CHAPTER 6

Fourier Transforms, Sampling and Aliasing

Thinking of a signal $g(x, y)$ as a weighted sum of a large (or infinite) number of small (or infinitely small) box functions emphasizes that a signal is an element of a vector space. The box functions form a convenient basis, and the weights are coefficients on this basis. Three important and interrelated problems can be addressed by working in a different basis.

- What is lost by sampling a signal?
- What signals can be recovered exactly from samples, and how?
- If a signal cannot be recovered exactly from samples, what form does the error take?

6.1 FOURIER TRANSFORMS

Figure 2.7 implies that sampling errors are related to fast changes in a signal. It is easiest to study these problems by *a change of basis* that makes fast changes in the signal obvious. An appropriate basis is a set of sinusoids, and the signal is represented as an infinite weighted sum of an infinite number of sinusoids.

The change of basis is effected by a *Fourier transform.* Write i for $\sqrt{-1}$, and define the Fourier transform of a 2D signal $g(x, y)$ to be

$$\mathcal{F}(g)(u, v) = \iint_{-\infty}^{\infty} g(x, y)e^{-i2\pi(ux+vy)}dxdy$$

(everything we do here can be done in arbitrary dimension, but there is no need; those who care are likely to be able to fill in the details themselves). Be aware that there are a variety of definitions in the literature, which differ by constants (a $\sqrt{2\pi}$ term moves around from definition to definition, and engineers tend to prefer to write j for $\sqrt{-1}$).

Assume that appropriate technical conditions are true to make this integral exist. It is sufficient for all moments of g to be finite; a variety of other possible conditions are available [?]. For all this to make sense, think of an image as a complex valued functions with zero imaginary component. The Fourier transform takes a complex valued function of x, y and returns a complex valued function of u, v.

For the moment, fix u and v, and consider the meaning of the value of the transform at that point. The exponential can be rewritten

$$e^{-i2\pi(ux+vy)} = \cos(2\pi(ux + vy)) + i\sin(2\pi(ux + vy)).$$

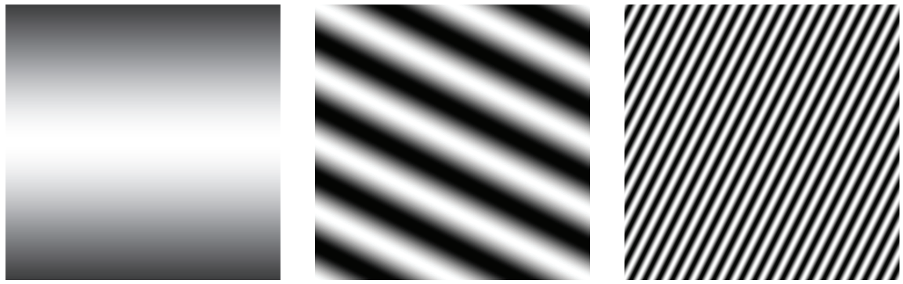

FIGURE 6.1: *The real component of Fourier basis elements shown as intensity images. The brightest point has value one, and the darkest point has value zero. The domain is* $[-1,1] \times [-1,1]$*, with the origin at the center of the image. On the* **left**, $(u,v) = (0, 0.4)$*; in the* **center**, $(u,v) = (1,2)$*; and on the* **right** $(u,v) = (10, -5)$*. These are sinusoids of various frequencies and orientations described in the text.*

These terms are sinusoids on the x, y plane, whose orientation and frequency are given by u, v. For example, consider the real term, which is constant when $ux+vy$ is constant (i.e., along a straight line in the x, y plane whose orientation is given by $\tan\theta = v/u$). The gradient of this term is perpendicular to lines where $ux+vy$ is constant, and the frequency of the sinusoid is $\sqrt{u^2+v^2}$. These sinusoids are often referred to as *spatial frequency components* or *spatial frequencies*; a variety are illustrated in Figure 6.1.

The integral should be seen as a dot product. For fixed u and v, the value of the integral is the dot product between a sinusoid in x and y and the original function. This is a useful analogy because dot products measure the amount of one vector in the direction of another. The value of the transform at a particular u and v can be seen as measuring the amount of the sinusoid with given frequency and orientation in the signal. The transform takes a function of x and y to the function of u and v whose value at any particular (u,v) is the amount of that particular sinusoid in the original function. This view justifies the model of a Fourier transform as a change of basis.

The Fourier transform is linear:

$$\mathcal{F}(g+h) = \mathcal{F}(g) + \mathcal{F}(h)$$

and for k any constant

$$\mathcal{F}(kg) = k\mathcal{F}(g).$$

It is useful to recover a signal $g(x,y)$ from its Fourier transform $\mathcal{F}(g)(u,v)$. This is another change of basis with the form

$$g(x,y) = \iint_{-\infty}^{\infty} \mathcal{F}(g)(u,v) e^{i2\pi(ux+vy)} du dv.$$

Proving that this inverse works requires a fair amount of ducking and weaving to

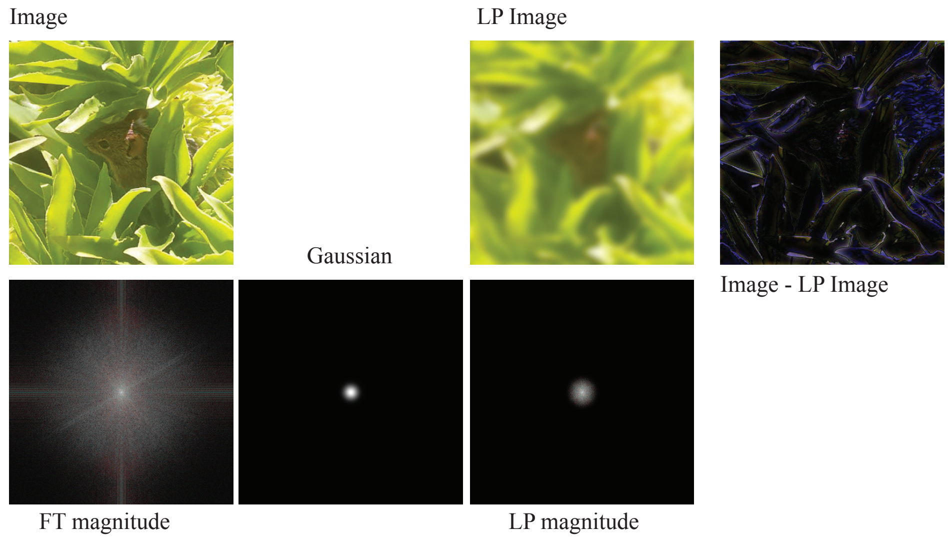

FIGURE 6.2: *On the* **top left**, *the image of a four striped grass mouse with the log magnitude of its Fourier transform on the* **bottom left**. **Center left**, *the gaussian with* $\sigma = 10$ *in* u, v *space. This is multiplied by the weights, and the log magnitude of the result appears* **center right**. **Above** *this is the image obtained by inverting the Fourier transform – equivalently, the low pass filtered image.* **Far left** *shows the high pass filtered image, obtained by subtracting the low pass filtered image from the original. I have* not *shown the log magnitude of the high pass filtered image, because scaling makes the result quite difficult to interpret (it doesn't look filtered). The low pass filtered version is heavily blurred, because only the lowest spatial frequencies appear in the result. Note the high pass filtered version contains what is missing from the low pass version, so has few large values which appear at edges.* Image credit: *Figure shows my photograph, taken at Kirstenbosch and Long Beach respectively.*

do with limits and function spaces, and I will omit a proof (you could look one up in []).

6.1.1 Filtering with a Fourier Transform

One obvious use of a Fourier transform is to change the amount of different spatial frequencies in an image. Do this by multiplying the Fourier transform by some set of weights, then applying an inverse Fourier transform to the result. The easiest case – which will prove fruitful later – is to use weights that are large around $(u, v) = (0, 0)$ and fall off as the frequency increases. A natural choice is a gaussian *in spatial frequency space.* Write

$$g_\sigma(u, v) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(u^2 + v^2)}{2\sigma^2}\right).$$

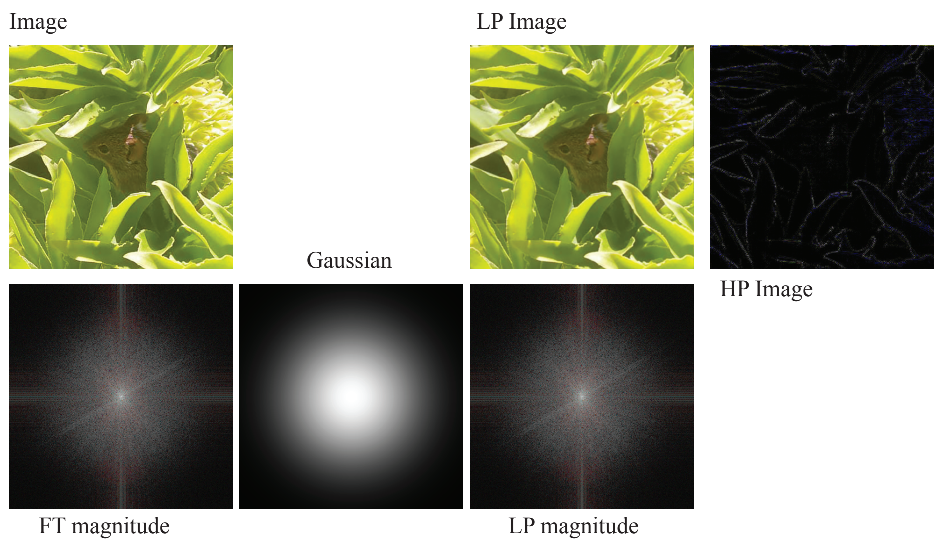

FIGURE 6.3: *On the* **top left**, *the image of a four striped grass mouse with the log magnitude of its Fourier transform on the* **bottom left**. **Center left**, *the gaussian with* $\sigma = 100$ *in* u, v *space. This is multiplied by the weights, and the log magnitude of the result appears* **center right**. **Above** *this is the image obtained by inverting the Fourier transform – equivalently, the low pass filtered image.* **Far left** *shows the high pass filtered image, obtained by subtracting the low pass filtered image from the original. I have* not *shown the log magnitude of the high pass filtered image, because scaling makes the result quite difficult to interpret (it doesn't look filtered). The low pass filtered version is less heavily blurred than that in Figure 6.2, because only the lowest spatial frequencies appear in the result. Note the high pass filtered version contains what is missing from the low pass version, so has very few large values which appear at edges.* Image credit: *Figure shows my photograph, taken at Kirstenbosch and Long Beach respectively.*

If σ is small, then the result of this process should have only low spatial frequencies, which will make it look blurry. The image has had a *low pass filter* applied. An alternative is to multiply the Fourier transform by $(1 - g_\sigma(u, v))$, which will yield an image of only high spatial frequencies (a *high pass filter*). Figure 6.2 and 6.3 show the results. Your suspicion of a strong relationship between multiplying the Fourier transform with a gaussian and convolving the image with a gaussian is correct.

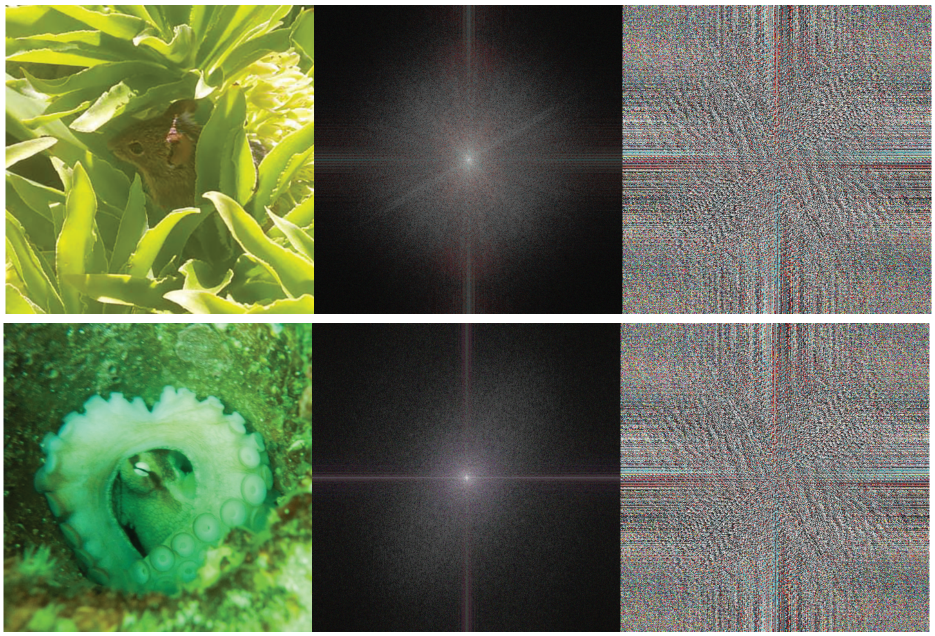

FIGURE 6.4: *On the* **left**, *images of a four striped grass mouse and an octopus;* **center**, *the log magnitude of the Fourier coefficients of the corresponding image, shown in a coordinate system where* $(0, 0)$ *is at the center of the image;* **right**, *the phase of the Fourier coefficients. The magnitude image appears monochrome because magnitudes in each color channel tend to be very similar. The phase appears colored because the phases in the color channels tend to be different. Notice that the magnitude images look quite similar, and that the phases are hard to interpret.* Image credit: *Figure shows my photographs, taken at Kirstenbosch and Long Beach respectively.*

6.1.2 Phase and Magnitude

The Fourier transform consists of a real and a complex component:

$$\begin{aligned}\mathcal{F}(g(x,y))(u,v) &= \int\int_{-\infty}^{\infty} g(x,y)\cos(2\pi(ux+vy))dxdy + \\ &\quad i\int\int_{-\infty}^{\infty} g(x,y)\sin(2\pi(ux+vy))dxdy \\ &= \Re(\mathcal{F}(g)) + i * \Im(\mathcal{F}(g)) \\ &= \mathcal{F}_R(g) + i * \mathcal{F}_I(g).\end{aligned}$$

It is usually inconvenient to draw complex functions of the plane. One solution is to plot $\mathcal{F}_R(g)$ and $\mathcal{F}_I(g)$ separately; another is to consider the *magnitude* and *phase* of the complex functions, and to plot these instead. These are then called the *magnitude spectrum* and *phase spectrum*, respectively.

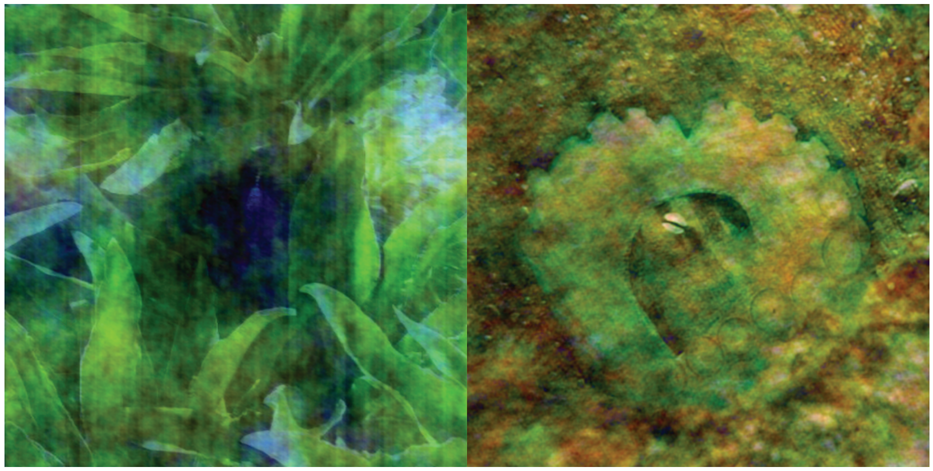

FIGURE 6.5: *Magnitudes of images tend to be the same, and most information is conveyed by phase. This is easily shown by swapping phase and magnitude for two images, applying an inverse, and looking at the result. This figure uses the images of Figure 6.4. On the* **left**, *the phase comes from the mouse and the magnitude from the octopus; on the* **right**, *the phase comes from the octopus and the magnitude from the mouse. Although 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.*

The value of the Fourier transform of a function at a particular u, v point depends on the whole function. This is obvious from the definition because the domain of the integral is the whole domain of the function. It leads to some subtle properties, however (below). The magnitude spectra of images tends to be similar. This appears to be a fact of nature, rather than something that can be proven axiomatically. As a result, the magnitude spectrum of an image is surprisingly uninformative (see Figure 6.5 for an example). Fourier transforms are known in closed form for a variety of useful cases; a large set of examples appears in **?**. We list a few in Table 6.1 for reference.

Table 6.1 contains mostly easy statements, made for reference and to save time. A few lines (2, 4, 5, 9, 12) require some care, and should be assumed true. Others are easy to derive assuming the form of the transform, that the integral exists, and so on (exercises).

There are a number of facts below the surface. Write `swap` for the operation that swaps first and second arguments. Then

$$\mathcal{F}(f \cdot \texttt{swap}) = \mathcal{F}(f) \cdot \texttt{swap}$$

(use line 12). This means that

$$\mathcal{F}(\frac{\partial f}{\partial y}) = v\mathcal{F}(u, v)$$

TABLE 6.1: *Some useful Fourier transform pairs.*

Function	Fourier transform	Tag
$f(x,y)$	$\iint\limits_{-\infty}^{\infty} f(x,y)e^{-i2\pi(ux+vy)}dxdy = \mathcal{F}(f)(u,v)$	1
$\iint\limits_{-\infty}^{\infty} \mathcal{F}(f)(u,v)e^{i2\pi(ux+vy)}dudv = f(x,y)$	$\mathcal{F}(f)(u,v)$	2
$\frac{\partial f}{\partial x}(x,y)$	$u\mathcal{F}(f)(u,v)$	3
$0.5\delta(x+a,y)+0.5\delta(x-a,y)$	$\cos 2\pi au$	4
$\cos 2\pi ax$	$0.5\delta(u+a,v)+0.5\delta(u-a,v)$	5
$e^{-\pi(x^2+y^2)}$	$e^{-\pi(u^2+v^2)}$	6
$box_1(x,y)$	$\frac{\sin u}{u}\frac{\sin v}{v}$	7
$f(ax,by)$	$\frac{\mathcal{F}(f)(u/a,v/b)}{ab}$	8
$\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)$	$\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(u-i,v-j)$	9
$f(x-a,y-b)$	$e^{-i2\pi(au+bv)}\mathcal{F}(f)$	10
$f(x\cos\theta-y\sin\theta, x\sin\theta+y\cos\theta)$	$\mathcal{F}(f)(u\cos\theta-v\sin\theta, u\sin\theta+v\cos\theta)$	11
$(f*g)(x,y)$	$\mathcal{F}(f)\mathcal{F}(g)(u,v)$	12

(use line 3).

Use line 11 and line 2 to get

$$\mathcal{F}(\mathcal{F}(f)) = f(-x,-y)$$

and notice that this means that, in principle, an inverse Fourier transform isn't really required (you could just Fourier transform twice, then rotate the resulting function).

6.1.3 Practical Details

Fourier transforms are an extremely helpful conceptual device, and can be very powerful computational tools, but need to be approached with caution because a given Fourier transform coefficient *depends on the entire image*. Changing one pixel in an image will change some, but not all, results of a convolution with that image because convolution is local – only a window of pixels affects the results. But change one pixel in an image, and you change the whole Fourier transform.

There is a version of the Fourier transform that maps discrete signals to discrete signals. This version applies to a discrete signal where only the values at the sample points $[1, 2, \ldots, N]$ are non-zero. The Fourier transform is linear, and so is the discrete version. Viewing the Fourier transform as a change of basis should suggest that the discrete Fourier transform can be represented as multiplication by an $N \times N$ complex matrix; this is correct. However, discrete Fourier transforms can be computed very much faster than by routine matrix multiplication by careful management of intermediate values, justifying the name *fast Fourier transform* or, almost always, *FFT*. Details are out of scope. Mostly, the FFT can be treated as a Fourier transform, but there are some important details to keep track of. The change of basis description should suggest to you that an $N \times N$ image will have an $N \times N$ Fourier transform, and this is the case.

For most people, it is "natural" to think of the spatial frequency where $(u, v) = (0, 0)$ as lying at the center of the image, with u and v running from negative to positive values from left to right and bottom to top. For computational reasons, most API's report the FFT of an image in a rather odd coordinate system where the highest spatial frequencies are at the center and the lowest ones are at the corners. If your API does this, it will also have some form of shift command that changes the coordinate system.

It is much more usual to think in terms of magnitudes and phases rather than real and imaginary components of the complex values of the transform. This is mostly because the magnitude of the FFT at u, v can be interpreted as "how much" of that spatial frequency is present. Finally, the magnitude of a Fourier transform tends to have quite large dynamic range, and it is usual to show pictures of log magnitude (actually $\log(\mathsf{abs}(z) + 1)$, to avoid problems with small numbers) rather than magnitude.

6.1.4 The Convolution Theorem and the Support of Filters

The *convolution theorem* (line 12 of Table 43.1) says convolution in the signal domain is the same as multiplication in the Fourier domain. This makes it possible to visualize the effect of a linear filter in the Fourier domain. Further, it makes it possible to think about what interpolation does to a signal (recall Section 5.3.3 establishes that interpolation is a convolution). Because the inverse Fourier transform is a Fourier transform (up to a flip, above), the convolution theorem works both ways. Multiplication in the signal domain is the same as convolution in the

Fourier domain.

One application of the convolution theorem illustrates some possible difficulties building filters. Write

$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{\left[x^2+y^2\right]}{2\sigma^2}\right)}$$

then

$$\mathcal{F}(g_\sigma(x,y)) = C g_{\frac{1}{2\pi\sigma}}(u,v)$$

(where the constant C depends on σ – work out the details in the exercises). There is a big point here: a gaussian that is spread out in x, y is concentrated in u, v, and vice versa. This is a rather distant manifestation of Heisenberg's uncertainty principle. Now consider building a low pass filter that accepts a very small range of spatial frequencies. This could be modelled as multiplying the Fourier transform of the image by a gaussian with very small σ. The convolution kernel that implements this filter is the inverse Fourier transform of this gaussian – which has very large σ. You would need a very large convolution to implement this filter without further tricks.

Here is a trick, which relies on the gaussian pyramid of Section 2.3.4. Recall that

$$g_{\sigma_1} * g_{\sigma_2} = g_{\sqrt{\sigma_1^2+\sigma_2^2}}$$

(exercises). This means that to convolve with a very large gaussian, you could convolve with a small one repeatedly, which is one use of the gaussian pyramid. Each layer of the gaussian pyramid is obtained by convolving the previous layer with a gaussian, then downsampling. For the moment, ignore the downsampling, and write $\mathcal{I}$ for the image. Then layer 0 is $\mathcal{I}$ and layer N is $g_\sigma * g_\sigma * \ldots * \mathcal{I}$ which is the same as $g_{\sigma\sqrt{N}} * \mathcal{I}$. Downsampling doesn't really affect this argument (which is why I omitted it), but just makes the convolution more efficient by removing redundant values.

These scaling effects are interesting for more than just gaussians. Imagine you wish to find large stripes in a large image (which you could apply a large convolution kernel to that image). A natural strategy is to downsample both kernel and image, and apply the small version of the kernel to the small image. Further, you could find many different sizes of stripe efficiently by applying one stripe filter to each layer of a gaussian pyramid. Responses at the early layers give fine stripes, and at the later layers give coarse stripes.

Line 8 of the table together with the convolution theorem supports this idea. Imagine you have a filter $f(x,y)$ that detects a small pattern. Then (say) $f(x/10, y/10)$ will detect a larger version of this pattern. Now line 8 shows that the Fourier transform of this new scaled filter will shrink by a factor of 10 in u, v space, so the value depends on only low spatial frequencies. In turn, not much will be lost if you apply the scaled filter to a low pass filtered version of the image. Further, applying the scaled filter to a low pass filtered version of the image will be equivalent to applying the original filter to a scaled version of the image (line 8 again). But this is equivalent to applying the original filter to a downsampled layer of the gaussian pyramid.

6.2 SAMPLING AND ALIASING

The crucial reason to discuss Fourier transforms is to get some insight into the difference between discrete and continuous images. In particular, it is clear that some information has been lost when we work on a discrete pixel grid, but what? A good, simple example comes from an image of a checkerboard, and is given in Figure 2.7. The problem has to do with the number of samples relative to the function; we can formalize this rather precisely given a sufficiently powerful model.

6.3 SAMPLING AND ALIASING

Sampling involves a loss of information. As this section shows, a signal sampled too slowly is misrepresented by the samples; high spatial frequency components of the original signal appear as low spatial frequency components in the sampled signal—an effect known as *aliasing*.

6.3.1 The Fourier Transform of a Sampled Signal

As Section 5.3.3 showed, an appropriate continuous model of a sampled signal consists of a δ-function at each sample point weighted by the value of the sample at that point. We can obtain this model by multiplying the sampled signal by a set of δ-functions, one at each sample point. In one dimension, a function of this form is called a *comb function* (because that's what the graph looks like). In two dimensions, a function of this form is called a *bed-of-nails function* (for the same reason). By the convolution theorem, the Fourier transform of this product is the convolution of the Fourier transforms of the two functions. This means that the Fourier transform of a sampled signal is obtained by convolving the Fourier transform of the signal with another bed-of-nails function.

Now convolving a function with a shifted δ-function merely shifts the function (see exercises). This means that the Fourier transform of the sampled signal is the sum of a collection of shifted versions of the Fourier transforms of the signal, that is,

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

where we have written the Fourier transform of $f(x,y)$ as $F(u,v)$.

If the support of these shifted versions of the Fourier transform of the signal does not intersect, reconstructing the signal from the sampled version is straightforward. Take the sampled signal, Fourier transform it, and cut out one copy of the Fourier transform of the signal and Fourier transform this back (Figure 43.8).

However, if the support regions *do* overlap, we are not able to reconstruct the

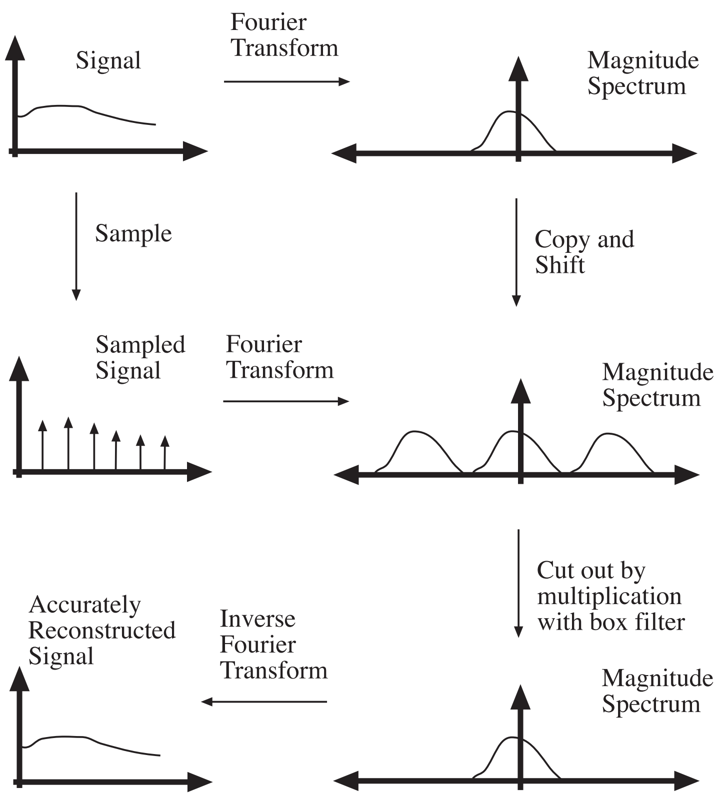

FIGURE 6.6: *The Fourier transform of the sampled signal consists of a sum of copies of the Fourier transform of the original signal, shifted with respect to each other by the sampling frequency. Two possibilities occur. If the shifted copies do not intersect with each other (as in this case), the original 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 in Figure 43.9), the intersection region is added, and so we cannot obtain a separate copy of the Fourier transform, and the signal has aliased.*

signal because we can't determine the Fourier transform of the signal in the regions of overlap, where different copies of the Fourier transform will add. This results in a characteristic effect, usually called *aliasing*, where high spatial frequencies appear to be low spatial frequencies (see Figure 43.10 and exercises). Our argument also yields *Nyquist's theorem*: the sampling frequency must be at least twice the highest frequency present for a signal to be reconstructed from a sampled version. By the same argument, if we happen to have a signal that has frequencies present only in the range $[2k-1\Omega, 2k+1\Omega]$, then we can represent that signal exactly if we sample at a frequency of at least 2Ω.

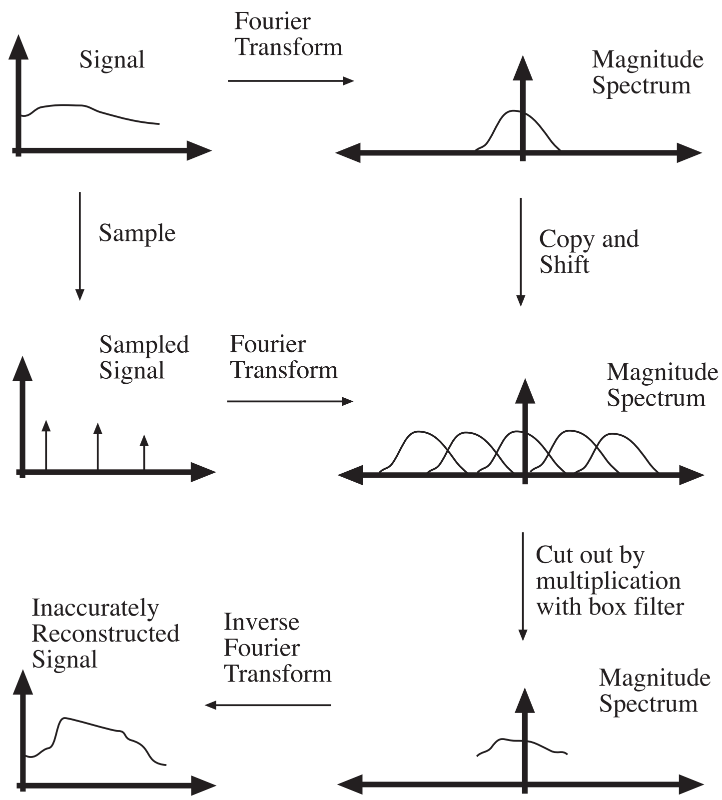

FIGURE 6.7: *The Fourier transform of the sampled signal consists of a sum of copies of the Fourier transform of the original signal, shifted with respect to each other by the sampling frequency. Two possibilities occur. If the shifted copies do not intersect with each other (as in Figure 43.8), the 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.*

6.3.2 Smoothing and Resampling

Nyquist's theorem means it is dangerous to shrink an image by simply taking every kth pixel (as Figure 43.10 confirms). Instead, filter the image so that spatial frequencies above the new sampling frequency are removed. You may think you could do this exactly by multiplying the image Fourier transform by a scaled 2D box function, which would act as a low-pass filter. The convolution theorem and Table 43.1 yield that this is equivalent to convolving the image with a kernel of the form $(\sin x \sin y)/(xy)$. This convolution is impossible, because the kernel has infinite support. You should wonder why a convolution that is impossible appears easy in the Fourier domain; it isn't easy in the Fourier domain (exercises).

Assume you wish to halve the width and height of the image. Assume that the sampled image has no aliasing (because if it did, there would be nothing to do about it anyway; once an image has been sampled, any aliasing that is going to occur has happened). This means that the Fourier transform of the sampled image is going to consist of a set of copies of some Fourier transform, with centers shifted to integer points in u, v space.

If this signal is resampled by half, the copies now have centers on the half-integer points in u, v space. This means that avoiding aliasing requires applying a filter that strongly reduces the content of the original Fourier transform outside the range $|u| < 1/2$, $|v| < 1/2$ *before you resample the signal*. This takes a filter whose response is pretty close to constant for some range of low spatial frequencies—the pass band—and whose response is also pretty close to zero—for higher spatial frequencies—the stop band.

A gaussian is a low-pass filter because its response at high spatial frequencies is low and its response at low spatial frequencies is high, so the downsampling process of Section 2.3.3 is justified. In fact, the Gaussian is not a particularly good low-pass filter. It is possible to design low-pass filters that are significantly better than Gaussians. The design process involves a detailed compromise between criteria of ripple—how flat is the response in the pass band and the stop band?—and roll-off—how quickly does the response fall to zero and stay there? Mostly, the advantages of being able to use a gaussian pyramid and the complexities of better filter design mean that, in practice, smoothing for subsampling is done with a gaussian.

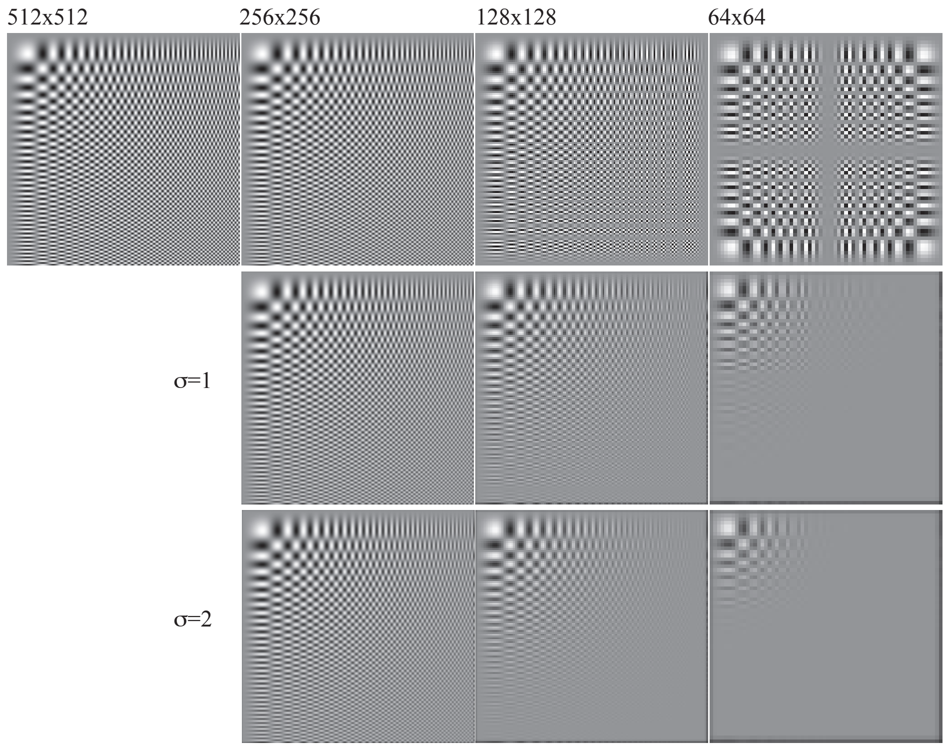

FIGURE 6.8: *The* **top row** *shows sampled versions of an image of a 512x512 grid obtained by multiplying two sinusoids with linearly increasing frequency—one in x and one in y. The other images in the series are obtained by resampling by factors of two without smoothing (i.e., the next is a 256x256, 128x128, then 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* **center row** *shows sampled versions where the image was smoothed by a gaussian with $\sigma = 1$ before downsampling* **i***n each round of downsampling (so the 256x256 is smoothed then downsampled once; the 128x128 is a smoothed and downsampled version of that; and so on). Note the reduction in aliasing (some remains), combined with a loss of information – rather than get the structure of the image wrong, one loses some information. The* **bottom row** *is same as center row, but now the image was smoothed by a gaussian with $\sigma = 2$ before downsampling. There is less aliasing and less information about the original image.*

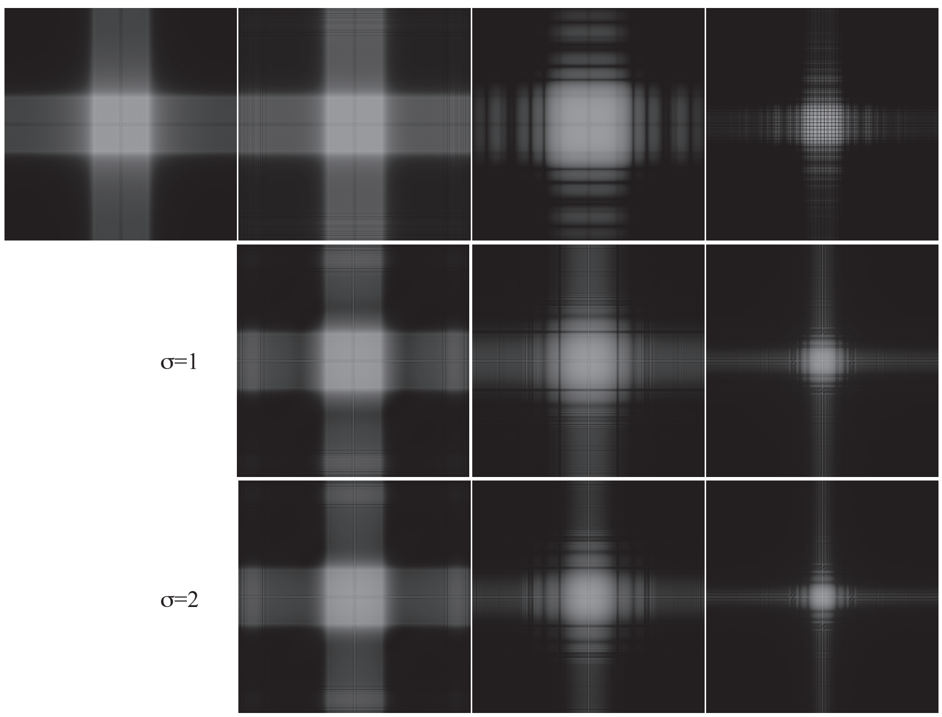

FIGURE 6.9: *Log magnitude of the Fourier transforms of Figure 43.10, showing the effect of the gaussian – reducing high spatial frequencies – and the reduction in aliasing.*