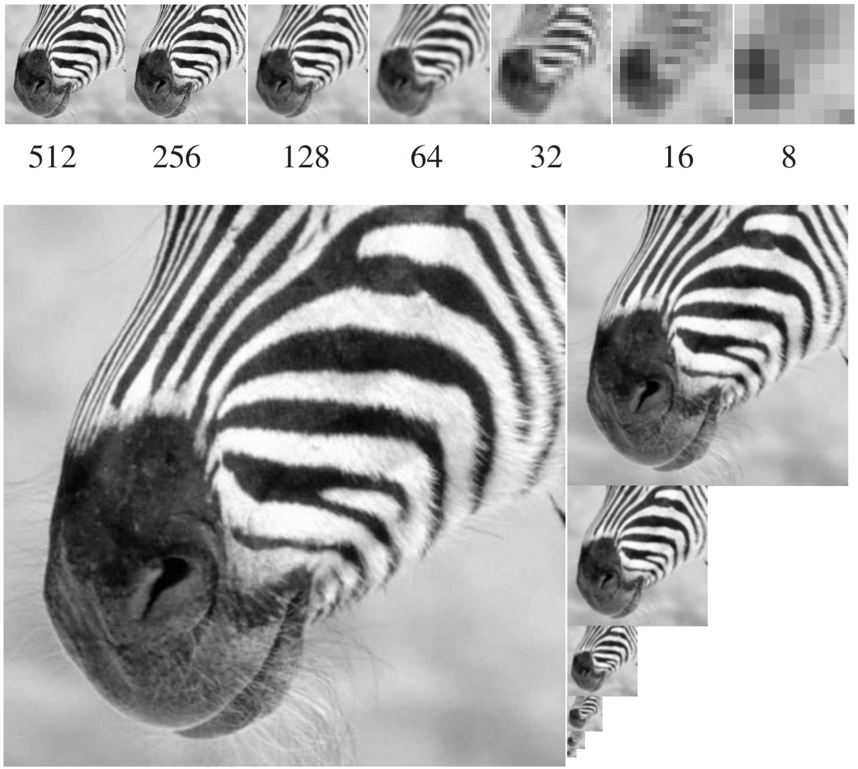


Figure 8.16. A Gaussian pyramid of images, running from 512x512 to 8x8. On the top row, we have shown each image at the same size (so that some have bigger pixels than others), and the lower part of the figure shows the images to scale. Notice that if we convolve each image with a fixed size filter, it will respond to quite different phenomena. An 8x8 pixel block at the finest scale might contain a few hairs; at a coarser scale it might contain an entire stripe; and at the coarsest scale, it contains the animal's nose.

尺度空间

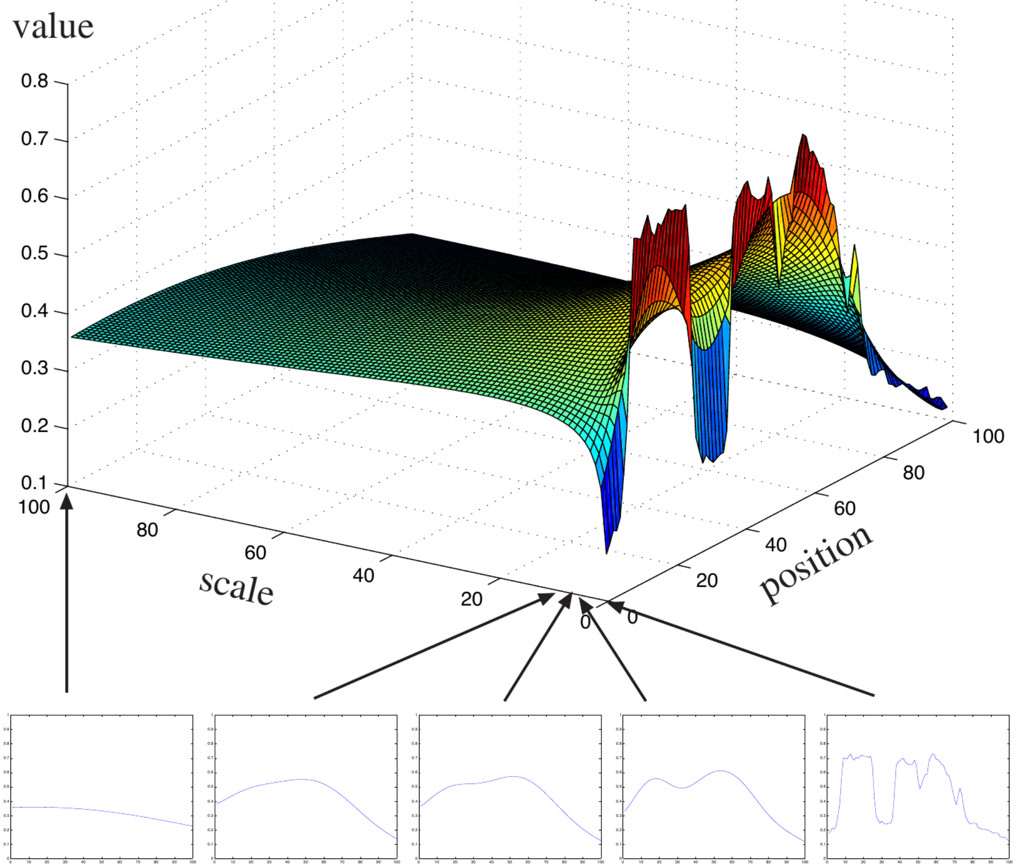


Figure 8.17. At the **top**, a 1D signal smoothed using Gaussian filters of increasing σ, plotted as a surface. As σ gets bigger, the detail is increasingly suppressed. At the **bottom** we show particular versions of the signal at particular scales. As the signal is smoothed, extrema merge and vanish. The smoothest versions of the signal can be seen as an indication of the "overall trend" of the signal, and the finer versions have increasing amounts of detail imposed.

Zero-crossing

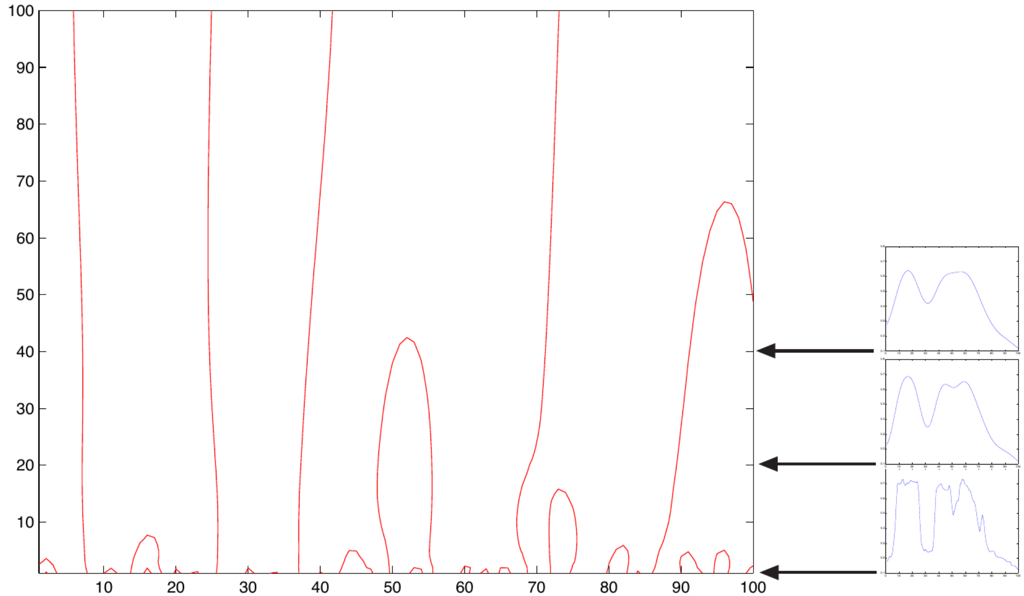


Figure 8.18. One feature of a signal that can be important is the position of fast changes; these can be found by marking the position of zero crossings of the second derivative. This feature develops in an orderly way as the signal is smoothed. The representation shown marks the position of zero crossings of the second derivative of the smoothed signal, as the smoothing increases (again, scale increases vertically). Notice that zero crossings can meet and obliterate one another as the signal is smoothed but no new zero-crossing is created. This means that the figure shows the characteristic structure of either vertical curves or inverted "u" curves. An inverted pitchfork shape is also possible — where three extrema meet and become one — but this requires special properties of the signal; for most signals, this inverted pitchfork shape degenerates to an inverted u next to a vertical curve. Notice also that the position of zero crossings tends to shift as the signal is smoothed.

2D

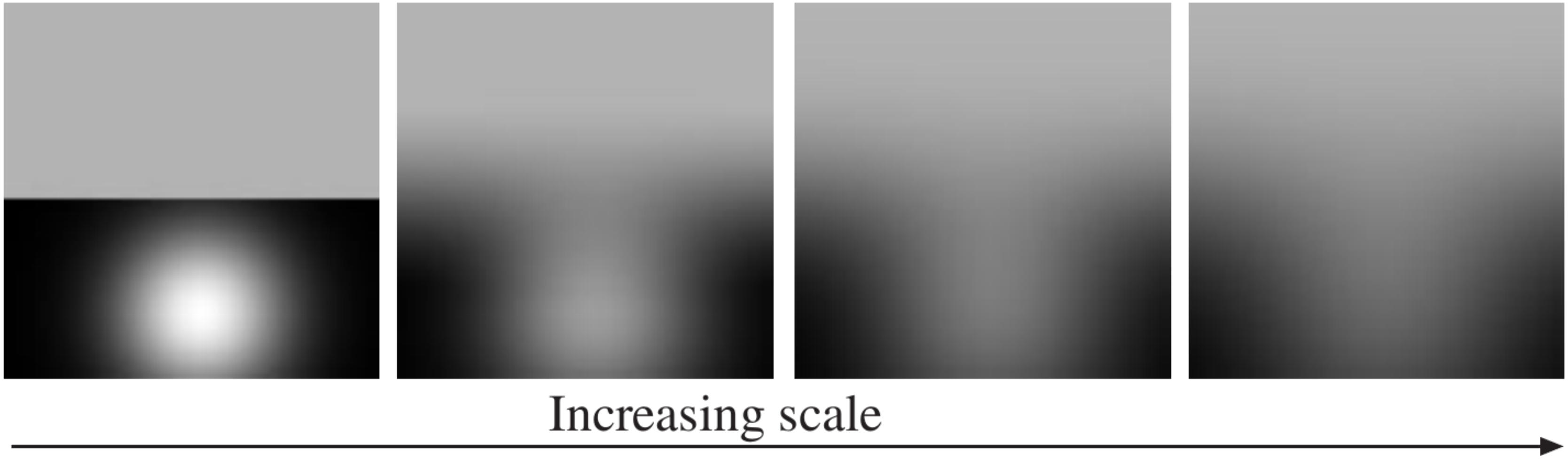


Figure 8.19. Smoothing an image with a symmetric Gaussian cannot create local maxima or local minima of brightness. However, local extrema can be extinguished. What happens is that a local maximum shrinks down to the value of the surrounding pixels. On the left, an image with one local maximum near a region of constant brightness. As the image is smoothed, the blob of brightness is smoothed into this region, and eventually (for some scale value between the last two images) the local maximum disappears. Recording the details of these disappearances — where the maximum that disappears is, the contour defining the blob around the maximum, and the scale at which it disappears — yields a scale-space representation of the image.