

Extended version of the notes on sampling theory for the Part II Information Theory & Coding course

Neil Dodgson*
based on Computer Lab. Tech. Report No. 261

February 18, 2003

These notes are based on a Technical Report which discusses the resampling of images. A lot of the text therefore refers to images but it can easily be generalised to all forms of data which need to be sampled. In this text, the term “image” will usually mean “digital image” and the the concept of a real image is captured by the phrase “intensity surface”, i.e. a function from the 2D real plane to intensity.

1 Sampling

Sampling is the process of converting a continuous function into a discrete representation. Conventional sampling theory deals with regularly spaced point samples, with each sample being produced by taking the value of the continuous function at a single point.

1.1 Converting the continuous into the discrete

Sampling takes a continuous signal, one defined over all space, and produces a discrete signal, one defined over only a discrete set of points. In practice the two signals are only defined within a region of interest. For example, when sampling images, the continuous signal is an intensity surface; that is: a three-dimensional function with two spatial dimensions and one intensity dimension where each point in the spatial plane has a single intensity. When dealing with colour, the number of dimensions increases, typically to three colour dimensions (but still only two spatial dimensions). When dealing with one spatial dimension (e.g. sound) or three spatial dimensions (e.g. a CAT or MRI scan of the human body) the situation also changes but the underlying principle is the same: at every spatial point there is either a scalar or vector value, which is the value of the function at that point, be it intensity, colour, pressure, density or whatever.

The discrete signal has the same number of spatial and function value dimensions as the continuous signal but is discrete in the spatial domain. Furthermore, a digital computer cannot represent a continuous quantity and so the sampled signal will be discrete in

*©2003 Neil A. Dodgson

value as well. The process of converting a continuous value to a discrete one is called quantisation.

1.2 Quantisation

Ideally, infinite precision in recorded value is desirable because no errors would then arise from slightly imprecise values. In practice, the sampling device has only finite precision and the values are recorded to only finite precision.

The important point is that sufficient quantisation levels must be used that any errors caused by imprecision are unimportant. For example, with sound and images, the number of quantisation levels must be sufficient to fool the human observer into believing that the signal is continuous. We considered this in the Part IB Computer Graphics course, and the conclusions we drew there are reproduced below as an example.

1.2.1 The minimum necessary number of intensity levels required to represent images

For images, quantisation errors are most noticeable when only a few discrete intensity levels are available, and most image processing texts include an example showing the effect on a displayed image of allowing a larger or smaller number of intensity levels [see, for example, Rosenfeld and Kak, 1982, pp.107–8 figs. 14 and 15; Gonzalez and Wintz, 1977, fig. 2.8]. The extreme case is where there are only two levels: black and white. Such an image takes comparatively little storage (one bit per pixel) but has a limited usefulness. With digital half-toning techniques it can be made to simulate a wide range of grey shades, but, for resampling, it is preferable to have true shades of grey rather than simulated ones.

The general case, then, is one where a significant number of discrete intensity levels are used to represent a continuous range of intensities. But how many intensity levels are sufficient?

The human visual system is limited in how small an intensity change it can detect. Research suggests that, for two large areas of constant intensity, a two percent difference can be just detected [Crow, 1978, p.4]. The minimum difference that can be detected rises to a higher value for very dark or very bright areas [Pratt, 1978, pp.17–18]. It also rises when comparing small areas of constant intensity [*ibid.*, pp.18–19, fig. 2.5]. When dealing with colour images the minimum noticeable differences for pure colour information (that is with no intensity component) are much larger than those for intensity information, hence broadcast television has an intensity channel with high spatial resolution and two channels carrying the colour information at low spatial resolution [NMFPT, 1992]. Here, we shall consider only intensity. The number of intensity levels required to produce a faithful representation of an original intensity surface depends on the image itself [Gonzalez and Wintz, pp.27-28] and on the display device.

If we wish to display any image on a ‘reasonable’ display device then how many levels are required? Many researchers have answered this question and their results are given below. We are working with binary digital computers and so, sensibly, most of the an-

swers are powers of two. Crow [1978, p.4] says that between six and eight bits (64 to 256 intensity levels) are required, depending on the quality of the display. He also calculates that the typical display would need about 162 intensity levels. This calculation is reproduced and explained later. Gonzalez and Wintz [1977, pp.19 and 26] suggest that 64 levels (six bits) may be sufficient in some cases but say that “to obtain displays that will appear reasonably smooth to the eye for a large class of image types, a range of over 100 intensity levels is generally required.” Rosenfeld and Kak [1982, p.106] also suggest that more than 100 levels may be needed for some applications. Finally, Foley and van Dam [1982, p.594] and Pavlidis [1982, p.39] both agree that 64 levels are generally adequate but that 256 levels may be needed for some images (Pavlidis hints that 256 levels may not be enough for a small number of images). The consensus appears to be that many images require only 64 intensity levels and virtually none need more than 256.

If we wish *all* images that we could possibly display to exhibit no artifacts due to intensity quantisation then we must take into account those images with large areas of slowly changing intensity at the dark end of the intensity range. Crow [1978, p.4] performs a ‘back of the envelope’ calculation to see how many intensity levels are required, given the two percent minimum perceivable intensity difference mentioned earlier. He notes that the most intense spot that a typical cathode ray tube (CRT) can produce is 25 times more intense than the dimmest. If the intensity levels are exponentially distributed (that is, each is two percent brighter than the previous level) then about $\frac{\log 25}{\log 1.02} = 163$ intensity levels are required.

However, for work in image resampling it is important that the intensities are distributed linearly. This means that the average of two intensities is a third intensity that is perceptually halfway between the two. Thus a checkerboard of squares in any two intensities, viewed from sufficient distance that the individual checks are imperceptible, will appear like a plane of constant intensity of the average of the two intensities. This property is important because virtually all resampling work involves taking the weighted average of a number of intensities and this must produce the perceptually correct result. (An exponential distribution could be used if it were transformed to the linear domain before any calculations and transformed back afterwards. This is a tremendous overhead for a resampling operation and should thus be avoided.)

With linearly distributed intensity levels all adjacent levels are the same distance apart, say ΔI , in intensity space. If, as above, the brightest level is 25 times as bright as the darkest and the total number of levels is n then ΔI will be:

$$\Delta I = \frac{25 - 1}{n - 1}$$

ΔI must be such that the second darkest level is two percent brighter than the darkest, therefore, in the above formula, $\Delta I = 0.02$. This means that the number of levels, n , is around $n = 1201$. 1200 is considerably more than the 256 quoted earlier, and is slightly more than 2^{10} (1024). Ten bits should however be sufficient, because the minimum perceivable intensity difference for the darkest levels is known to be higher than two percent. With 1024 levels the difference between the darkest and second darkest levels is 2.35%. This should be small enough provided that the darkest intensity on the screen is dark enough to be in the area of the response curve which rises above the two percent level. With 256 levels (eight bits) the difference is 9.41%, which may be too large.

Observations on an eight bit, linearly distributed display show that it *is* too large: false contouring can be seen in areas of slowly varying dark intensity.

Blinn [1990, p.85] notes the same effect in digital video. In his discussion of digital video format D1 he says: “Digital video sounds like the world of the future, but I understand there’s still a bit of a problem: 8 bits don’t really give enough resolution in the darker areas to prevent contouring or banding.”

Murch and Weiman [1991] have performed experiments which show that ten is the maximum number of bits required to make intensity differences between adjacent levels imperceptible, with the possible exceptions of CRTs with an extremely high maximum intensity.

For any image without large areas of slowly varying dark intensities, ten bits of intensity information is too much, and so, for many images, if the intensity values are stored to a precision of eight bits, few visible errors will result from the quantisation of intensity. If they are stored to a precision of ten bits or more, practically no visible errors will arise.

1.3 The sampling process

Let us now assume that our function values are quantised to sufficient precision that the errors introduced by quantisation are negligible. We therefore go on to consider the process of sampling.

The aim of sampling is to generate sample values so that they “best represent” the original continuous function. This is a good concept but what is meant by ‘best represent’? To some it may mean that the samples obey classical sampling theory. To others it may mean that the resulting samples, when reconstructed on a particular device (e.g. a monitor or a loudspeaker), is indistinguishable from the original as far as is possible.

1.3.1 Classical sampling theory

The roots of sampling theory go back to 1915, with Whittaker’s work on interpolation of a set of equispaced samples. However, most people attribute the sampling theorem to Shannon [1949], who acknowledges a debt to many others in its development. Attribution of the theorem has been jointly given to Whittaker and Shannon [Gonzalez and Wintz, 1977, p.72] Shannon and Nyquist [Turkowsky, 1986], and to Shannon, Kotelnikof and Whittaker [Petersen and Middleton, 1962, p.279]. Shannon’s statement of the sampling theorem is¹:

If a function $f(t)$ contains no frequencies higher than W cps it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2W}$ seconds apart.

[Shannon, 1949, Theorem 1]².

¹The theory laid out by Shannon [1949] and others is for one dimensional sampling only. Petersen and Middleton [1962] extended Shannon’s work to many dimensions.

²cps = cycles per seconds \equiv Hertz

Mathematically, the sampling process is described as the product of $f(t)$ with the comb function:

$$\text{comb}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

where $\delta(t)$ is the Dirac delta-function³. This gives the sampled function:

$$\begin{aligned} \hat{f}(t) &= \left(\sum_{n=-\infty}^{\infty} \delta(t - nT) \right) f(t) \\ &= \sum_{n=-\infty}^{\infty} (\delta(t - nT) f(t - nT)) \end{aligned} \quad (1)$$

This is not the same as:

$$\hat{f}_n = f(nT) \quad (2)$$

Equation 1 is a continuous function which consists of an infinite sum of weighted, shifted Dirac delta functions and is zero everywhere except at $t = nT, n \in \mathcal{Z}$. Equation 2 is a discrete function which is defined on the set $n \in \mathcal{Z}$. The values of the discrete function are the weights on the delta functions that make up the continuous function, that is:

$$\hat{f}(t) = \sum_{n=-\infty}^{\infty} \hat{f}_n \delta(t - nT)$$

In a computer we, of course, store the discrete version; but mathematically and theoretically we deal with the continuous one. The sampling theorem can be justified by considering the function in the frequency domain.

The continuous spatial domain function, $f(t)$, is bandlimited. That is it contains no frequencies higher than ν_b (W in the statement of Shannon's theorem above). Its Fourier transform, $F(\nu)$ is thus zero outside the range $(-\nu_b, \nu_b)$. Sampling involves multiplying $f(t)$ by $\text{comb}(t)$. The equivalent operation in the frequency domain is to convolve $F(\nu)$ by $\text{Comb}(\nu)$, the Fourier transform of $\text{comb}(t)$. $\text{Comb}(\nu)$ is composed of Dirac delta-functions at a spacing of $\frac{1}{T}$ (for a proof of this see Marion [1991, pp.31–32]). Convolution with $F(\nu)$ produces replicas of $F(\nu)$ at a spacing of $\frac{1}{T}$. Figure 1 illustrates this process.

If $T < \frac{1}{2\nu_b}$ then the copies of $F(\nu)$ will not overlap and the original $F(\nu)$ can be retrieved by multiplying $\hat{F}(\nu)$ by a box function:

$$\text{Box}(\nu) = \begin{cases} 1, & |\nu| < \nu_b \\ 0, & \text{otherwise} \end{cases}$$

This removes all the copies of the original except for the copy centred at $\nu = 0$. As this is the original frequency domain function, $F(\nu)$, the spatial domain function will also be perfectly reconstructed.

³The Dirac delta function is zero everywhere except at $t = 0$. The area under the function is unity, that is $\int_{-\infty}^{\infty} \delta(t) dt = 1$. One definition of the Dirac delta function is:

$$\delta(t) = \lim_{a \rightarrow \infty} \sqrt{a} e^{-a\pi x^2}$$

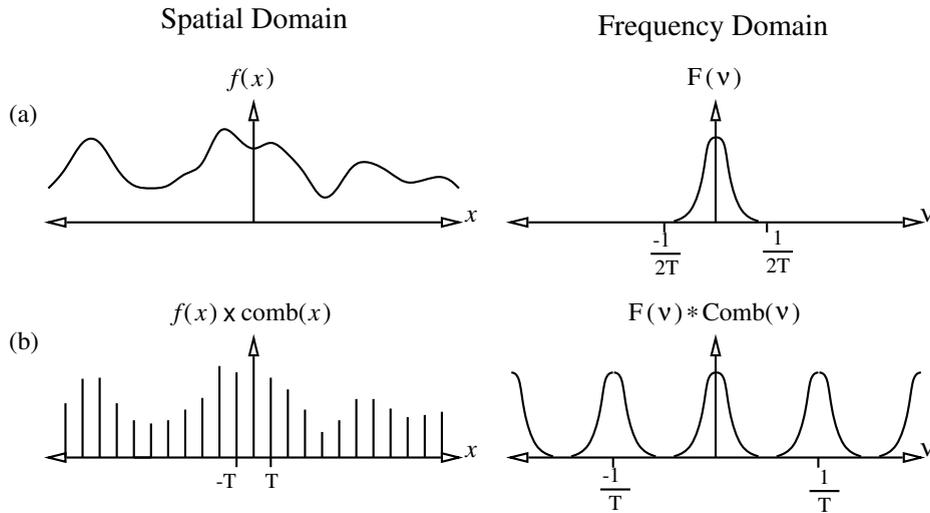


Figure 1: The sampling process: (a) the continuous spatial domain function, $f(x)$, has a Fourier transform, $F(\nu)$, bandlimited to below half the sampling frequency, $\frac{1}{2T}$; (b) when it is sampled ($f(x) \times \text{comb}(x)$) its Fourier transform is convolved with $\text{Comb}(\nu)$ producing replicas of the original Fourier transform at a spacing of $\frac{1}{T}$.

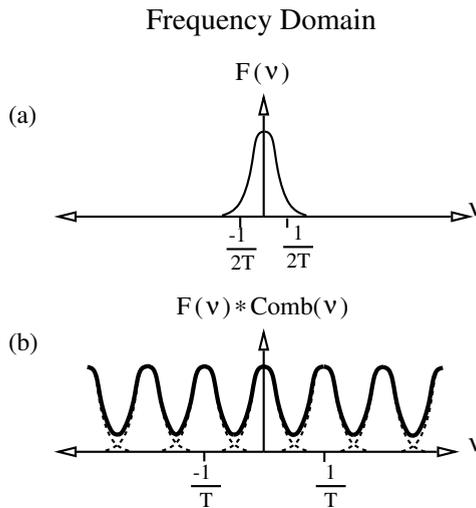


Figure 2: An example of sampling at too low a frequency. Here we see the frequency domain representations of the functions. The original function, (a), is not bandlimited to within half the sampling frequency and so when it is sampled, (b), the copies overlap and add up producing the function shown by the dark line. It is impossible to recover the original function from this aliased version.

Multiplying by a box function in the frequency domain is equivalent to convolving in the spatial domain by the box's inverse Fourier transform, $s(x)$. This can be shown to be $s(x) = 2\nu_b \text{sinc}(2\nu_b x)$, where $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ (a proof of this can be found in appendix A).

If $T \geq \frac{1}{2\nu_b}$ then the copies of $F(\nu)$ will overlap (figure 2). The overlapping parts sum

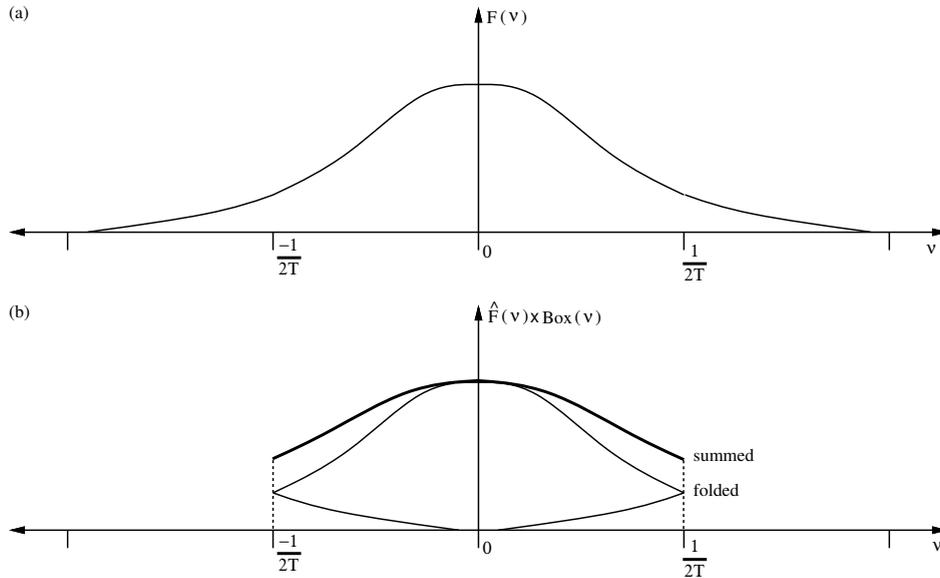


Figure 3: An example of aliasing. (a) is the frequency spectrum of the original image. If the original image is sampled and then reconstructed by multiplying its frequency spectrum by a box function then (b) is the result. The copies of the spectrum overlap and add up, as if $F(\nu)$ in (a) had been folded back about $\frac{1}{2T}$ and summed.

to produce $\hat{F}(\nu)$. There is no way that this process can be reversed to retrieve $F(\nu)$ and so $f(t)$ cannot be perfectly reconstructed. If $\hat{F}(\nu)$ is multiplied by the box function (the perfect reconstructor) then the resulting function is as if $F(\nu)$ had been folded about the frequency $\frac{1}{2T}$ and summed (figure 3). For this reason $\nu = \frac{1}{2T}$ is known as the folding frequency. The effect of this folding is that the high frequencies in $F(\nu)$ alias into low frequencies. This causes artifacts in the reconstructed image which are collectively known as ‘aliasing’. In computer graphics the term aliasing is usually incorrectly used to refer to both aliasing and rastering artifacts [Pavlidis, 1990]. This point is picked up in section 1.5. Figure 4 gives an example of aliasing. It normally manifests as unsightly ripples, especially near sharp changes in intensity.

To avoid aliasing we must ensure that the sampled intensity surface is bandlimited to below the folding frequency. If it is not then it can be prefiltered to remove all information above this frequency. The procedure here is to multiply the intensity surface’s Fourier transform by a box function, a process known as *bandlimiting* the intensity surface. Foley *et al* [1990, fig. 14.29] give an example of this procedure. This prefiltering followed by point sampling is equivalent to an area sampling process, as is explained later.

These procedures are mathematically correct and produce an image free of aliasing. However they have their drawbacks. Firstly, if an intensity surface has to be prefiltered then the image does not represent the original intensity surface but rather the filtered one. For certain applications this may be undesirable. To represent an intensity surface with flat or linearly sloped areas, infinite frequencies *are* required. Bandlimiting prevents such surfaces from being perfectly represented, and so any ‘flat’ part of a band-limited image will be ripply. An example of this effect is that any sharp edge in a band-



Figure 4: An example of a reconstructed intensity surface which contains aliases. These manifest themselves as ripples around the sharp edges in the image. The small image of the letter 'P' is the original. The large is an intensity surface reconstructed from this small original using an approximation to sinc reconstruction.

unlimited intensity surface will have a 9% overshoot either side if it is bandlimited, no matter what the bandlimit is [Lynn and Fuerst, 1989, p.145]; ripples will also propagate out from the discontinuity, reducing in magnitude as they get farther away. Thus a bandlimited intensity surface is, by its very nature, ripply. This can be seen in figure 8(a); there is no aliasing in this figure, the intensity surface is inherently ripply due to being bandlimited. The human visual system abhors ripples, they seriously degrade an image. If however, the ripples in the reconstructed intensity surface are undetectable by the human eye then the surface looks very good.

More importantly, it is practically impossible to achieve perfect prefiltering and so some aliasing will creep in. In an image capture device, some (imperfect) prefiltering will occur; that is: the image capture device does not perform perfect prefiltering. Fraser [1987] asserts that most people depend on this prefiltering to bandlimit the intensity surface enough that little aliasing occurs. In the digital operations of rendering and resampling, perfect prefiltering is simply impossible, ensuring that some aliasing always occurs. This is because perfect prefiltering involves either continuous convolution by an *infinite* sinc function or multiplication of the *continuous* Fourier transform by a bandlimiting function. Both of these operations are impossible in a discrete computer.

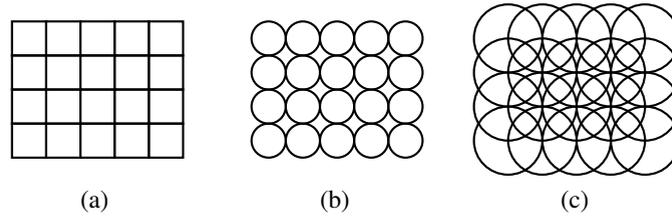


Figure 5: Three common assumptions about pixel shape: (a) abutting squares, (b) abutting circles, and (c) overlapping circles.

Finally, perfect reconstruction is also practically impossible because it requires an infinite image. Thus, whilst it would be theoretically possible to exactly recreate a band-limited intensity surface from its samples, in practice it is impossible. In fact, most display devices reconstruct so badly that correct sampling can produce visually worse results than the optimal incorrect sampling method.

Before discussing this, however, we need to examine the various types of sampling. These fall into two broad categories: area samplers and point samplers.

1.4 Area vs point sampling

Any sampling method will fall into one of these two categories. Area sampling produces a sample value by taking into account the values of the intensity surface over an area. These values are weighted somehow to produce a single sample value. Point sampling takes into account the values of the intensity surface only at a finite set of distinct points⁴. If the set contains more than one point then these values must be combined in some way to produce a single sample value. This is not the usual description of point sampling, because ‘point sampling’ is usually used in its narrow sense: to refer to single point sampling. Fiume [1989, section 3.2.4] studied the various sampling techniques, the following sections draw partly on his work.

1.4.1 Area sampling

Exact-area sampling The assumptions behind exact-area sampling are that each pixel has a certain area, that every part of the intensity surface within that area should contribute equally to the pixel’s sample value, and that any part of the intensity surface outside that area should contribute nothing. Hence, if we make the common assumption that pixels are abutting squares then for a given pixel the sample value produced by the exact area sampler will be the average intensity of the intensity surface within that pixel’s square (figure 5(a)) Alternately, we could assume that pixels are abutting circles [Durand, 1989] or overlapping circles [Crow, 1978] and produce the average value of the intensity surface over the relevant circular area (figure 5(b) and (c)). Obviously other assumptions about pixel shape are possible.

⁴Fiume [1989, p.82] states that the set of point samples should be countable, or more generally, a set of measure zero.

General area sampling The general case for area sampling is that some area is chosen, inside which intensity surface values will contribute to the pixel's sample value. This area is usually centred on the pixel's centre. Some weighting function is applied to this area and the pixel's value is the weighted average over the area. Exact-area sampling is obviously a special case of this.

Equivalence to prefiltering Area sampling is equivalent to prefiltering followed by single point sampling. In area sampling, a weighted average is taken over an area of the intensity surface. With prefiltering a filter is convolved with the intensity surface and a single point sample is taken for each pixel off this filtered intensity surface. To be equivalent the prefilter will be the same function as the area sample weighting function. The two concepts are different ways of thinking about the same process. Heckbert [1989] proposed that sampling could be decomposed into prefiltering and single point sampling. Whilst this is entirely appropriate for all area samplers, not all point samplers can be represented in this way, because some point samplers cannot be represented by filters. Specific examples are adaptive super-samplers and stochastic point samplers, discussed below.

1.4.2 Point sampling

Unlike an area-sampling process, a point-sampler ignores all but a countable set of discrete points to define the intensity value of a pixel. Point sampling comes in two flavours: regular and irregular (or stochastic) point sampling. Wolberg [1990, sections 6.2 and 6.3] summarises the various types of point sampling; here we only outline them.

Single point sampling In single point sampling, the samples are regularly spaced and each pixel's sample value is produced by a single point. In fact this is exactly the method of sampling described in classical sampling theory (section 1.3.1). All other point sampling methods are attempts to approximate area sampling methods or attempts to reduce artifacts in the reconstructed intensity surface (often they attempt to be both).

Super-sampling As in single point sampling, the samples are arranged in a regular fashion, but more than one sample is used per pixel. The pixel's intensity value is produced by taking a weighted average of the sample values taken for that pixel. This can be seen as a direct approximation to area sampling. Fiume [1989, p.96, theorem 6] proves that, given a weighting function, a super-sampling method using this weighting function for its samples converges, as the number of samples increases, toward an area sampling method using the same weighting function.

Such a super-sampling technique can be represented as a prefilter followed by single point sampling; on the other hand, the adaptive super-sampling method cannot.

Adaptive super-sampling Adaptive super-sampling is an attempt to reduce the amount of work required to produce the samples. In order to produce a good quality image, many

super-samples need to be taken in areas of high intensity variance but a few samples would give good results in areas of low intensity variance. Ordinary super-sampling would need to take many samples in all areas to produce a good quality image, because it applies the same sampling process for every pixel. Adaptive super-sampling takes a small number of point samples for a pixel and, from these sample values, ascertains whether more samples need to be taken to produce a good intensity value for that pixel. In this way fewer samples need to be taken in areas of lower intensity variance and so less work needs to be done.

Stochastic sampling Stochastic, or irregular, sampling produces a sample value for each pixel based on one or more randomly placed samples. There are obviously some bounds within which each randomly placed sample can lie. For example it may be constrained to lie somewhere within the pixel's area, or within a third of a pixel width from the pixel's centre. The possible locations may also be constrained by some probability distribution so that, say, a sample point has a greater chance of lying near the pixel centre than near its edge. Stochastic methods cannot be represented as a prefilter followed by single point sampling, because of this random nature of the sample location.

Stochastic sampling is in favour amongst the rendering community because it replaces the regular artifacts which result from regular sampling with irregular artifacts. The human visual system finds these irregular artifacts far less objectionable than the regular ones and so an image of equal quality can be achieved with fewer stochastic samples than with regularly spaced samples (see Cook [1986] but also see Pavlidis' [1990] comments on Cook's work).

Fiume [1989, p.98, theorem 7] proves that the choice of point-samples stochastically distributed according to a given probability distribution will converge to the analogous area-sampling process as the number of points taken increases, a result similar to that for super-sampling.

1.4.3 Summary

Area sampling yields mathematically precise results, if we are attempting to implement some prefilter (section 1.3.1). However, in digital computations it may be impossible, or difficult to perform area sampling because it involves continuous convolution. The point sampling techniques have been shown to be capable of approximating the area sampling methods. Such approximations are necessary in cases where area sampling is impracticable [Fiume, 1989, p.102].

1.5 Anti-aliasing

The purpose of all sampling, other than single-point sampling, is ostensibly to prevent any artifacts from occurring in the image. In fact it is usually used to prevent any artifacts from occurring in the intensity surface reconstructed from the image by the display device. This prevention is generally known in computer graphics as anti-aliasing. This is something of a misnomer as many of the artifacts do not arise from aliases. The



Figure 6: Two very similar images: (a) on the left, was generated by summing the first twenty-nine terms of the Fourier series representation of the appropriate square wave; (b) on the right, was generated by super-sampling a perfect representation of the stripes.

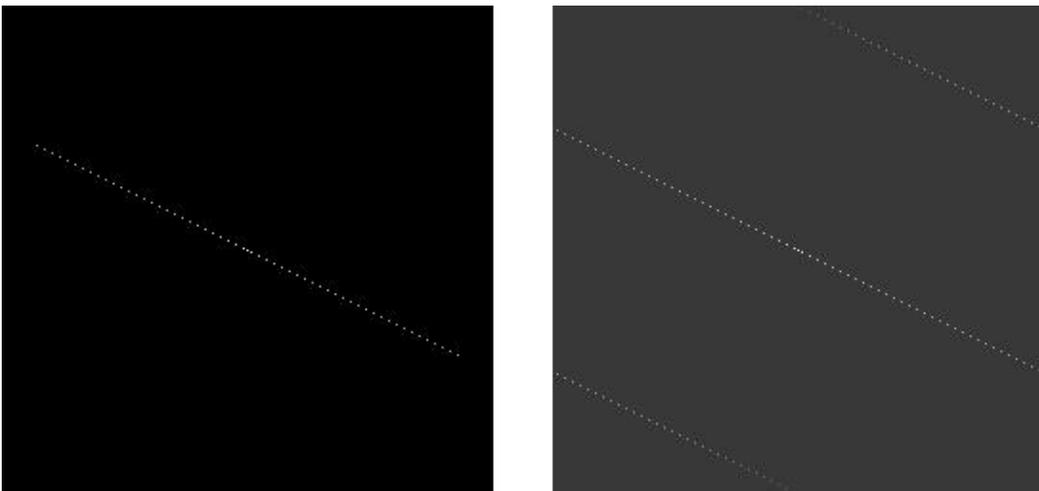


Figure 7: The Fourier transforms of the images in figure 6. (a) on the left and (b) on the right. The Fourier transforms are shown on a logarithmic scale.



Figure 8: The two images of figure 6 reconstructed using as near perfect reconstruction as possible. (a) on the left and (b) on the right. This figure shows part of the reconstructed intensity surface. Aliasing artifacts occur in (b) only

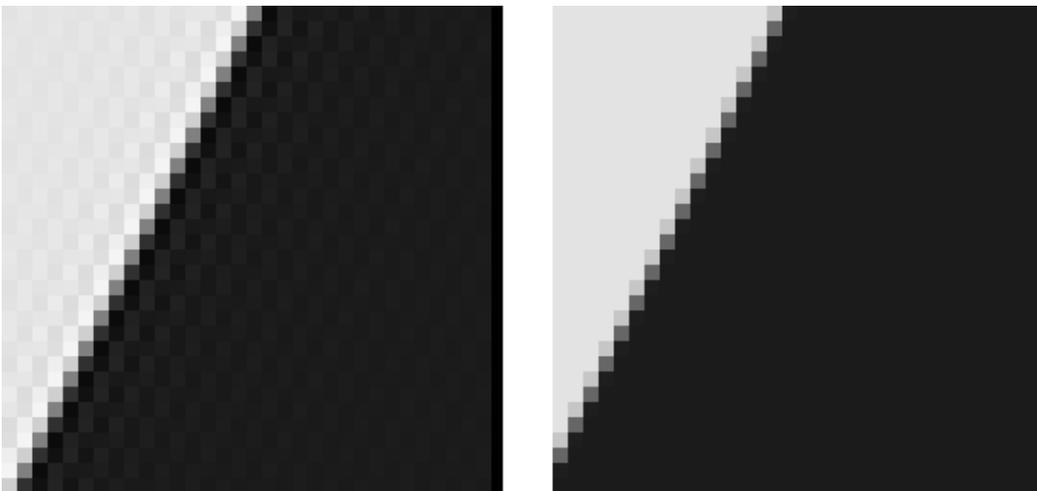


Figure 9: The two images of figure 6 reconstructed using as nearest-neighbour reconstruction. (a) on the left and (b) on the right. This figure shows part of the reconstructed intensity surface. (a) contains only rastering artifacts, while (b) contains a combination of rastering and aliasing artifacts.

perpetuation of the incorrect terminology is possibly due to the fact that the common solution to cure aliasing also alleviates rastering, the other main source of artifacts in images [Pavlidis, 1990, p.234].

Crow [1977] was the first to show that aliasing (sampling a signal at too low a rate) was the cause of some of the artifacts in digital imagery. He also notes [*ibid.*, p. 800] that some artifacts are due to failing to reconstruct the signal properly. This latter problem he termed ‘rastering’. ‘Aliasing’, however, became the common term for both of these effects, with some subsequent confusion, and it is only occasionally that one sees the term ‘rastering’ [Foley *et al.*, 1990, p.641]. Mitchell and Netravali [1988], for example, discuss the two types of artifact as distinct effects but perpetuate the terminology by naming aliasing and rastering, pre-aliasing and post-aliasing respectively.

Sampling has thus been used to alleviate both types of artifact. This has an inbuilt problem that, to correct for reconstruction artifacts (rastering), one needs to know the type of reconstruction that will be performed on the image. Most images are displayed on a CRT, and all CRTs have a similar reconstruction method, so this is not too big a problem. However, when displayed on a different device (for example a film recorder) artifacts may appear which were not visible on the CRT because the reconstruction method is different.

Figures 6 through 9 illustrate the distinction between aliasing artifacts and reconstruction artifacts. Two images are shown of alternate dark and bright stripes. These images were specially designed so that no spurious information due to edge effects would appear in their discrete Fourier transforms (that is: the Fourier transforms shown here are those of these simple patterns copied off to infinity so as to fill the whole plane).

Figure 6(a) was generated by summing the first twenty-nine terms of the Fourier series representation of the appropriate square wave as can be seen by its Fourier transform (figure 7(a)). Figure 6(b) was generated by one of the common anti-aliasing techniques. It was rendered using a 16×16 super-sampling grid on each pixel with the average value of all 256 super-samples being assigned to the pixel. Figure 7(b) shows its Fourier transform. It is similar in form to figure 7(a) but the aliasing can be clearly seen in the wrap around effect of the line of major components, and also in that the other components are not zero, as they are in the unaliased case.

When these are reconstructed using as near-perfect reconstruction as possible we get the intensity surfaces shown in figure 8. The ripples in figure 8(a) are not an aliasing artifact but the correct reconstruction of the function; it is, after all, a sum of sine waves. The ripply effect in figure 8(b) is due to aliasing. The intensity surface that was sampled had constant shaded stripes with infinitely sharp transitions between the dark and light stripes. The perfectly reconstructed intensity surface shown here perfectly reconstructs all of the aliases caused by sampling the original intensity surface.

By contrast, figure 9 shows the intensity surfaces which result from an imperfect reconstructor: the nearest-neighbour interpolant. The artifacts in figure 9(a) are entirely due to *rastering* (the image contains no aliasing). This is the familiar blocky artifact so often attributed, incorrectly, to aliasing. The artifacts in figure 9(b) are due to a combination of aliasing and rastering. Oddly, it is this latter intensity surface which we tend to find intuitively preferable. This is probably due to the areas perceived as having constant

intensity in figure 6(b) retaining this constant intensity in figure 9(b).

How these intensity surfaces are perceived does depend on the scale to which they are reconstructed. The ‘images’ in figure 6 are of course intensity surfaces but are reconstructed to one eighth the scale of those in figure 8 and figure 9. If one stands far enough back from these larger-scale figures then they all look identical.

It is fascinating that the intensity surface with both types of artifact in it appears to be the preferred one. This is possibly due to the facts that (a) the human visual system is good at detecting intensity changes, hence rippling is extremely obvious; and (b) most scenes consist of areas of constant intensity or slowly varying intensity separated by fairly sharp edges, hence representing them as a bandlimited sum of sinusoids will produce what we perceive as an incorrect result: most visual scenes simply do not contain areas of sinusoidally varying intensity.

Thus there is a tension between the sampling theory and the visual effect of the physical reconstruction on the display device. Indeed many researchers, instead of turning to the perfect sampling theory method of sampling, turn instead to methods which take the human observer into account. The main point to take home here is that, in order to make truly successful compression algorithms, we need an understanding of both sampling theory and of how human beings perceive the world. Thus we get to Markus Kuhn’s part of the course, where he discusses practical algorithms which are used for compressing data intended for human perception, i.e. images and sound.

2 Fast Fourier Transform Scaling of Sampled Data

This section is included to give further insight into how the frequency and spatial domains relate to one another through the Fourier transform. It is not intended to be examinable.

The theoretically perfect reconstructed function is generated by convolving the sample data with a sinc function. Whatever the advantages and disadvantages of this reconstruction method, it is sometimes necessary to implement it. Sinc reconstruction is, however, extremely expensive to implement. The sinc function cannot be truncated to produce a local reconstructor without severe artifacts; windowing gives better results but is still not ideal. To implement ideal sinc reconstruction will be an $O(N^2)$ process for every sample point for an $N \times N$ image. Thus resampling an $N \times N$ image to another $N \times N$ image with single point sampling will be an $O(N^4)$ process.

Fraser [1987, 1989a, 1989b] presents a method which uses the fast Fourier transform (FFT) to produce practically identical results to sinc reconstruction in $O(N^2 \log N)$ time. This section discusses his method and presents two variations on it: one to improve its accuracy near image edges; the other to increase its speed and reduce its memory requirements.

2.1 The reconstructed intensity surface

The fast Fourier transform techniques are fast ways of implementing the discrete Fourier transform (DFT). The DFT transforms a finite-length discrete image into a finite-length discrete frequency spectrum. The intensity surface generated by DFT reconstruction is the periodic function described by a Fourier series consisting of the DFT coefficients below the Nyquist frequency and zeros above it [Fraser, 1987]. No doubt this statement requires some explanation.

Given a one-dimensional sequence of N samples, f_j , the DFT produces N frequency values, F_k [Lynn and Fuerst, 1989, p.212]:

$$F_k = \sum_{j=0}^{N-1} f_j e^{-i2\pi jk/N}, \quad k \in \{0, 1, \dots, N-1\}$$

These two discrete series, f_j and F_k , each form the weights of periodic weighted comb functions:

$$\begin{aligned} \iota(x) &= \sum_{j=-\infty}^{\infty} f_j \delta(x - \frac{j}{N}), \quad f_j = f_{j-N} \\ I(\nu) &= \sum_{k=-\infty}^{\infty} F_k \delta(\nu - k), \quad F_k = F_{k-N} \end{aligned}$$

Watson [1986] shows that these two comb functions are Fourier transforms of one another. The relationship is illustrated in figure 10.

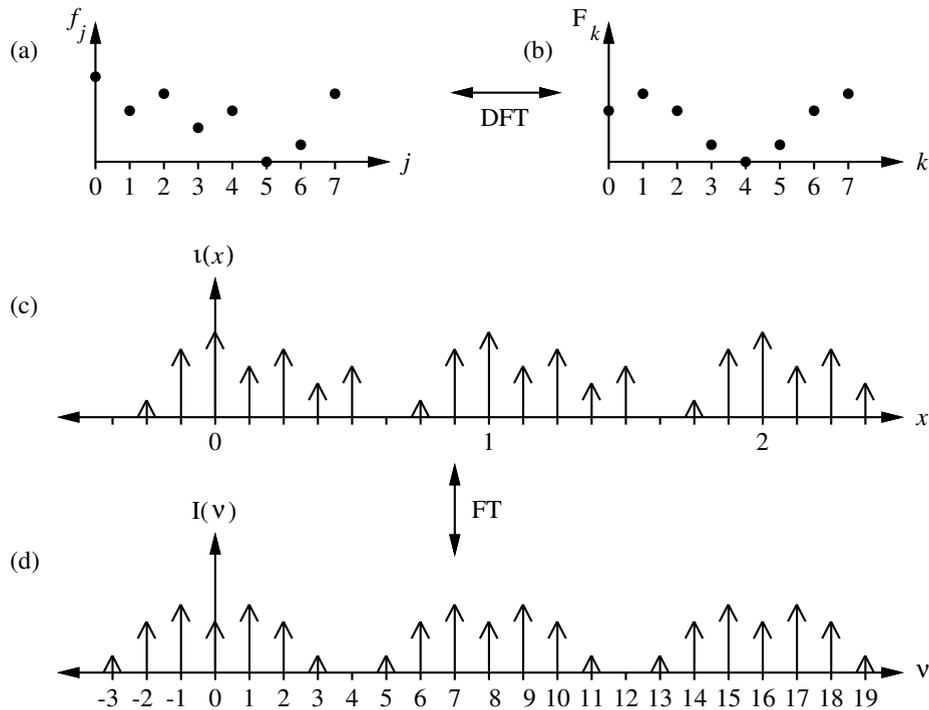


Figure 10: The relationship between the discrete and continuous representations of a sampled function. (a) shows the discrete samples, which can be discrete Fourier transformed to give (b) the DFT of (a). (c) shows $u(x)$, the continuous version of (a), it consists of an infinite periodic set of Dirac delta functions. Its Fourier transform, (d), is also an infinite periodic set of Dirac delta functions and is the continuous representation of (b).

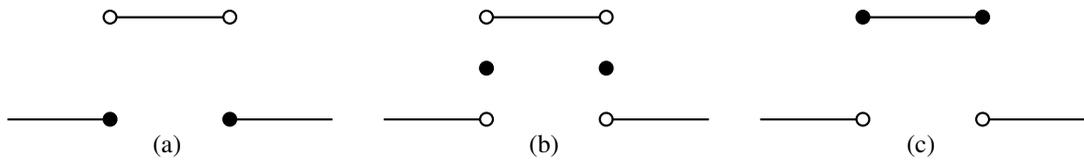


Figure 11: The three possible forms of the box function:

$$(a) b_1(x) = \begin{cases} 1, & |x| < \frac{m}{2} \\ 0, & |x| \geq \frac{m}{2} \end{cases} \quad (b) b_2(x) = \begin{cases} 1, & |x| < \frac{m}{2} \\ \frac{1}{2}, & |x| = \frac{m}{2} \\ 0, & |x| > \frac{m}{2} \end{cases} \quad (c) b_3(x) = \begin{cases} 1, & |x| \leq \frac{m}{2} \\ 0, & |x| > \frac{m}{2} \end{cases}$$

Now, to perfectly reconstruct $f(x)$ from $u(x)$ we convolve it with $N\text{sinc}(Nx)$. This is equivalent to multiplying $I(v)$ in the frequency domain by a box function, suppressing all periods except the central one between $v = -N/2$ and $v = N/2$.

An important question here is what happens at the discontinuities of the box function? Figure 11 gives the three plausible answers to this: the value at the discontinuity can be zero, a half, or one. In continuous work it does not matter particularly which version is used. Here, however, we are dealing with a weighted comb function. If this comb has a tooth at the same place as the box function's discontinuity then it makes a good deal of difference which version of the box function is used. Fraser [1987, p.122] suggests that

the central version be used (where $b_2(\frac{N}{2}) = \frac{1}{2}$). His reasoning is that all components of the Fourier sequence appear once in the box windowed function but the component at the Nyquist frequency occurs twice (once at each end) and so must be given a weight of one half relative to the other components.

A more rigorous reason can be found from taking the Fourier transform of the spatial domain sinc function:

$$B(\nu) = \int_{-\infty}^{\infty} N \text{sinc}(Nx) e^{-i2\pi\nu x} dx, \quad N > 0$$

This can be shown (appendix B) to be:

$$B(\nu) = \begin{cases} 1, & |\nu| < \frac{N}{2} \\ \frac{1}{2}, & |\nu| = \frac{N}{2} \\ 0, & |\nu| > \frac{N}{2} \end{cases} \quad (3)$$

Thus we see that the middle version of the box function is the mathematically correct one to use⁵.

Applying the box function (equation 3) to the function $I(\nu)$ gives a non-periodic weighted comb function, $V(\nu)$:

$$\begin{aligned} V(\nu) &= I(\nu) * B(\nu) \\ &= \sum_{k=-\infty}^{\infty} V_k \delta(\nu - k) \\ V_k &= \begin{cases} F_k, & 0 \leq k < \frac{N}{2} \\ F_{N-k}, & -\frac{N}{2} < k < 0 \\ \frac{1}{2}F_{N/2}, & |k| = \frac{N}{2} \\ 0, & |k| > \frac{N}{2} \end{cases} \end{aligned}$$

Note that there will only be coefficients at $|k| = \frac{N}{2}$ when N is even.

The coefficients of $V(\nu)$ form a Fourier series which describes a continuous function in the spatial domain [Lynn and Fuerst, 1989, p.338]:

$$f(x) = \sum_{k=-\infty}^{\infty} V_k e^{i2\pi kx} \quad (4)$$

This is the intensity surface reconstructed by DFT reconstruction. The whole process is illustrated in figures 12 and 13.

This same sequence of events can be seen in Watson [1986, figs 2 and 4] except that Watson uses the third version of the box function ($b_3(\frac{N}{2}) = 1$) rather than the second.

The reconstructed intensity surface is thus perfectly reconstructed from the samples by theoretically convolving it with a sinc function and, in practice, using the DFT method.

⁵Any of the versions, when Fourier transformed, gives the sinc function, but the inverse transform gives the middle version of the box function.

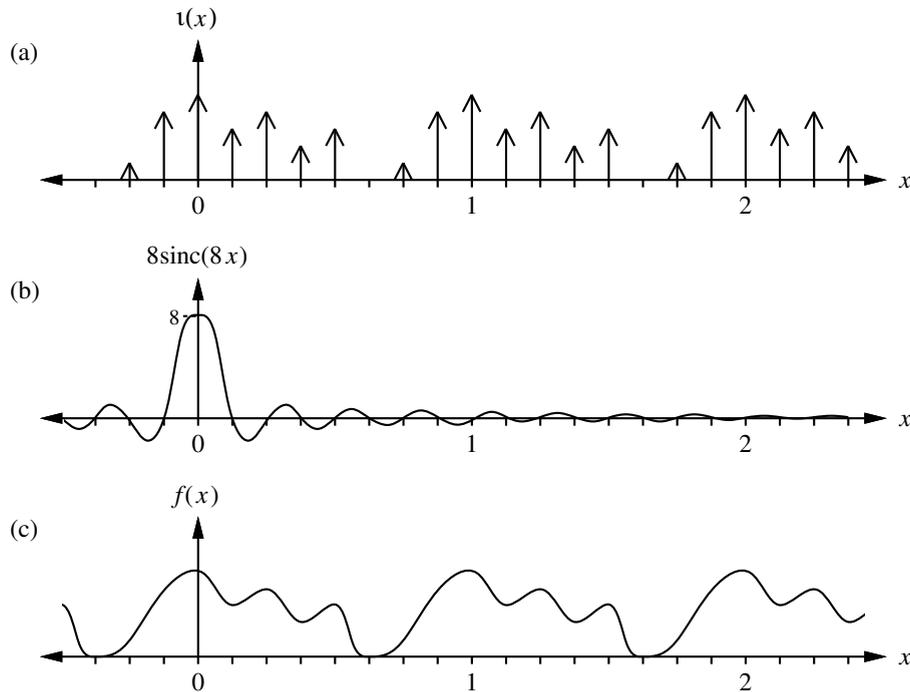


Figure 12: The spatial domain representation of FFT reconstruction. The sampled function, $\iota(x)$ is shown in (a). It is Fourier transformed to give figure 13(a). This Fourier transform is multiplied by a box function (figure 13(b)) to give a finite extent function (figure 13(c)) in the frequency domain. This is inverse Fourier transformed to give the continuous function (c) here. Multiplication by a box in the frequency domain is equivalent to convolving by a sinc function, (b), in the spatial domain.

The significant fact about this surface is that it is periodic. Edge extension has been done by replication and thus the infinite convolution of $\iota(x)$ by $N\text{sinc}(Nx)$ can be performed in finite time: the value at any point on the surface can be found by evaluating the sum of N or $N + 1$ terms in equation 4, although this method of evaluation is as expensive as sinc convolution of the spatial domain samples.

The edge extension implicit in the DFT method is different from that which must be used for sinc interpolation. There we saw that all pixels beyond the image edges must be given the value zero (or somehow be faded off to zero). Here edge extension is by copying, which means that edge effects will be visible in the intensity surface unless we get a fortuitous case where the edges match [Fraser, 1989a, pp.667–668].

2.2 DFT sample rate changing

The DFT can be used to change the sample rate of the image, that is: to scale the image. The seed of this idea can be found in Schafer and Rabiner [1973]. The idea itself was developed by Prasad and Satyanarayana [1986] and modified slightly by Fraser [1989a] to use the correct box function. Their implementation only allows for magnification by an integer power of two. Watson [1986] independently developed the idea, though he does

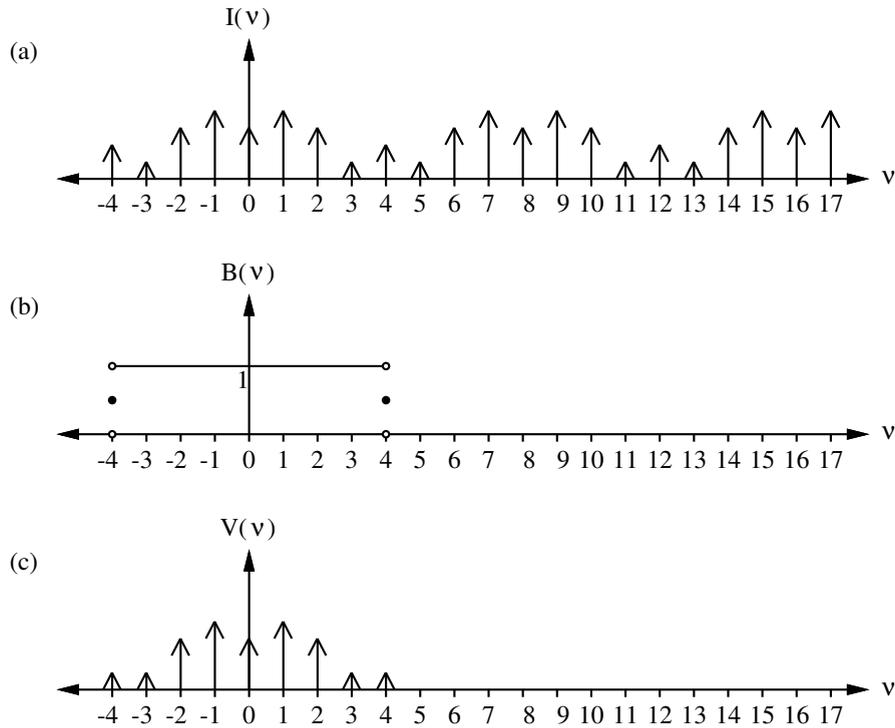


Figure 13: The frequency domain representation of FFT reconstruction. See the caption of figure 12.

not use the correct box function in part of his implementation. His algorithm allows for scaling by a rational factor such that, if the original sequence is N samples and the final L samples, then scaling is by a factor L/N . Watson's algorithm can thus be used to scale from any whole number of samples to any other whole number of samples whilst Fraser's can only scale from N to $2^n N$, $n \in \mathcal{N}$.

2.2.1 How it works

The following explanation is based on Watson's [1986] work, with the modification that the correct box function is used in the first multiplication (Watson already uses it in the second stage so it is surprising that he is inconsistent). We first present the algorithm in the continuous domain, then explain how it can be implemented digitally. Again the one-dimensional case is used for clarity.

The continuous version of the DFT sample rate changing process is illustrated in figure 15 (minification) and figure 16 (magnification). Our image sample values represent the weights on one period of a periodic, weighted comb function (figure 15(a)). This is Fourier transformed to give another periodic weighted comb function (figure 15(b)). The Fourier transform is then correctly box filtered (figure 15(d)) which gives a continuous intensity surface in the spatial domain (figure 15(c)). To resample this intensity surface at a different frequency (equivalent to scaling it and then sampling it at the same frequency) we first perfectly prefilter it, using another correct box function in the fre-

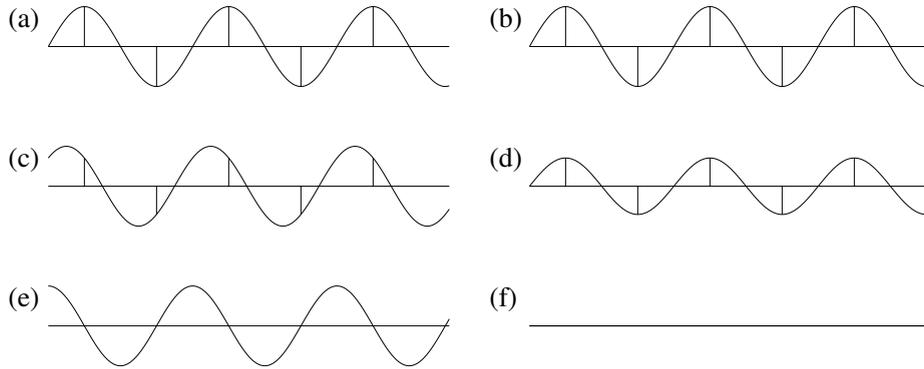


Figure 14: Critical sampling. The sine waves in (a), (c), and (e) are sampled at exactly twice their frequencies. The perfectly reconstructed versions are shown in (b), (d), and (f) respectively. The function is only correctly reconstructed when the samples are in exactly the right place.

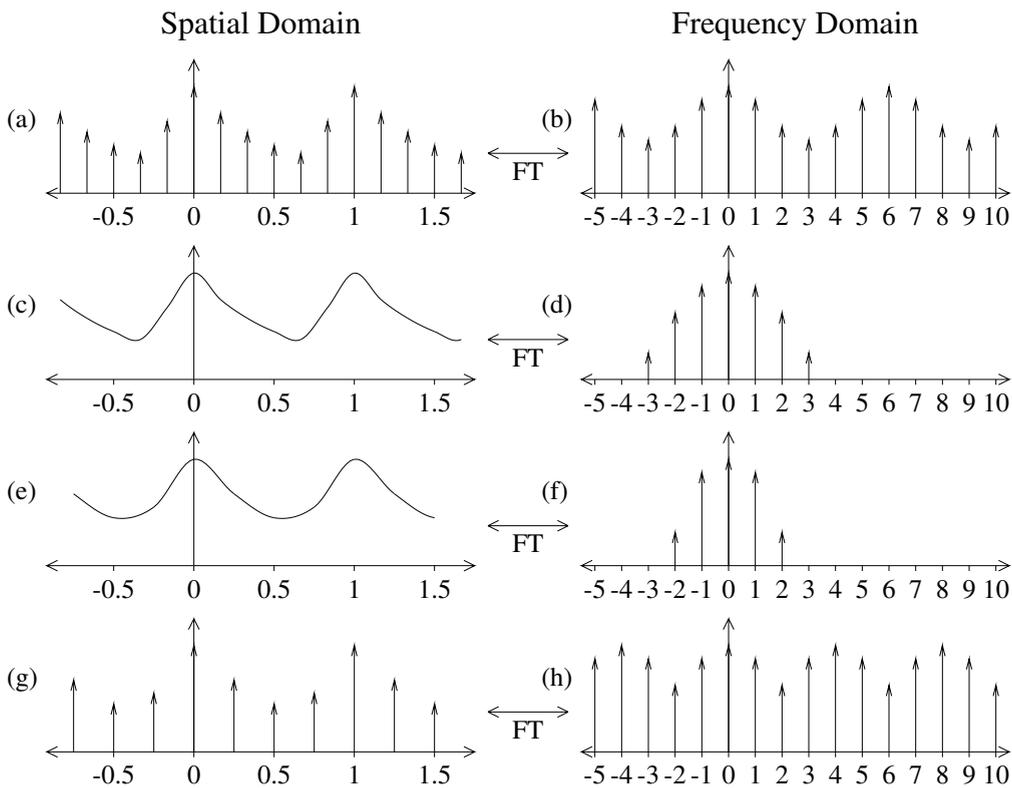


Figure 15: An example of ‘perfect’ minification. The sampled spatial function (a) is transformed to give (b), this is multiplied by a box function producing (d) [and thus generating a continuous function (c) in the spatial domain]. (d) is multiplied by another box function to band-limit it [filtering the spatial domain function (e)], and then sampling is performed to replicate the frequency spectrum (h) and produce a minified image in the spatial domain (g).

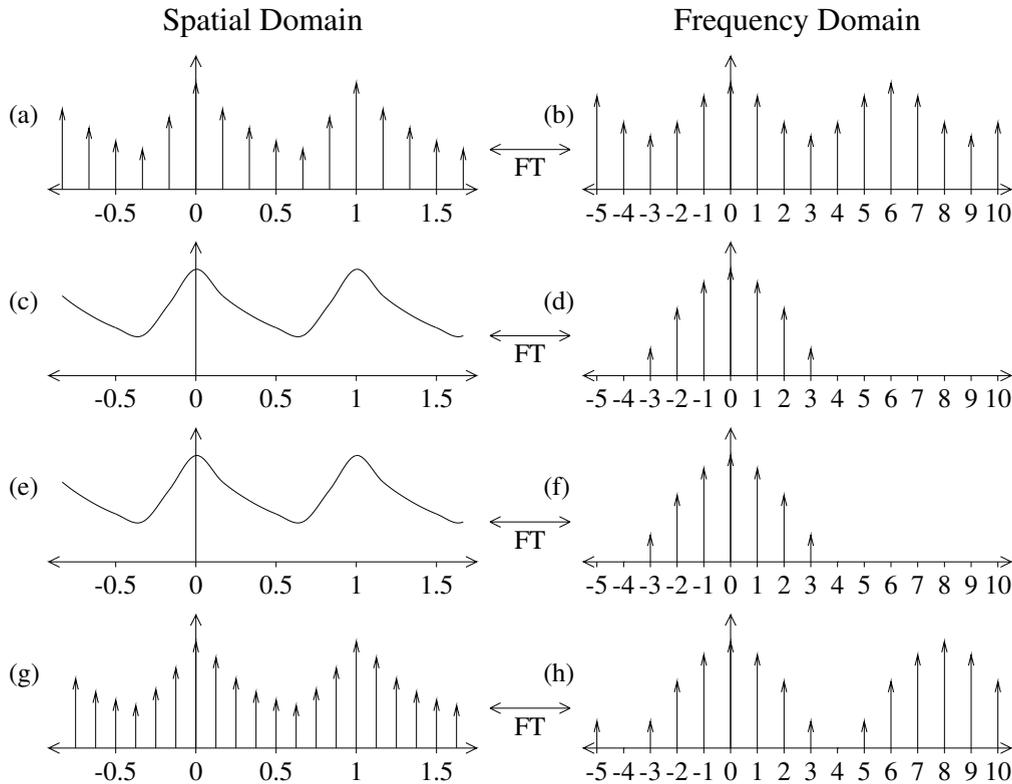


Figure 16: An example of ‘perfect’ magnification. See the caption of figure 15 for a description of the processes. Note that multiplying (d) by the appropriate box function to give (f) has no effect on the function in this magnification case.

quency domain. If we are enlarging the image this makes no difference to the image (figure 16(f)), if reducing it then the higher frequency teeth in the comb function are set to zero and any component at the new Nyquist frequency is halved (figure 15(f)). We can then point sample at the new frequency, which produces periodic replication in the frequency domain. Figure 15(h) shows this for reduction and figure 16(h) for enlargement.

Notice that the first multiplication with a box is redundant for reduction and the second is redundant for enlargement. So, in either case only one multiplication by a box function is required. Note also that the periodic copies move farther apart in enlargement, as if they had been pushed apart at the Nyquist frequency (and its copies); and they have been pushed closer together in reduction, hence the need to prefilter to prevent overlap and thus aliasing. The overlap at the Nyquist frequency (and its copies at $(2N + 1)\nu_N$, $N \in \mathbb{Z}$) appears to be acceptable [Watson, 1986]. Here the negative and positive Nyquist frequencies add together (hence, again, the need for the box function to have half values at the Nyquist frequencies). With a real spatial domain function this will mean that the even component at this frequency will be doubled and the odd component will cancel itself. So, if the original spectrum is non-zero at the Nyquist frequency then samples at the sampling frequency preserve only the even portion of this component. This is known as critical sampling [Watson, 1986, p.4]. Figure 14 shows three sine waves sampled at

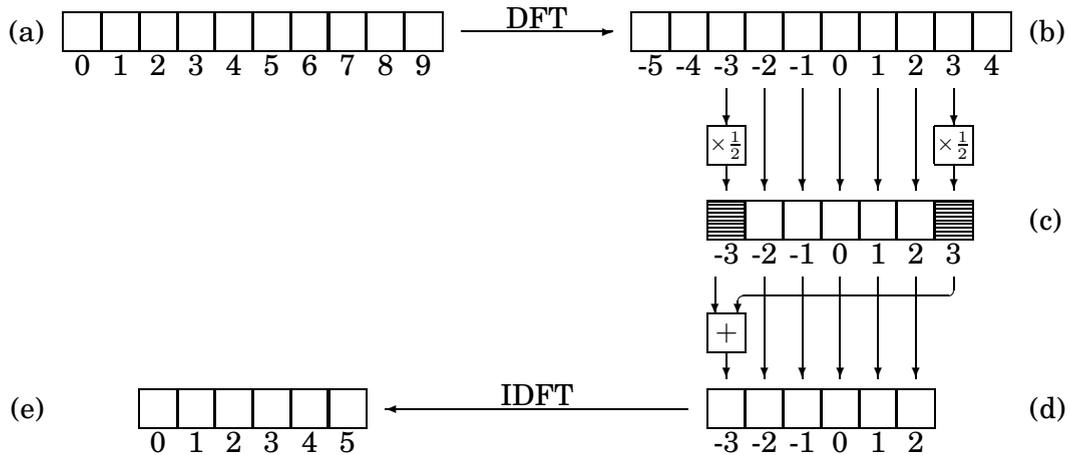


Figure 17: An example of DFT minification. The image, (a), is discrete Fourier transformed, giving (b), which is multiplied by a box function to give (c). The two Nyquist components are collapsed into one, producing (d), which is inverse discrete Fourier transformed, thus generating (e), a minified version of (a).

twice their own frequency, and the perfectly reconstructed version of these: only the even component survives.

Whilst this theoretical explanation clearly shows that a continuous intensity surface is generated and resampled, as a practical reconstruction method a super-skeleton surface is generated, because samples can only be produced at regularly spaced points, not at any arbitrary point. To be able to sample at any point there would need to be an infinite number of points generated in each period, and hence an infinite length of time to evaluate all the points (they must all be evaluated in a single operation using this method).

2.2.2 Practical implementation

This algorithm is implemented using the discrete Fourier transform (normally its fast version: the fast Fourier transform (section 2.2.4)). The process mirrors that shown in figure 15 and figure 16. In this section we consider first the case of discrete reduction and then that of enlargement. In both cases we use even length sequences in our examples. We consider the differences between even and odd length sequences in the next section.

For reduction, the image (figure 17(a)) is discrete Fourier transformed (figure 17(b)). Notice that this transform has only one Nyquist component (here shown at the negative end of the spectrum). The periodic nature of the continuous equivalent means that the other Nyquist component, and all the other copies of this component, have the same value. The transformed data are then multiplied by a box function the same width as the new sample sequence. For an even length sequence, this creates a sequence of length one more than the new sample sequence, with a Nyquist frequency component at each end (figure 17(c)). These two components are added together to give a single Nyquist component, here shown at the negative end of the period (figure 17(d)). Compare this

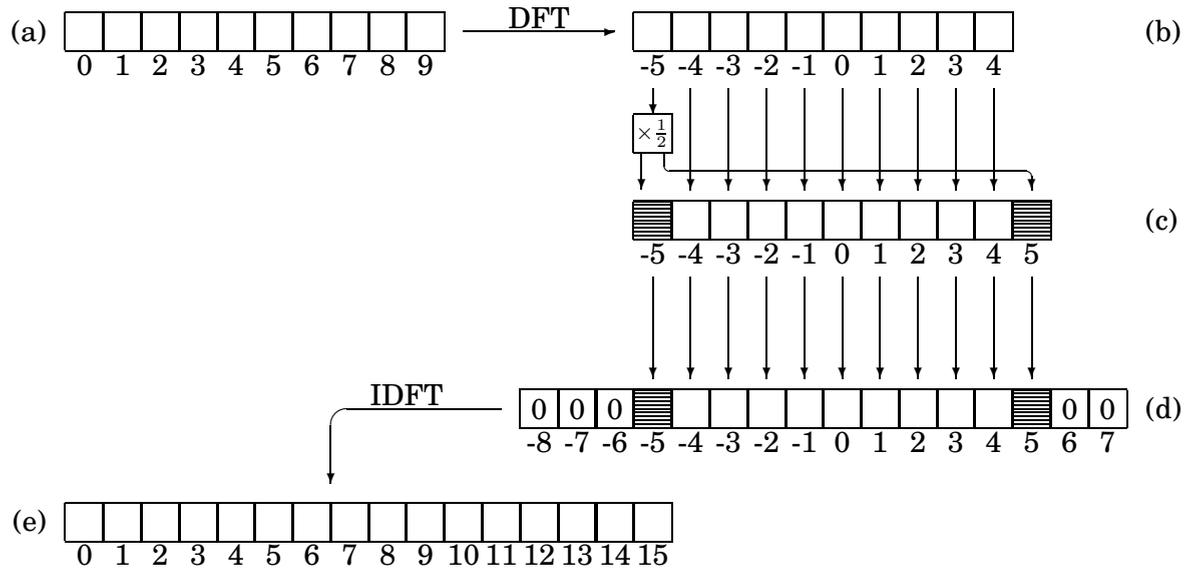


Figure 18: An example of DFT magnification. The image, (a), is discrete Fourier transformed, giving (b). The Nyquist component is split into two halves, producing (c). This is padded with zeros at either end to give (d), which is inverse discrete Fourier transformed, thus generating (e), a magnified version of (a).

with figure 15(f) and (h) where sampling in the spatial domain causes the copies of the spectrum to overlap at the Nyquist frequency and its copies, and hence add up at these points. The modified sequence can now be inverse transformed to give the resampled image (figure 17(e)).

Enlargement is a similar process. The image (figure 18(a)) is discrete Fourier transformed to give its frequency domain representation (figure 18(b)). The Nyquist frequency component is split into two halves, one at each end of the sequence (figure 18(c)). The sequence is now one sample longer. Compare this step with figure 16(b) and (d) where the positive and negative Nyquist components are both halved leaving a sequence with one tooth more in figure 16(d) than the number of teeth in each period in figure 16(b). The altered transformed sequence in figure 18(c) is now padded with zeroes at each end to give a sequence of the desired length (figure 18(d)). Note, in this example, that one more zero is added at the negative end than at the positive end because we are assuming that the Nyquist component is at the negative end of the spectrum. The discrete spectrum is finally inverse discrete Fourier transformed to produce the desired enlarged image (figure 18(e)).

2.2.3 Odd vs even length sequences

All of the examples, up to this point, have used even length sequences. The processing required for an odd length sequence is slightly different and slightly easier. This is due to the fact that there is no frequency component at the Nyquist limit in an odd length sequence. Refer back to the continuous case in figure 15(a) and (b). In general, for

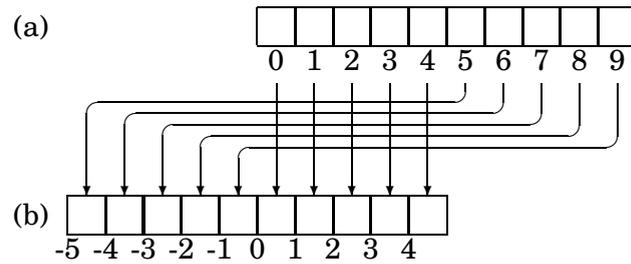


Figure 19: Comparison of the usual FFT output order and the DFT output order used in figure 17 and figure 18. At the top is the usual FFT output order. The Nyquist component is number 5. This can be easily converted to our DFT output order, shown at the bottom. The Nyquist component here is -5 .

a sequence of length L , there are L samples (teeth) in a unit distance in the spatial domain. Each period in the frequency domain is L units long, with the teeth of the comb function at integer locations. The Nyquist frequency is at $\pm \frac{L}{2}$. For an even length sequence there are components (teeth) at these frequencies; for an odd length sequence there are no components at these frequencies because they are not at integer locations. There are therefore no special case frequencies to consider when dealing with an odd-length sequence, because there is no Nyquist component.

2.2.4 Implementation using the FFT

Implementing this algorithm using a fast Fourier transform is straightforward. Two transforms are required, one of the original length and one of the final length. Provided an FFT algorithm can be implemented for both lengths the whole algorithm can be implemented. All FFT algorithms decompose the DFT into a number of successively shorter, and simpler DFTs [Lynn and Fuerst, 1989, p.221]. Thus an FFT of length 2^n , $n \in \mathcal{N}$ is well known and widely used. Decomposition of other, non-prime length DFTs is less widely used. Finally, FFTs for prime length DFTs or lengths with large prime factors are most difficult, because the DFT has to be split into pieces of unequal length.

One important implementation issue is that most FFT algorithms produce N coefficients in the frequency domain ranging from $\nu = 0$ to $\nu = 2\nu_N - 1$. The examples shown here in figures 17 and 18 show them ranging from $\nu = -\lfloor \nu_N \rfloor$ to $\nu = \lfloor \nu_N - \frac{1}{2} \rfloor$. This is no great problem as the values generated are exactly the same because the sequence is periodic with period $2\nu_N$ (see figure 10). Figure 19 shows the equivalence. So long as this is borne in mind in implementation, no problems will arise.

A The inverse Fourier transform of a box function

$$\begin{aligned}
s(x) &= \int_{-\infty}^{\infty} \text{Box}(\nu) e^{i2\pi\nu x} d\nu \\
&= \int_{-\nu_b}^{\nu_b} e^{i2\pi\nu x} d\nu \\
&= \frac{1}{i2\pi x} e^{i2\pi\nu x} \Big|_{-\nu_b}^{\nu_b} \\
&= \frac{1}{i2\pi x} (e^{i2\pi\nu_b x} - e^{-i2\pi\nu_b x}) \\
&= \frac{1}{i2\pi x} (\cos(2\pi\nu_b x) + i \sin(2\pi\nu_b x) - (\cos(2\pi\nu_b x) - i \sin(2\pi\nu_b x))) \\
&= \frac{1}{i2\pi x} (2i \sin(2\pi\nu_b x)) \\
&= \frac{\sin(2\pi\nu_b x)}{\pi x} \\
&= 2\nu_b \text{sinc}(2\nu_b x)
\end{aligned}$$

$\text{sinc}(x)$ is defined in the literature as either $\text{sinc}(x) = \frac{\sin(x)}{x}$ or as $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$. In this thesis we will stick with the latter.

B The Fourier transform of the sinc function

The sinc function is defined as:

$$h(x) = N \frac{\sin \pi N x}{\pi N x}, \quad N > 0$$

It's Fourier transform is:

$$\begin{aligned}
H(\nu) &= \int_{-\infty}^{\infty} h(x) e^{-i2\pi\nu x} dx \\
&= \int_{-\infty}^{\infty} N \frac{\sin(\pi N x)}{\pi N x} [\cos(2\pi\nu x) - i \sin(2\pi\nu x)] dx \\
&= \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\sin(\pi N x) \cos(2\pi\nu x)}{x} dx - \frac{i}{\pi} \int_{-\infty}^{\infty} \frac{\sin(\pi N x) \sin(2\pi\nu x)}{x} dx \\
&= \frac{1}{\pi} A(x, \nu) - \frac{i}{\pi} B(x, \nu)
\end{aligned}$$

Let $y = \pi N x$. Thus: $x = \frac{y}{\pi N}$ and $\frac{dy}{dx} = \frac{1}{\pi N}$.

$$\begin{aligned}
A(x, \nu) &= \int_{-\infty}^{\infty} \frac{\sin(\pi N x) \cos(2\pi\nu x)}{x} dx \\
&= \int_{-\infty}^{\infty} \frac{\sin(y) \cos(2y/N)}{y} dy
\end{aligned}$$

\sin is an odd function, $\frac{1}{y}$ is an odd function, and \cos is an even function, therefore the function that is being integrated is even, and so:

$$A(x, \nu) = 2 \int_0^\infty \frac{\sin(y) \cos(2y/N)}{y} dy$$

$$= \begin{cases} 0, & \left| \frac{2\nu}{N} \right| > 1 \\ \frac{\pi}{2}, & \left| \frac{2\nu}{N} \right| = 1 \\ \pi, & \left| \frac{2\nu}{N} \right| < 1 \end{cases}$$

[Dwight, 1934, 858.9]

$$B(x, \nu) = \int_{-\infty}^\infty \frac{\sin(\pi Nx) \sin(2\pi \nu x)}{x} dx$$

$$= \frac{1}{2} \int_{-\infty}^\infty \frac{\cos(\pi(N - 2\nu)x)}{x} dx - \frac{1}{2} \int_{-\infty}^\infty \frac{\cos(\pi(N + 2\nu)x)}{x} dx$$

[Dwight, 1934, 401.07]

Now, \cos is an even function and $\frac{1}{x}$ is an odd function, so the equation inside each of the integrals is odd, meaning that each integral equates to zero ($\int_{-\infty}^0 \frac{\cos kx}{x} dx = - \int_0^\infty \frac{\cos kx}{x} dx$).

Thus:

$$B(x, \nu) = 0$$

and therefore:

$$H(\nu) = \begin{cases} 1, & |\nu| < \frac{N}{2} \\ \frac{1}{2}, & |\nu| = \frac{N}{2} \\ 0, & |\nu| > \frac{N}{2} \end{cases}$$

C Proof that a signal cannot be simultaneously of finite extent in both the frequency domain and the spatial domain

Assume that $f(x)$ and $F(\nu)$ are a Fourier transform pair. Further assume that $f(x)$ is real, of finite extent, and infinitely piecewise differentiable:

$$f(x) \in \mathcal{R}$$

$$f(x) = 0, |x| > x_K$$

and that $F(\nu)$ is bandlimited:

$$F(\nu) = 0, |\nu| > \nu_N$$

$$\begin{aligned}
F(\nu) &= \int_{-\infty}^{\infty} f(x)e^{-i2\pi\nu x} dx \\
&= \int_{-x_K}^{x_K} f(x)e^{-i2\pi\nu x} dx \\
&= \frac{f(x)e^{-i2\pi\nu x}}{-i2\pi\nu} \Big|_{-x_K}^{+x_K} - \frac{1}{(-i2\pi\nu)} \int_{-x_K}^{x_K} f'(x)e^{-i2\pi\nu x} dx \\
&\quad \text{[Dwight, 1934, 79]} \\
&= \frac{f(x)e^{-i2\pi\nu x}}{-i2\pi\nu} \Big|_{-x_K}^{+x_K} - \frac{f'(x)e^{-i2\pi\nu x}}{(-i2\pi\nu)^2} \Big|_{-x_K}^{+x_K} + \frac{1}{(-i2\pi\nu)^2} \int_{-x_K}^{x_K} f''(x)e^{-i2\pi\nu x} dx \\
&= e^{iax} \left(\frac{f(x)}{ia} - \frac{f'(x)}{(ia)^2} + \frac{f''(x)}{(ia)^3} - \frac{f'''(x)}{(ia)^4} + \dots \right) \Big|_{-x_K}^{x_K}, a = -2\pi\nu \\
&= (\cos(ax) + i \sin(ax)) \left(\frac{f(x)}{(ia)} - \frac{f'(x)}{(ia)^2} + \frac{f''(x)}{(ia)^3} - \frac{f'''(x)}{(ia)^4} + \dots \right) \Big|_{-x_K}^{x_K} \\
&= \cos(ax) \left(\frac{f(x)}{a^2} - \frac{f'''(x)}{a^4} + \dots \right) \Big|_{-x_K}^{x_K} \\
&\quad + \sin(ax) \left(\frac{f(x)}{a} - \frac{f''(x)}{a^3} + \dots \right) \Big|_{-x_K}^{x_K} \\
&\quad + i \cos(ax) \left(-\frac{f(x)}{a} + \frac{f''(x)}{a^3} - \dots \right) \Big|_{-x_K}^{x_K} \\
&\quad + i \sin(ax) \left(\frac{f'(x)}{a^2} - \frac{f'''(x)}{a^4} + \dots \right) \Big|_{-x_K}^{x_K} \\
&= \cos(ax)\phi(a, x) \Big|_{-x_K}^{x_K} \\
&\quad + \sin(ax)\chi(a, x) \Big|_{-x_K}^{x_K} \\
&\quad + i \cos(ax)(-\chi(a, x)) \Big|_{-x_K}^{x_K} \\
&\quad + i \sin(ax)\phi(a, x) \Big|_{-x_K}^{x_K} \\
&= \cos(ax_K)(\phi(a, x_K) - \phi(a, -x_K)) \\
&\quad + \sin(ax_K)(\chi(a, x_K) + \chi(a, -x_K)) \\
&\quad + i \cos(ax_K)(-\chi(a, x_K) + \chi(a, -x_K)) \\
&\quad + i \sin(ax_K)(\phi(a, x_K) + \phi(a, -x_K)) \tag{5}
\end{aligned}$$

Where:

$$\phi(a, x) = \left(\frac{f'(x)}{a^2} - \frac{f'''(x)}{a^4} + \dots \right)$$

and:

$$\chi(a, x) = \left(\frac{f(x)}{a} - \frac{f''(x)}{a^3} + \dots \right)$$

Now, if we set equation 5 to be zero outside the bandlimit of $F(\nu)$ (as we assumed) and remember that $f(x)$ is a real function, then we find that:

$$\begin{aligned}
\phi(a, x_K) &= -\phi(a, -x_K), |a| > \nu_N/2\pi \\
\phi(a, x_K) &= \phi(a, -x_K), |a| > \nu_N/2\pi \\
\chi(a, x_K) &= -\chi(a, -x_K), |a| > \nu_N/2\pi \\
\chi(a, x_K) &= \chi(a, -x_K), |a| > \nu_N/2\pi
\end{aligned}$$

These four equations show that:

$$\begin{aligned}
 \phi(a, x_K) &= 0, |a| > \nu_N/2\pi \\
 \phi(a, -x_K) &= 0, |a| > \nu_N/2\pi \\
 \chi(a, x_K) &= 0, |a| > \nu_N/2\pi \\
 \chi(a, -x_K) &= 0, |a| > \nu_N/2\pi
 \end{aligned} \tag{6}$$

Taking equation 6 as an example:

$$\begin{aligned}
 \phi(a, x_K) &= \left(\frac{f'(x_K)}{a^2} - \frac{f'''(x_K)}{a^4} + \frac{f^{(V)}(x_K)}{a^6} - \dots \right) \\
 &= 0, |a| > \nu_N/2\pi
 \end{aligned}$$

which gives the following equalities:

$$\begin{aligned}
 f'(x_K) &= 0, \\
 f'''(x_K) &= 0, \\
 f^{(V)}(x_K) &= 0 \\
 &\vdots
 \end{aligned}$$

This gives us the result:

$$\phi(a, x_K) = 0, \forall a$$

A similar argument holds for $\phi(a, -x_K)$, $\chi(a, x_K)$ and $\chi(a, -x_K)$. From these and equation 5 it can be seen that:

$$F(\nu) = 0, \forall \nu$$

which implies that:

$$f(x) = 0, \forall x$$

So there exists only one trivial case in which the original assumption holds, that is $f(x) = 0$ and $F(\nu) = 0$. Therefore the Fourier transform of a real, finite-extent function, $f(x)$ cannot be bandlimited, unless $f(x) = 0$. Further, the Fourier transform of a bandlimited function, $F(\nu)$ cannot be a real, finite-extent function, unless $F(\nu) = 0$.

D The Fourier transform of a finite-extent comb

Let the finite extent, one-dimensional comb function, $d(x)$, be defined as:

$$d(x) = \sum_{j=-b}^b \delta(x - j\Delta x)$$

Its Fourier transform is:

$$D(\nu) = \int_{-\infty}^{\infty} \sum_{j=-b}^b \delta(x - j\Delta x) e^{-i2\pi\nu x} dx$$

$$\begin{aligned}
&= \sum_{j=-b}^b e^{-i2\pi\nu j\Delta x} \\
&= \sum_{j=-b}^b \cos 2\pi\nu j\Delta x - i \sin 2\pi\nu j\Delta x \\
&= \cos 0 + 2 \sum_{j=1}^b \cos 2\pi\nu j\Delta x \quad [\sin \text{ odd, cos even}] \\
&= 1 + 2 \sum_{j=1}^b \cos 2\pi\nu j\Delta x \\
&= 1 + 2 \frac{\cos(b+1)\pi\nu\Delta x \sin b\pi\nu\Delta x}{\sin \pi\nu\Delta x} \quad [\text{Dwight, 1934, 420.2}] \\
&= 1 + \frac{\sin((2b+1)\pi\nu\Delta x)}{\sin \pi\nu\Delta x} + \frac{\sin(-\pi\nu\Delta x)}{\sin \pi\nu\Delta x} \quad [\text{Dwight, 1934, 401.05}] \\
&= 1 + \frac{\sin((2b+1)\pi\nu\Delta x)}{\sin \pi\nu\Delta x} - 1 \\
&= \frac{\sin((2b+1)\pi\nu\Delta x)}{\sin \pi\nu\Delta x}
\end{aligned}$$

In two-dimensions:

$$\begin{aligned}
d(x, y) &= d(x) * d(y) \\
&= \sum_{j=-b_x}^{b_x} \sum_{k=-b_y}^{b_y} \delta(x - j\Delta x, y - k\Delta y)
\end{aligned}$$

Implying that the two-dimensional Fourier transform is:

$$\begin{aligned}
D(\nu_x, \nu_y) &= D(\nu_x) \times D(\nu_y) \\
&= \frac{\sin((2b_x+1)\pi\nu_x\Delta x)}{\sin \pi\nu_x\Delta x} \times \frac{\sin((2b_y+1)\pi\nu_y\Delta y)}{\sin \pi\nu_y\Delta y}
\end{aligned}$$

Bibliography

- Blinn, J.F. [1990]** “Wonderful world of video”, *IEEE Computer Graphics and Applications*, Vol. 10, No. 3, May, 1990, pp.83–87.
- Cook, R.L. [1986]** “Stochastic sampling in computer graphics”, *ACM Transactions on Graphics*, Vol. 5, No. 1, Jan, 1986, pp.51–72.
- Crow, F.C [1978]** “The use of grayscale for improved raster display of vectors and characters”, *Computer Graphics*, Vol. 12, No. 3, Aug, 1978, pp.1–5.
- Durand, C.X. [1989]** “Bit map transformations in computer 2-D animation”, *Computers and Graphics*, Vol. 13, No. 4, 1989, pp.433–440.
- Dwight, H.B. [1934]** *Tables of Integrals and other Mathematical Data*, The MacMillan Company, New York, 1934.
- Fiume, E.L. [1986]** “A mathematical semantics and theory of raster graphics”, PhD thesis, technical report CSRI-185, Computer Systems Research Group, University of Toronto, Toronto, Canada M5S 1A1. Published, with minor additions, as Fiume [1989].
- Fiume, E.L. [1989]** *The Mathematical Structure of Raster Graphics*, Academic Press, San Diego, 1989, ISBN 0–12–257960–7. Based on Fiume [1986].
- Foley, J.D. and van Dam. A. [1982]** *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading, Massachusetts, USA, 1982, ISBN 0–201–14468–9.
- Foley, J.D., van Dam, A., Feiner, S.K. and Hughes, J.F. [1990]** *Computer Graphics: Principles and Practice* (‘second edition’ of Foley and van Dam [1982]), Addison-Wesley, Reading, Massachusetts, USA, 1990, ISBN 0–201–12110–7.
- Fraser, D. [1987]** “A conceptual image intensity surface and the sampling theorem”, *Australian Computer Journal*, Vol. 19, No. 3, Aug, 1987, pp.119–125.
- Fraser, D. [1989a]** “Interpolation by the FFT revisited — an experimental investigation”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 5, May, 1989, pp.665–675.
- Fraser, D. [1989b]** “Comparison at high spatial frequencies of two-pass and one-pass geometric transformation algorithms”, *Computer Vision, Graphics and Image Processing*, Vol. 46, No. 3, Jun, 1989, pp.267–283.
- Gonzalez, R.C. and Wintz, P. [1977]** *Digital Image Processing*, Addison-Wesley, 1977, ISBN 0–201–02596–5.
- Heckbert, P.S. [1989]** “Fundamentals of texture mapping and image warping”, Master’s thesis, report number UCB/CSD 89/516, Computer Science Division (EECS), University of California, Berkeley, California 94720, USA; June, 1989.
- Lynn, P.A. and Fuerst, W. [1989]** *Introductory digital signal processing with computer applications*, John Wiley and Sons, Ltd, Chichester, 1989, ISBN 0–471–91564–5.

- Marion, A. [1991]** *An introduction to image processing*, Chapman and Hall, 2–6 Boundary Row, London, 1991, ISBN 0–412–37890–6.
- Mitchell, D.P. and Netravali, A.N. [1988]** “Reconstruction filters in computer graphics”, *Computer Graphics*, Vol. 22, No. 4, Aug, 1988, pp.221–288.
- Murch, G.M. and Weiman, N. [1991]** “Gray-scale sensitivity of the human eye for different spacial frequencies and display luminance levels”, *Proceedings of the Society for Information Display*, Vol. 32, No. 1, 1991, pp.13–17.
- NMFPT, [1992]** National Museum of Film, Photography, and Television, Bradford, West Yorkshire.
- Pavlidis, T. [1982]** *Algorithms for Graphics and Image Processing*, Springer-Verlag, 1982.
- Pavlidis, T.[1990]** “Re: Comments on ‘Stochastic Sampling in Computer Graphics’ [Cook, 1986]”, Letter to the Editor, *ACM Transactions on Graphics*, Vol. 9, No. 2, Apr, 1990, pp.233–326.
- Petersen, D.P. and Middleton, D. [1962]** “Sampling and reconstruction of wave-number-limited functions in N -dimensional Euclidean spaces”, *Information and Control*, Vol. 5, No. 4, Dec, 1962, pp.279–323.
- Prasad, K.P. and Satyanarayana, P. [1986]** “Fast interpolation algorithm using FFT”, *Electronics Letters*, Vol. 22, No. 4, 13 Feb, 1986, pp.185–187.
- Pratt, W.K [1978]** *Digital Image Processing*, John Wiley and Sons, New York, USA, 1978, ISBN 0–471–01888–0.
- Rosenfeld, A. and Kak, A.C. [1982]** *Digital Picture Processing* (second edition, two volumes), Academic Press, Orlando, Florida, USA, 1982, ISBN 0–12–597301–2.
- Schafer, R.W. and Rabiner, L.R. [1973]** “A digital signal processing approach to interpolation”, *Proceedings of the IEEE*, Vol. 61, No. 6, Jun, 1973, pp.692–702.
- Shannon, C.E. [1949]** “Communication in the presence of noise”, *Proceedings of the IRE*, Vol. 37, No. 1, Jan, 1949, pp.10–21.
- Turkowski, K. [1986]** “Anti-aliasing in topological color spaces”, *Computer Graphics*, Vol. 20, No. 4, Aug, 1986, pp.307–314.
- Watson, A.B. [1986]** “Ideal shrinking and expansion of discrete sequences”, NASA Technical Memorandum 88202, Jan, 1986, NASA, Ames Research Center, Moffett Field, California 94035, USA.
- Whittaker, E.T. [1915]** “On the functions which are represented by the expansions of the interpolation theory”, *Proceedings of the Royal Society of Edinburgh*, Vol. 35, Part 2, 1914–1915 session (issued separately July 13, 1915), pp.181–194.