CHAPTER 7

Sampling and Aliasing

We need a new technique to deal with related problems so far left open:

- why does differentiation emphasize noise and smoothing deemphasize it?
- what information is lost passing from a continuous image to a discrete image?
- how can images be downsampled safely?
- are there limits to downsampling accuracy?

All of these problems are related to the presence of fast changes in an image. Derivatives are large at fast changes, and noise produces spurious fast changes; sampling an image might miss fast changes; downsampling is clearly affected by fast changes; and fast changes should impose limits to downsampling accuracy – you can't reasonably expect to represent an 8x8 checkerboard in a 4x4 pixel image.

These effects can be studied by *a change of basis*. We change the basis to be a set of sinusoids and represent the signal as an infinite weighted sum of an infinite number of sinusoids. This means that fast changes in the signal are obvious, because they correspond to large amounts of high-frequency sinusoids in the new basis.

7.1 FOURIER SERIES

For the moment, we work with functions that are defined on the closed unit interval $[0, 1]$ and are *periodic* (so that $f(0) = f(1)$). These can be thought of as functions on the circle. Changing the domain of such a function to another closed interval is straightforward. Pretty much any such function $f(t)$ can be represented as a series

$$A_0 + \sum_{k=1}^{\infty} \left[A_k \cos(2\pi k t) + B_k \sin(2\pi k t) \right].$$

Notice that this representation decomposes the function on a *basis* of sinusoids. The frequency of these sinusoids is k (check this). The coefficients A_k and B_k can be thought of as "how much" of frequency k appears in the function. Straightforward integral identities (Chapter 33.2) yield

$$\begin{aligned} A_0 &= \int_0^1 f(t)dt \\ \frac{A_k}{2} &= \int_0^1 f(t)\cos(2\pi k t)\,dt \\ \frac{B_k}{2} &= \int_0^1 f(t)\sin(2\pi k t)\,dt \end{aligned}$$

.

This representation of a function is known as a *Fourier series*. Notice that we can transform from a periodic function to a set of coefficients (compute the integrals) OR from a set of coefficients to a periodic function (sum the terms). About 150 years of analysis has determined: precisely when and how the series converges (for all cases we care about); and the mild technical condition on $f(t)$ not worth expounding here required for a series representation to exist. For any function we are likely to encounter, we will be able to: compute a series; operate on the coefficients; then compute a new function from the new coefficients. Furthermore, each set of coefficients corresponds to a unique function and each function corresponds to a unique set of coefficients.

Fourier series are interesting because a linear operator that takes a function to a function will have an analog that takes the relevant series to another series. In some kinds of mathematical problem solving, this property can be useful.

Worked example 7.1 *Solving a problem with a Fourier series*

Find a periodic function $f(t)$ on $[0, 1]$ that solves

$$\frac{d^2 f}{dt^2} - f(t) = g(t)$$

Solution: Write

$$\begin{aligned} g(t) &= G_0 + \sum_{k=0}^{N} \left[H_k \cos(2\pi kt) + J_k \sin(2\pi kt)\right] \\ f(t) &= A_0 + \sum_{k=0}^{N} \left[A_k \cos(2\pi kt) + B_k \sin(2\pi kt)\right] \end{aligned}$$

and notice that

$$\frac{d^2 f}{dt^2} - f(t) = -A_0 + \sum_{k=0}^{N} \left[-(2\pi k^2 + 1) A_k \cos(2\pi kt) - (2\pi k^2 + 1) B_k \sin(2\pi kt)\right].$$

Our problem can now be seen as a problem in series. The coefficients of a solution will have the properties

$$\begin{aligned} A_0 &= -G_0 \\ A_k &= -\frac{H_k}{(2\pi k^2 + 1)} \\ B_k &= -\frac{J_k}{(2\pi k^2 + 1)} \end{aligned}$$

(remember, we *know* $g(t)$ so we *know* G_0, H_k, J_k).

Working with sines and cosines is a mild nuisance. A more elegant version series representation uses complex exponentials and represents $f(t)$ as

$$\sum_{k=0}^{\infty} c_k \exp\left[i2\pi kt\right]$$

where $i^2 = -1$. Recall the Euler formula

$$\exp\left[i2\pi kt\right] = \cos\left(2\pi kt\right) + i\sin\left(2\pi kt\right)$$

so these complex exponentials are, in fact, sinusoids. Because for integers k, m,

$$\int_0^1 \exp\left[i2\pi kt\right]\exp\left[-i2\pi mt\right]dt = \begin{cases} 0 & k \neq m \\ 1 & \text{otherwise} \end{cases}$$

we have that

$$c_k = \int_0^1 f(t)\exp\left[-i2\pi kt\right]dt$$

(you should check this carefully). You should think of these complex exponential functions as basis functions or as elements of a basis, and the coefficients c_k are the coefficients of a $f(t)$ represented on this basis. The convenient form of the expression for c_k occurs because the basis is *orthonormal* (compare a rotation matrix). Again, this representation of a function is known as a *Fourier series*. Again, we can transform from a periodic function to a set of coefficients (compute the integrals) OR from a set of coefficients to a periodic function (sum the terms). And again, for any function we are likely to encounter, we will be able to: compute a series; operate on the coefficients; then compute a new function from the new coefficients. Furthermore, each set of coefficients corresponds to a unique function and each function corresponds to a unique set of coefficients. The Fourier series computed for a function $f(t)$ can be written $\mathcal{F}(f(t))$ or $\mathcal{F}(f)$ (we will soon overload this notation!). Notice that taking a Fourier series is a linear operation. You should check that:

- $\mathcal{F}(cf) = c\mathcal{F}(f)$ for any constant c.
- $\mathcal{F}(f + g) = \mathcal{F}(f) + \mathcal{F}(g)$.

Worked example 7.2 *Solving a problem with a complex Fourier series*

Given $g(t)$, a periodic function on $[0, 1]$, find $f(t)$ that solves

$$\frac{d^2 f}{dt^2} - f(t) = g(t)$$

Solution: A fairly compact answer is:

$$f(t) = \sum_{k=0}^{\infty} - \left(\frac{\int_0^1 f(t) \exp\left[-i2\pi kt\right] dt}{2\pi k^2 + 1} \right) \exp\left[i2\pi kt\right]$$

but you should check it's right!

Using complex exponentials as a basis means that the coefficients c_k may themselves be complex. You may be concerned that this makes them somewhat hard to interpret. There are two strategies. You could look at the real and imaginary components separately, so representing $f(t)$ by two series – ($\Re(c_k)$ and $\Im(c_k)$). Alternatively, recall that a complex c_k can be written as $Ae^{i\theta}$, where A is called the *magnitude* of c_k and θ is called the *phase* of c_k. It is often easy to interpret a Fourier series in terms of the *magnitude spectrum* (the magnitude of all c_k's) and the *phase spectrum* (the phase of all c_k).

Worked example 7.3 *Differentiation and Fourier series*

Given $f(t) = \sum_{k=0}^{\infty} c_k \exp\left[i2\pi kt\right]$, find

$$\mathcal{F}(\frac{df}{dt})$$

Solution: Notice that

$$\frac{d \exp\left[i2\pi kt\right]}{dt} = i2\pi k \exp\left[i2\pi kt\right]$$

so that

$$\frac{df}{dt} = \sum_{k=0}^{\infty} c_k \left(i2\pi k\right) \exp\left[i2\pi kt\right]$$

so that differentiation maps the magnitude of c_k to $2\pi k c_k$. Notice how the amount of the high frequency components in $f(t)$ increases. In particular, if we have a poor estimate of c_k for a big k, our estimate of that component of the derivative will be much worse.

> **Remember this:** *A Fourier series maps a periodic function on* $[0, 1]$ *to a series of coefficients that measure the amount of each integer frequency in that function. This exists for all functions we will care about, and there is a corresponding inverse. The process of taking a Fourier series is linear. Fourier series can help solve some kinds of differential equation.*

7.2 FOURIER SERIES TO FOURIER TRANSFORMS

Functions on the whole real line are somewhat different from periodic functions on the unit interval. An example of the kind of behavior that can occur is $\cos t^2$ – this must wiggle faster and faster as $t \to \infty$ and as $t \leftarrow \infty$. Behavior like this isn't possible for a periodic function on the unit interval because the ends have to join up. In turn, this suggests that to represent these functions we may need "more" basis elements than in the case of a Fourier series.

The trick to represent a function on the real line is to use a continuous set of frequencies. So rather than transforming a function to a series – which represents the amount for each of a discrete set of frequencies – we will transform a function to a function. The value of the new function at some point will be the weight of the frequency represented by that point. In particular, we represent $f(t)$ (a function on the real line) by

$$\int_{-\infty}^{\infty} F(u) \exp\left[i 2\pi u t\right] du$$

where $F(u)$ is the weight associated with the frequency u, and is referred to as the *Fourier transform* of $f(t)$. With some lively work on limits, you could establish that

$$\int_{-\infty}^{\infty} \exp\left[i2\pi ut\right] \exp\left[-i2\pi vt\right] dt = \delta(u - v)$$

where $\delta(u - v)$ is a rather odd object. If $u \neq v$, $\delta(u - v) = 0$. We do not worry about the *value* of $\delta(u - v)$ when $u - v = 0$; instead, $\delta(u - v)$ has the property that

$$\int_{-\infty}^{\infty} g(u)\delta(u - v)du = g(v).$$

This object is pretty obviously unlike functions we are used to. It is usually called a *delta function* and belongs to a class of objects sometimes referred to as *generalized functions* or *distributions*. For our purposes, the consequence is that

$$\begin{aligned}
\int_{-\infty}^{\infty} f(t) \exp\left[-i2\pi vt\right] dt &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} F(u) \exp\left[i2\pi ut\right] du\right] \exp\left[-i2\pi vt\right] dt \\
&= \int_{-\infty}^{\infty} F(u) \left[\int_{-\infty}^{\infty} \exp\left[i2\pi ut\right]\right] \exp\left[-i2\pi vt\right] dtdu \\
&= \int_{-\infty}^{\infty} F(u)\delta(u - v)du \\
&= F(v)
\end{aligned}$$

which means that we can recover $F(u)$ from $f(t)$. This process is sometimes called *Fourier transforming* $f(t)$, and we write

$$F(u) = \mathcal{F}(f)$$

(overloading the notation for Fourier series – context will have to tell you what is intended). We have already seen that we can recover $f(t)$ from $F(u)$ (take the integral above), and doing so is known as *inverse Fourier transforming* $F(u)$. For the 1D case, it is usual to refer to the original function as being in the *time domain* or the *signal domain*, and its Fourier transform as being in the *frequency domain*.

You should see the Fourier transform as a mildly more complicated version of a Fourier series. We now allow $f(t)$ to have components at any frequency, rather than just integer frequencies. But not much else has changed. The value of the Fourier transform at u measures the amount of that frequency in $f(t)$. This value is complex, but the machinery of magnitude and phase still works. The Fourier transform is linear, and the expressions for this don't change (which is why we overload the notation):

- $\mathcal{F}(cf) = c\mathcal{F}(f)$ for any constant c.
- $\mathcal{F}(f+g) = \mathcal{F}(f) + \mathcal{F}(g)$.

Few people need to be able to compute a closed form expression for the Fourier transform of a function; instead, one looks up interesting cases in tables. We defer this to the 2D case, which is more important for us.

Remember this: *A Fourier transform maps a function on the real line to a complex function on the real line. This function measures the amount of each frequency in the original function. The transform exists for all functions we will care about, and there is a corresponding inverse. The process of taking a Fourier transform is linear.*

7.2.1 Convolution and the Convolution Theorem in 1D

The convolution of Section **??** is a discrete analog of a continuous process. In 1D, the *convolution* of two functions $f(t)$ and $g(t)$ is given by

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

Notice that: indices became arguments; and summing has turned into integration. This is the sort of thing you expect when passing between discrete and continuous objects. The "flip" has been preserved. Convolution has deep mathematical roots, which we mostly won't need to investigate. It is of interest, because it interacts very well with Fourier transforms.

To see this, we need one more result. Write $\texttt{shift}_\tau$ for the operation that maps the function $f(t)$ to the function $\texttt{shift}_\tau(f) = f(t-\tau)$. Then

$$\mathcal{F}(\texttt{shift}_\tau f) = \mathcal{F}(f) \exp\left[-i2\pi u\tau\right]$$

because

$$\begin{aligned}\int_{-\infty}^{\infty} f(t-\tau)\exp\left[-i2\pi ut\right]dt &= \int_{-\infty}^{\infty} f(w)\exp\left[-i2\pi u(w+\tau)\right]dw \\ &= \int_{-\infty}^{\infty} f(w)\exp\left[-i2\pi uw\right]dw\exp\left[-i2\pi u\tau\right] \\ &= \left[\mathcal{F}(f)\right]\exp\left[-i2\pi u\tau\right]\end{aligned}$$

This yields a remarkable result, known as the *convolution theorem.* We have

$$\mathcal{F}(f*g) = \left[\mathcal{F}(f)\right]\left[\mathcal{F}(g)\right].$$

From this and material above, it follows that: Multiplication in the Fourier domain is equivalent to convolution in the time domain, *AND* convolution in the Fourier domain is equivalent to multiplication in the time domain. By our relaxed standards of proof, it is easy and quite informative to prove. We have

$$\begin{aligned}\mathcal{F}(f*g) &= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau\right)\exp\left[-i2\pi ut\right]dt \\ &= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} f(t-\tau)g(\tau)\right)\exp\left[-i2\pi ut\right]dtd\tau \\ &= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} f(t-\tau)\exp\left[-i2\pi ut\right]dtg(\tau)\right)d\tau \\ &= \int_{-\infty}^{\infty}\left[\mathcal{F}(f)\right]\exp\left[-i2\pi u\tau\right]g(\tau)d\tau \\ &= \left[\mathcal{F}(f)\right]\int_{-\infty}^{\infty} g(\tau)\exp\left[-i2\pi u\tau\right]d\tau \\ &= \left[\mathcal{F}(f)\right]\left[\mathcal{F}(g)\right].\end{aligned}$$

To make this proof rigorous, you'd need to establish that swapping the order of integration is OK (it is, here!) and that the integrals will exist (they will, for our functions!).

7.2.2 Scale and Fourier Transforms

Scaling the argument of a function has an interesting effect on the Fourier transform. Choose some constant a; we have

$$\begin{aligned}\mathcal{F}(f(at)) &= \int_{-\infty}^{\infty} f(at)\exp\left[-i2\pi ut\right]dt \\ &= \frac{1}{a}\int_{-\infty}^{\infty} f(s)\exp\left[-i2\pi u/as\right]dt \\ &= \frac{1}{a}\mathcal{F}(f)(u/a).\end{aligned}$$

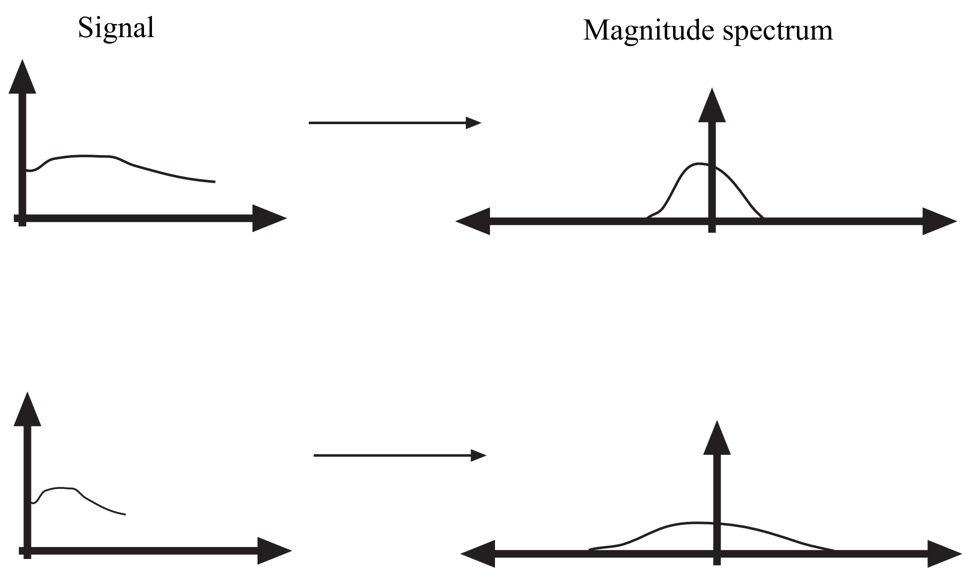

FIGURE 7.1: **Top** *shows* $f(t)$ *and its magnitude spectrum, and* **bottom** $f(2t)$ *and its magnitude spectrum. Notice how narrowing the function broadens the Fourier transform (from top to bottom); or broadening it narrows the Fourier transform (from bottom to top).*

If you scale so that the function is narrowed, then the Fourier transform gets broader; and if you scale so the function gets broader, the Fourier transform is narrowed (Figure 7.1).

7.2.3 Sampling in 1D

Sampling in one dimension takes a function and returns a discrete set of values. We need a model that can be used together with Fourier transform machinery. The most important case involves sampling on a uniform discrete grid, and we assume that the samples are defined at integer points (you can get anything else by adjusting the indexing). We could model sampling as a process that takes a function to a series of values, one for each integer point. The problem with this model is that we have no way to talk about the Fourier transform of this series. Taking a Fourier transform involves computing an integral. One property we would like from the model of sampling is that behaves sensibly when we compute integrals. In particular, we would like

$$\int_W \texttt{sample}_{1D}(f(t))g(t)dt$$

(where W is some interval) to be as similar as possible to

$$\int_W f(t)g(t)dt.$$

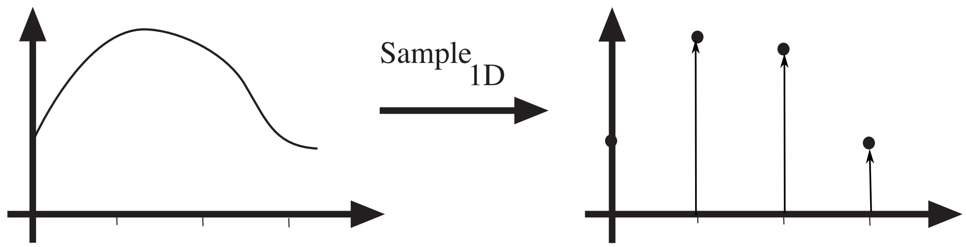

FIGURE 7.2: *Sampling in 1D takes a function $f(t)$ and returns a set of weighted delta functions at each sample point, $\sum_{i=-\infty}^{\infty} f(i)\delta(t-i)$. It is usual to draw delta functions as arrows, with the height of the arrow determined by the weight of the delta function, even though the value of the delta function is hard to talk about.*

This means we can't model the process as producing something that is zero at non-integer points and takes the value $f(t)$ for t an integer point. This model produces something that is indistinguishable from zero – it is different from zero only at a finite set of points.

The right model to use is

$$\texttt{sample}_{1D}(f(t)) = \sum_{i=-\infty}^{\infty} f(i)\delta(t-i).$$

Recall the odd properties of the delta function from above. These properties can be used to establish that

$$\int_W \texttt{sample}_{1D}(f(t))g(t)dt = \sum_{i \in W} f(i)g(i)$$

(where $i \in W$ refers to integer points in the interval). Notice that whether this is an accurate estimate of the integral or not depends a lot on how f and g behave – if they don't vary much between sample points, the estimate will be good, and if there is a lot of variation between sample points, then the estimate will be poor. This should be true of a sampled function.

Remember this: *The proper mathematical model to use for sampling in 1D is:*

$$\texttt{sample}_{1D}(f(t)) = \sum_{i=-\infty}^{\infty} f(i)\delta(t-i).$$

it is a straightforward exercise to modify this model so that samples occur at discrete points that aren't integer points (just fiddle with the indexing).

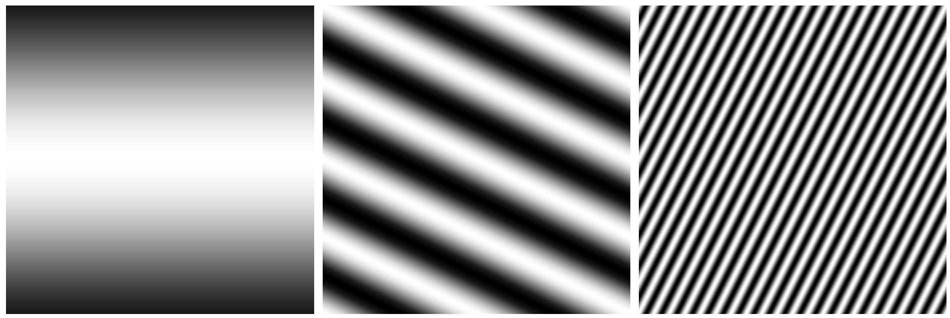

FIGURE 7.3: *The real component of Fourier basis elements shown as intensity images. The brightest point has value one, and the darkest point has value zero. The domain is* $[-1,1] \times [-1,1]$*, with the origin at the center of the image. On the* **left***,* $(u,v) = (0, 0.4)$*; in the* **center***,* $(u,v) = (1,2)$*; and on the* **right** $(u,v) = (10,-5)$*. These are sinusoids of various frequencies and orientations described in the text.*

7.3 FOURIER TRANSFORMS IN 2D

A Fourier transform in 2D is an easy generalization. We now use complex exponentials in two dimensions whose form is

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

These terms are sinusoids on the x, y plane. Their orientation and frequency are given by u, v. These sinusoids are often referred to as *gratings* or *spatial frequency components*; a variety are illustrated in Figure 7.3.

For example, if $u = 0$ and $v = 1$, the real term is a sinusoid that changes along the y-axis and not along the x-axis (a vertical sinusoid); if $u = 1$ and $v = 0$, a horizontal one. The real term 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 the real term is perpendicular to lines where $ux + vy$ is constant, and the frequency of the sinusoid is $\sqrt{u^2+v^2}$. This means that changing the direction of the vector (u,v) changes the orientation of the sinusoid, and changing its length changes the frequency.

A cleaner notation, which we will adopt, is to write (x,y) as $\mathbf{x}$ and (u,v) as $\mathbf{u}$. Then we have

$$\exp\left[i2\pi\mathbf{u}^T\mathbf{x}\right] = \cos(2\pi\mathbf{u}^T\mathbf{x}) + i\sin(2\pi\mathbf{u}^T\mathbf{x}).$$

The Fourier transform of a function of 2D $g(\mathbf{x})$ is given by

$$\mathcal{F}(f) = \mathcal{F}(f(\mathbf{x}))(\mathbf{u}) = \int_{-\infty}^{\infty} f(\mathbf{x})e^{-i2\pi\mathbf{u}^T\mathbf{x}}d\mathbf{x}.$$

Assume that appropriate technical conditions are true to make this integral exist. It is sufficient for all moments of f to be finite; a variety of other possible conditions are available. The process takes a complex valued function of $\mathbf{x}$ and returns a complex valued function of $\mathbf{u}$ (images are complex valued functions with zero imaginary component).

The integral should be seen as a dot product. If we fix $\mathbf{u}$, the value of the integral is the dot product between a sinusoid in $\mathbf{x}$ and the original function. This is a useful analogy because dot products measure the amount of one vector in the direction of another.

You should see the Fourier transform in 2D as a very mildly more complicated version of a Fourier transform in 1D. The difference you will most commonly encounter is that functions are now sometimes referred to as being in the *spatial domain* (rather than the time domain), but their Fourier transforms are still in the Fourier domain. The value of the Fourier transform at u, v measures the amount of that frequency in $f(x, y)$. This value is complex, but the machinery of magnitude and phase still works. The Fourier transform in 2D is linear, and the expressions for this don't change (which is why we overload the notation):

- $\mathcal{F}(cf) = c\mathcal{F}(f)$ for any constant c.
- $\mathcal{F}(f + g) = \mathcal{F}(f) + \mathcal{F}(g)$.

> **Remember this:** *A Fourier transform in 2D maps a function on the plane to a complex function on the plane. This function measures the amount of each frequency in the original function. The transform exists for all functions we will care about, and there is a corresponding inverse. The process of taking a Fourier transform is linear.*

7.3.1 Convolution and the Convolution Theorem in 2D

Because the Fourier transform in 2D is so similar to that in 1D, everything we proved in 1D works in 2D as well. The analog of the scaling result of Section 7.2.2 is Choose some constant a; we have

$$\mathcal{F}(f(ax, by)) = \frac{1}{ab}\mathcal{F}(f)(u/a, v/b).$$

As for 1D, if you scale so that the function is narrowed, then the Fourier transform gets broader; and if you scale so the function gets broader, the Fourier transform is narrowed. In 2D, the *convolution* of two functions $f(\mathbf{x})$ and $g(\mathbf{x})$ is given by

$$(f * g)(\mathbf{x}) = \int_{-\infty}^{\infty} f(\mathbf{x} - \mathbf{z})g(\mathbf{z})d\mathbf{z} = \int_{-\infty}^{\infty} f(\mathbf{z})g(\mathbf{x} - \mathbf{z})d\mathbf{z}$$

Shifting is different (in a fairly obvious way); now $\mathtt{shift}_{\mathbf{z}}$ is the operation that maps the function $f(\mathbf{x})$ to the function $\mathtt{shift}_{\mathbf{z}}(f) = f(\mathbf{x} - \mathbf{z})$. Then

$$\mathcal{F}(\mathtt{shift}_{\mathbf{z}} f) = \mathcal{F}(f) \exp\left[-i2\pi\mathbf{u}^T\mathbf{z}\right]$$

yielding

$$\mathcal{F}(f * g) = [\mathcal{F}(f)]\,[\mathcal{F}(g)]\,.$$

In 2D (and in any D): Multiplication in the Fourier domain is equivalent to convolution in the spatial domain, *AND* convolution in the Fourier domain is equivalent to multiplication in the spatial domain.

> **Remember this:** *The convolution theorem – Multiplication in the Fourier domain is equivalent to convolution in the signal domain,* AND *convolution in the Fourier domain is equivalent to multiplication in the signal domain.*

7.3.2 Sampling in 2D

Sampling in 2D is very like sampling in 1D. We need a delta function in two dimensions (which behaves a lot like one in 1D). We have $\delta(x-s, y-t)$ is zero at all points where $x \neq s$ or $y \neq t$. We do not discuss the value at the crucial point; instead, the function has the property

$$\int_{-\infty}^{\infty} f(x,y)\delta(x-s,y-t)dxdy = f(s,t).$$

We now use the model

$$\texttt{sample}_{2D}(f(t)) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)\delta(x-i,y-j).$$

The properties of the delta function can be used to establish that

$$\int_W \texttt{sample}_{2D}(f(x,y))g(x,y)dt = \sum_{i\in W} f(i)g(i)$$

(where $i \in W$ refers to integer points in the interval). Notice that whether this is an accurate estimate of the integral or not depends a lot on how f and g behave – if they don't vary much between sample points, the estimate will be good, and if there is a lot of variation between sample points, then the estimate will be poor. This should be true of a sampled function.

> **Remember this:** *The proper mathematical model to use for sampling in 2D is:*
>
> $$\texttt{sample}_{2D}(f(x,y)) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)\delta(x-i,y-j).$$
>
> *it is a straightforward exercise to modify this model so that samples occur at discrete points that aren't integer points (just fiddle with the indexing).*

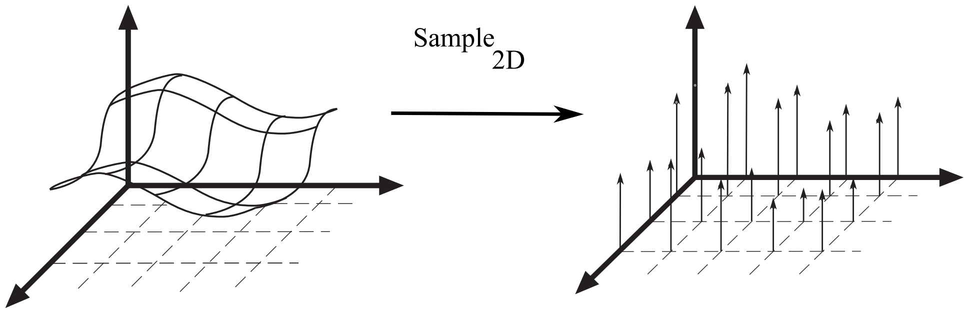

FIGURE 7.4: *Sampling in 2D takes a function and returns an array; again, we allow the array to be infinite dimensional and to have negative as well as positive indices.*

7.4 FOURIER TRANSFORM FACTS

7.4.1 Fourier Transform Pairs

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 7.1 for reference.

7.4.2 Phase and Magnitude for Images

The value of the Fourier transform of a function at a particular $\mathbf{u}$ 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. A local change in the function (e.g., zeroing out a block of points) is going to lead to a change *at every point* in the Fourier transform. This means that the Fourier transform is quite difficult to use as a representation (e.g., it might be very difficult to tell whether a pattern was present in an image just by looking at the Fourier transform). One important exception occurs when one wants to change particular frequency components. This can occur when a set of images is pasted together, for example. Figure 7.5 shows a composite image of the elephant's ear of Africa, obtained by the European space agency. In this case, there is an obvious periodic overlap. As the figure shows, this can be corrected by: taking a Fourier transform, zeroing a particular set of magnitudes, then taking the inverse of the result.

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 7.6 for an example). Fourier transform two images, swap their magnitude spectra, then inverse Fourier transform the results. As the figure shows, the resulting images mostly look like the image from which the phase was obtained.

TABLE 7.1: *A variety of functions of two dimensions and their Fourier transforms. This table can be used in two directions (with appropriate substitutions for u, v and x, y) because the Fourier transform of the Fourier transform of a function is the function. Observant readers might suspect that the results on infinite sums of δ functions contradict the linearity of Fourier transforms. By careful inspection of limits, it is possible to show that they do not (see, for example, ?). Observant readers also might have noted that an expression for $\mathcal{F}(\frac{\partial f}{\partial y})$ can be obtained by combining two lines of this table.*

Function	Fourier transform
$f(\mathbf{x})$	$\int_{-\infty}^{\infty} f(\mathbf{x}) \exp\left[-i2\pi(\mathbf{u}^T\mathbf{x})\right] d\mathbf{x}$
$\int_{-\infty}^{\infty} \mathcal{F}(f)(\mathbf{u}) \exp\left[i2\pi(\mathbf{u}^T\mathbf{x})\right] d\mathbf{u}$	$\mathcal{F}(f)(\mathbf{u})$
$\delta(\mathbf{x})$	1
$\frac{\partial f}{\partial x}(\mathbf{x})$	$u\mathcal{F}(f)(\mathbf{u})$
$0.5\delta(x+a,y)+0.5\delta(x-a,y)$	$\cos 2\pi a u$
$\exp\left[-\pi(\mathbf{x}^T\mathbf{x})\right]$	$\exp\left[-\pi(\mathbf{u}^T\mathbf{u})\right]$
$box_1(x,y)$	$\frac{\sin u}{u}\frac{\sin v}{v}$ (often called a *sinc function*)
$f(ax,by)$	$\frac{\mathcal{F}(f)(u/a,v/b)}{ab}$
$\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)$	$\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(u-i,v-j)$
$(f * g)$	$\mathcal{F}(f)\mathcal{F}(g)$
$f(\mathbf{x}-\mathbf{z})$	$\exp\left[i2\pi\mathbf{u}^T\mathbf{z}\right]\mathcal{F}(f)$
for $\mathcal{R}$ a rotation matrix $f(\mathcal{R}\mathbf{x})$	$\mathcal{F}(f)(\mathcal{R}\mathbf{u})$
for $\mathcal{A}$ an invertible matrix and $\mathbf{z}$ a constant vector $f(\mathcal{A}\mathbf{x}+\mathbf{w})$	$\frac{1}{\det(\mathcal{A})}\exp\left[-i2\pi\mathbf{u}^T\mathbf{w}\right]\mathcal{F}(f)(\mathcal{A}^{-T}\mathbf{u})$

7.5 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 7.7. The problem has to do with the number of samples relative to the

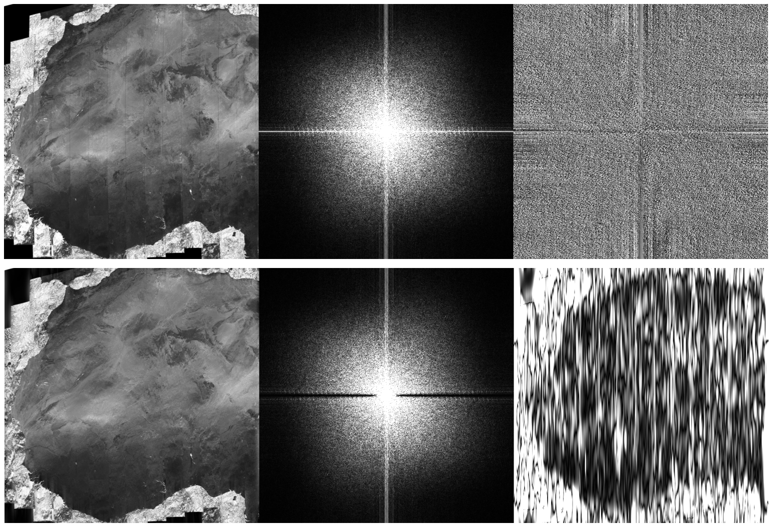

FIGURE 7.5: **Top left** *shows a portion of a mosaic of satellite images of Africa, prepared by the European Space Agency (the original shows the whole continent, at* `https://www.esa.int/ESA_Multimedia/Images/2016/05/African_mosaic`*);* **top center***, the magnitude of the FFT, with zero frequency at the center; and* **top right***, the phase of the FFT. Notice how the vertical stripes caused by overlap problems in the image mosaic result in frequency effects (look at the bright vertical bars on the horizontal axis of the magnitude). Zeroing the horizontal axis removes some frequency components with vertical level curves (equivalently, components that wiggle horizontally), and so suppresses the stripes.* **Bottom left***, a version of the image obtained by masking the magnitude;* **bottom center***, the masked magnitude, where I have zeroed the problem components and likely others, too.* **Bottom right** *shows the absolute value of the difference between original and processed image; notice how what has changed is mostly these components (the residual has strong vertical structure).*

function; we can formalize this rather precisely given a sufficiently powerful model.

7.5.1 The Fourier Transform of a Sampled Signal

We will work in 2D because we care mainly about images, but everything said here will work in other dimensions too. We start with $f(x, y)$ and produce a sampled signal:

$$\texttt{sample}_{2D}(f(x,y)) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i,j)\delta(x-i, y-j).$$

Now we must find the Fourier transform of $\texttt{sample}_{2D}(f(x,y))$.

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

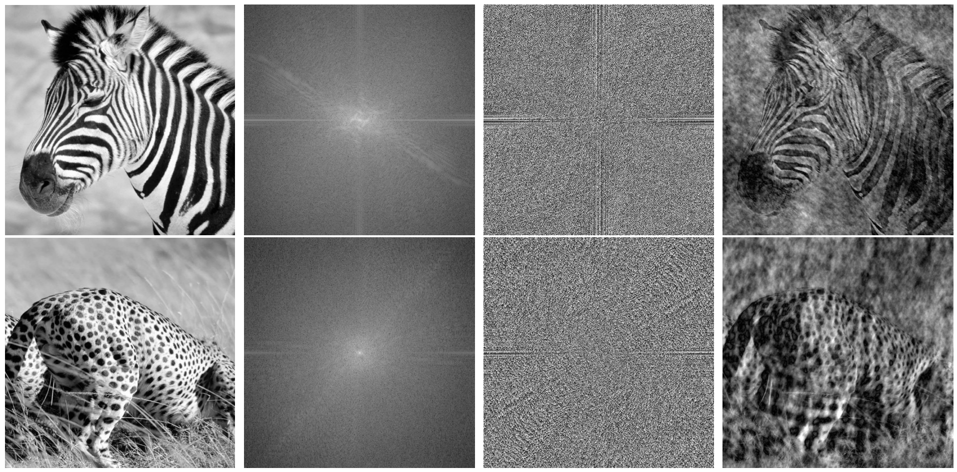

FIGURE 7.6: *The second image in each row shows the log of the magnitude spectrum for the first image in the row; the third image shows the phase spectrum scaled so that $-\pi$ is dark and π is light. The final images are obtained by swapping the magnitude spectra. Although this swap leads to substantial 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 signal with another bed-of-nails function.

But convolving a function with a shifted δ-function just shifts the function. We have

$$f * \delta(x-i, y-j) = \int_{-\infty}^{\infty} f(x-s, y-t)\delta(s-i, t-j) ds dt = f(x-i, y-j)$$

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,

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

where we have written the Fourier transform of $f(x, y)$ as $F(u, v)$. Now if $F(u, v)$ is narrow enough in u, v space, then we can 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 7.8). This will recover the original signal.

But this procedure could fail. If $F(u, v)$ is broad enough, the translated copies will overlap one another and add. We cannot separate them, meaning that the signal we reconstruct from a cut out portion will not be the original (Figure 7.9). This results in a characteristic effect, usually called *aliasing*, where high spatial frequencies appear to be low spatial frequencies (see Figure 7.10 and exercises).

A somewhat roundabout argument establishes that taking more samples may allow us to reconstruct the function. It is convenient to have samples on an integer grid, so rather than sampling more often, we will scale the arguments of the

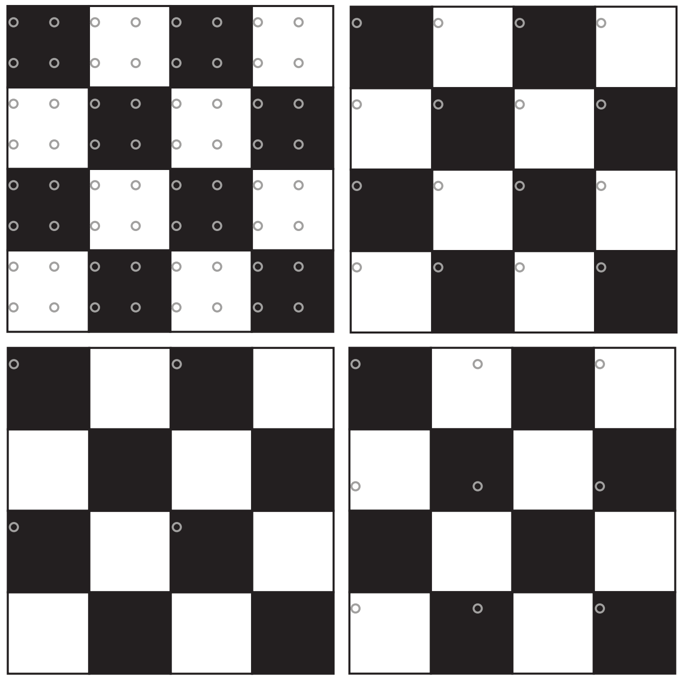

FIGURE 7.7: *The two checkerboards on the* **top** *illustrate a sampling procedure that appears to be successful (whether it is or not depends on some details that we will deal with later). The gray circles represent the samples; if there are sufficient samples, then the samples represent the detail in the underlying function. The sampling procedures shown on the* **bottom** *are unequivocally unsuccessful; the samples suggest that there are fewer checks than there are. This illustrates two important phenomena: first, successful sampling schemes sample data often enough; and second, unsuccessful sampling schemes cause high-frequency information to appear as lower-frequency information.*

function. You should check that sampling $f(2x, 2y)$ on the integer grid will give a representation of f that is equivalent to sampling $f(x, y)$ on a half-integer grid. But the Fourier transform of $f(2x, 2y)$ is narrower than that of $f(x, y)$. If the Fourier transform of f is non-zero on a bounded region, eventually we can shrink it so there are no overlaps — equivalently, we can sample f frequently enough to be able to reconstruct it precisely.

All this 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.

The convolution theorem explains why the Gaussian filter is quite good at smoothing images and the box filter is not. Convolving with a Gaussian is equivalent to multiplying the Fourier transform with a Gaussian (the Fourier transform of a

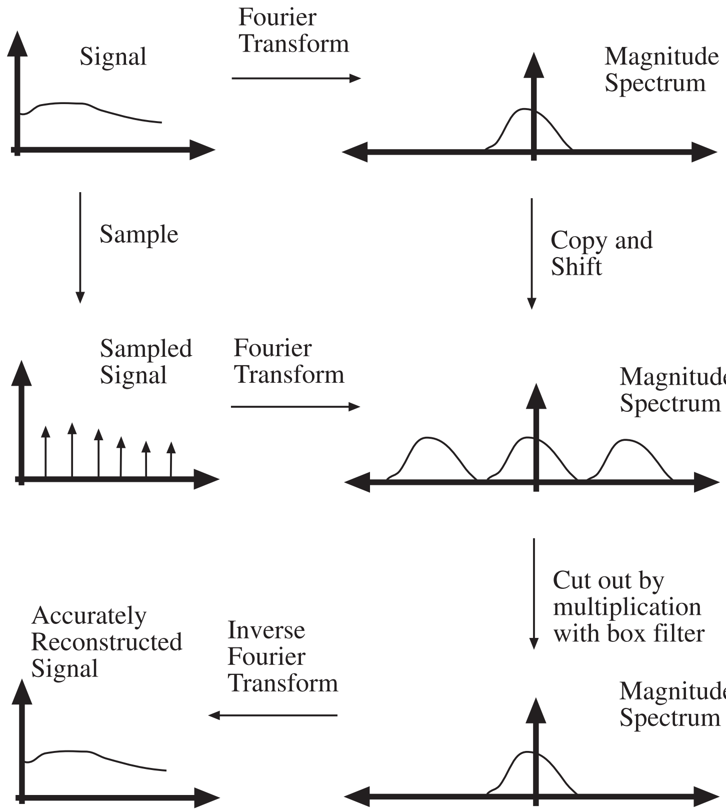

FIGURE 7.8: *The Fourier transform of the sampled signal consists of a sum of copies of the Fourier transform of the original signal, shifted with respect to each other by the sampling frequency. Two possibilities occur. If the shifted copies do not intersect with each other (as in this case), the original 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 7.9), the intersection region is added, and so we cannot obtain a separate copy of the Fourier transform, and the signal has aliased.*

Gaussian is a Gaussian, Table 7.1). Doing so suppresses high spatial frequencies. But the Fourier transform of a box is a sinc function (Table 7.1). Multiplying with the Fourier transform of the image with a sinc function will enhance some high spatial frequencies and suppress others. This is why you see the odd ringing effects in Figure 3.6.

7.5.2 Reconstruction

Assume we have some function f that we must sample and reconstruct. Assume we sample at the rate required by Nyquist's theorem. We must now reconstruct the original signal from the samples. The argument of Section 7.5.1 and Figure 7.8

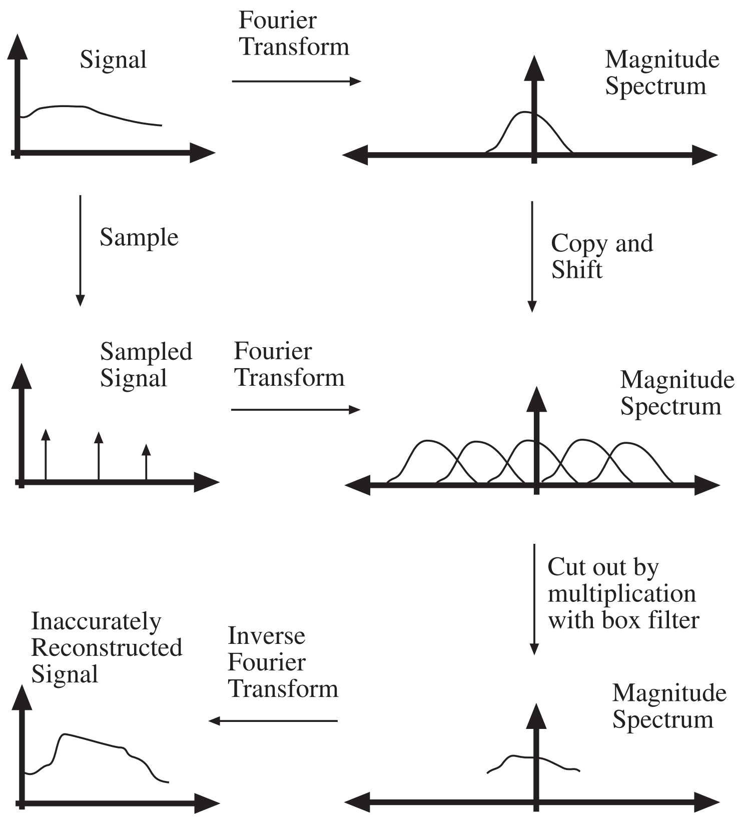

FIGURE 7.9: *The Fourier transform of the sampled signal consists of a sum of copies of the Fourier transform of the original signal, shifted with respect to each other by the sampling frequency. Two possibilities occur. If the shifted copies do not intersect with each other (as in Figure 7.8), the original 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.*

suggests a way to do this. Take the Fourier transform of the sampled signal. Multiply that by a box function to cut out one spectrum, then apply an inverse Fourier transform.

The convolution theorem, together with the Fourier transform pairs of Table 7.1, suggests this might be difficult to do in practice. Multiplying the Fourier transform of the sampled signal by a box function is equivalent to convolving the sampled signal with the Fourier transform of a box function. But the Fourier transform of a box function is $\frac{\sin u}{u}\frac{\sin v}{v}$ (sometimes called a *sinc function*). This has infinite support – it is non-zero for arbitrarily large values of u and v. Convolving with such a function is going to present challenges because you will need

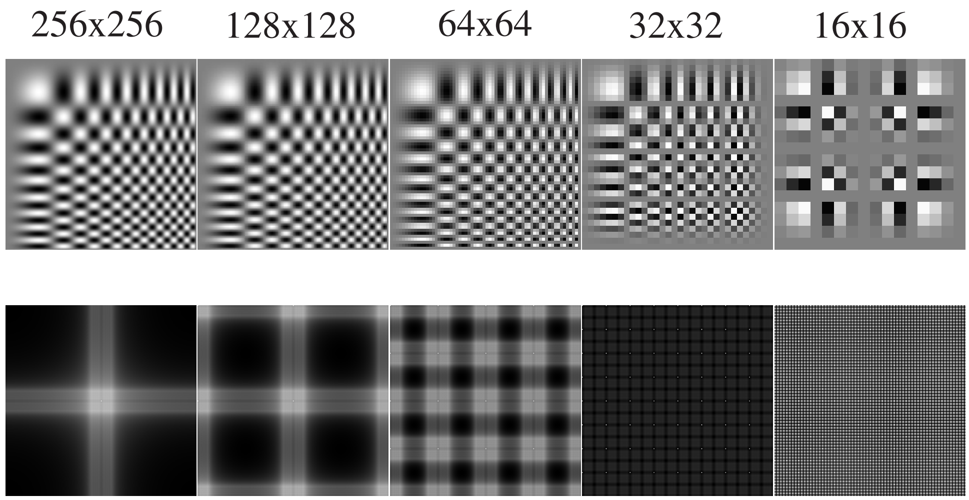

FIGURE 7.10: *The* **top row** *shows sampled versions of an image of a grid obtained by multiplying two sinusoids with linearly increasing frequency—one in x and one in y. The other images in the series are obtained by resampling by factors of two without smoothing (i.e., the next is a 128x128, then 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.*

to know the value of f at infinite numbers of samples.

Generally, we will reconstructing a function from samples by convolving the sampled function with a *reconstruction kernel* or *reconstruction filter*. But this filter must have finite support. Recall the property of Fourier transforms that "narrow" functions have "broad" Fourier transforms. This suggests correctly that the Fourier transform of a filter with finite support will be not have finite support. Equivalently, avoiding aliasing entirely won't be possible in practice. More important is to control aliasing. In the exercises, you will show that bilinear interpolation between samples is equivalent to convolving with a "tent" filter, and that this doesn't give ideal reconstructions.

7.5.3 Smoothing and Resampling

Nyquist's theorem means it is dangerous to shrink an image by simply taking every kth pixel (as Figure 7.10 confirms). Instead, we need to filter the image so that spatial frequencies above the new sampling frequency are removed. The most in-

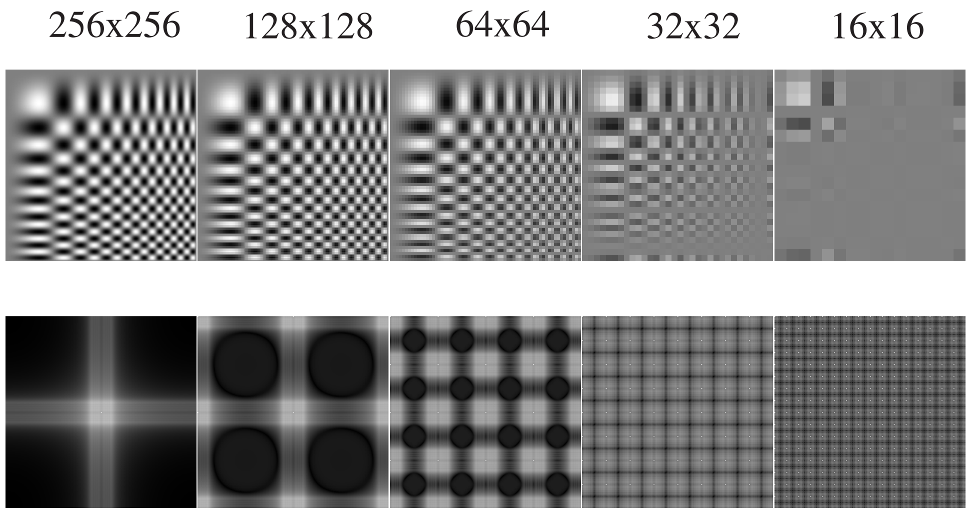

FIGURE 7.11: **Top:** *Resampled versions of the image of Figure 7.10, again by factors of two, but this time each image is smoothed with a Gaussian of σ one pixel before resampling. This filter is a low-pass filter, and so suppresses high spatial frequency components, reducing aliasing.* **Bottom:** *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.*

teresting case occurs when we want to halve the width and height of the image. We assume that the sampled image has no aliasing (because if it did, there would be nothing we could do about it anyway; once an image has been sampled, any aliasing that is going to occur has happened, and there's not much we can do about it without an image model). 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 we resample this signal, the copies now have centers on the half-integer points in u, v space. This means that, to avoid aliasing, we need to apply a filter that strongly reduces the content of the original Fourier transform outside the range $|u| < 1/2$, $|v| < 1/2$. Of course, if we reduce the content of the signal *inside* this range, we might lose information, too. Now the Fourier transform of a Gaussian is a Gaussian, and Gaussians die away fairly quickly. Thus, if we were to convolve the image with a Gaussian—or multiply its Fourier transform by a Gaussian, which is the same thing—we could achieve what we want.

The choice of Gaussian depends on the application. If σ is large, there is less aliasing (because the value of the kernel outside our range is very small), but information is lost because the kernel is not flat within our range; similarly, if σ is small, less information is lost within the range, but aliasing can be more substantial. Figures 7.11 and 7.12 illustrate the effects of different choices of σ.

A particularly attractive feature of a Gaussian filter is that if you convolve a Gaussian with a Gaussian, you get another Gaussian. One neat way to see this

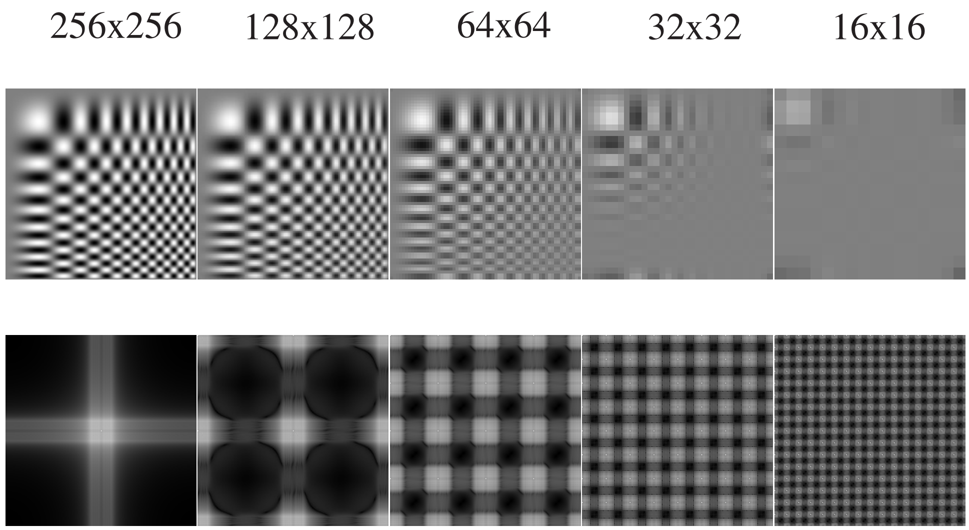

FIGURE 7.12: **Top:** *Resampled versions of the image of Figure 7.10, again by factors of two, but this time each image is smoothed with a Gaussian of σ two pixels before resampling. This filter suppresses high spatial frequency components more aggressively than that of Figure 7.11.* **Bottom:** *The 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.*

is to use the convolution theorem, which also yields how the standard deviations scale (exercises).

Apply a low-pass filter to the original image
  (a Gaussian with a σ of between one
  and two pixels is usually an acceptable choice).
Create a new image whose dimensions on edge are half
  those of the old image
Set the value of the i, jth pixel of the new image to the value
  of the $2i$, $2j$th pixel of the filtered image

Algorithm 7.1: *Subsampling an Image by a Factor of Two.*

We have been using a Gaussian as a low-pass filter because its response at high spatial frequencies is low and its response at low spatial frequencies is high. In fact, the Gaussian is not a particularly good low-pass filter. What one wants is 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. 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? The basic steps for resampling an image are given in Algorithm 7.1.