Uniform Sampling and Reconstruction

May 16, 2011

Contents

1 Introduction

One very important area of computer graphics is signal processing. People normally only have a rough idea of what it is and what possibilities this area offers. The aim of this document is to give a brief description of uniform sampling and reconstruction, which basically is the most common way of signal processing.

The reader should get an overview about how it works and which problems can occur.

First an overview on signals and functions will be given and the difficulties of signal processing will be explained. Next we put our focus on the basic sampling process using convolution for merging signal and filters. There we will see what Nyquist Frequency is and what it is used for. Aliasing will then be discussed with an example. For structuring the already stated concepts we will introduce the 1D sampling theorem. Then basic filters will be explained and our main interest will move on to the Convolution Theorem and the Fourier Transform. The basics of the Fourier Transform will be discussed and we will see how the Fourier Transform helps us doing sampling and reconstruction. The second part of this paper mainly deals with reconstruction.

1.1 Signals and Functions

The basic concept from the field of signal processing is the concept of a *signal*. The *signal* is a function that conveys information. Signals are often thought of as functions of time (signals in the *temporal domain*) or as functions of spatial coordinates (signals in the *spatial domain*). Most of the signals in computer graphics are in the spatial domain. For example, we can think of images and 3D volumes as intensity variations over space. An example of a 1D signal in the spatial domain is given in Figure 1.

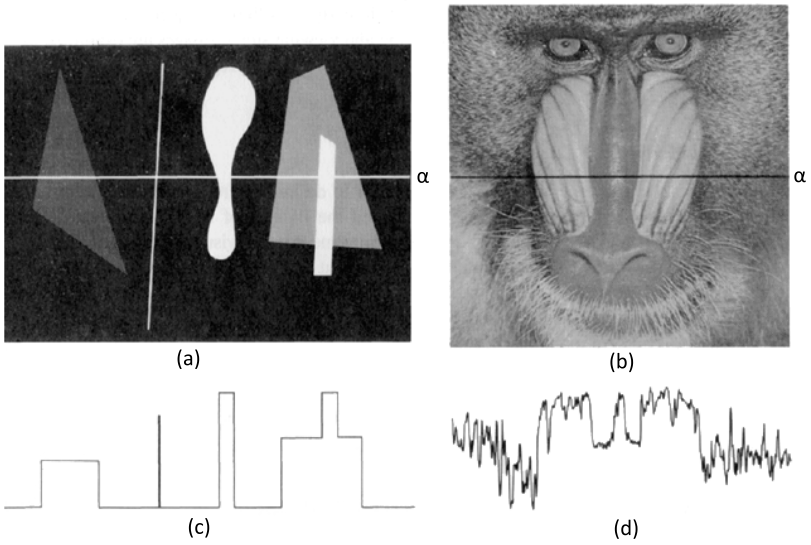


Figure 1: Image. (a) Graphical primitives. (b) Mandrill. (c) Intensity plot of scan line α in (a). (d) Intensity plot of scan line α in (b). (Taken from [1])

Signals can be additionally classified as *continuous* or *discrete*. A *continuous signal* is defined at a continuum of positions in space; a *discrete signal* is defined

at a set of discrete points in space. The functions of most analog signals, which come from the real world, are mainly continuous. The functions of most digital signals are discrete.

Looking at the definitions above and comparing them with the possibilities of computers, which means digital representation of information, we mainly see two problems:

1. Functions are continuous
2. Functions are infinite

The problems we deal with are quite old: Since the digital world was developed, people tried to convert analog signals into digital ones and vice versa in order to be able to transfer and modify them in an easy way. Problems started with the invention of telegraph system some hundred years ago. At that time the Morse code was used to encode messages in a reasonable way. The system had quite big advantages: low costs, easier transport and a cheap opportunity to modify. But also disadvantages remained: transferring signals at reasonable costs always means a loss of information. But we have a possibility to influence which information will be lost in order to keep the most useful information.

A continuous signal may contain arbitrarily fine detail in the form of very rapid (high-frequency) variations in its value as its continuous parameter is changed. Since a discrete signal can change value only at discrete points, it clearly has a maximum rate of variation. Therefore, converting a continuous signal to a finite array of values may result in a loss of information. The main goal is to ensure that as little information as possible is lost. The process of selecting a finite set of values from a signal is known as *sampling*, and the selected values are called *samples*. Once we have selected these samples, we must then display them using a process, known as *reconstruction*, that attempts to recreate the original continuous signal from the samples.

Taking things back to computer graphics a common example might be drawing a line in a screen matrix or processing a monitor scan-line. To sum it up digitizing an analog signal is what we do by sampling and turning a digitized signal back to its analog representation will be done by reconstruction.

2 Sampling

As stated above, we are talking about processing aperiodic continuous signals representing an image to make display and computation possible. Digital image processing is very expensive; mainly because all the information has to be represented properly in order to make any processing possible. All transformations to achieve these aims are costly to compute, create a large amount of data and will always be a compromise between cost and quality.

What happens to the signal is the following: We decide to use a certain sampling interval (*uniform sampling*) and take samples there. This way of sampling often leads to big errors in the results because we leave out parts of

our functions which do not influence the output at all. What we can do to avoid this is building an average over an interval of our function just to let all parts of the original function influence the result, which actually should represent the whole function. To get an idea about this we will now take a look at an example.

2.1 The Basic Sampling Process

First we introduce two functions we would like to have a closer look at:

$$f_1(x) = -\frac{x^4}{3} + \frac{x^3}{2} - 3x + 30$$

and

$$f_2(x) = 6x \sin \sqrt{|x^5|} + 20$$

Looking at the plotted functions, we see that $f_1(x)$ varies less over time than $f_2(x)$ does. Functions may consist of many different frequencies as explained with the Fourier Transform. Going back to the above mentioned difference between $f_1(x)$ and $f_2(x)$, this is a quite important fact because it determines our sampling frequency. We will get back to this later.

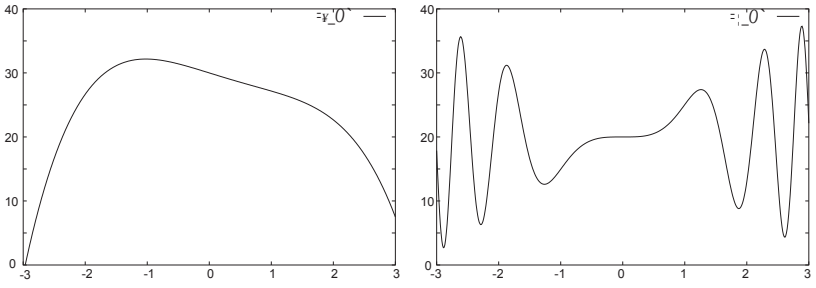


Figure 2: The function plots for $f_1(x)$ and $f_2(x)$.

We want both functions sampled which means that we take values at certain positions based on a regular interval. In our example we sample at an interval (also called sampling frequency) of 1. The positions we sample at are [-3; -2; -1; 0; 1; 2; 3]. The result can be seen in Figure 3.

Now we have two sampled functions from which we may guess the original function. If we do so with $f_1(x)$ we can at least very easily imagine that we might come quite dose to the original function whereas doing so with $f_2(x)$ results in a completely different function. The reason for this is that for $f_2(x)$ we took too few samples. We have to increase the sampling density for being able to better reconstruct this function. This raises the question for the best sampling frequency.

The sampling works best with sampling at two times the highest frequency of the function. This is called *Nyquist frequency* $F_{N(f)} = 2u$. Sampling theory tells

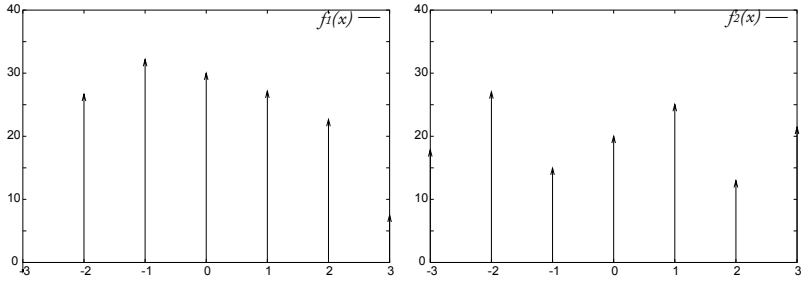


Figure 3: Results of sampling functions $f_1(x)$ and $f_2(x)$.

us that a signal can be properly reconstructed from its samples if the original signal is sampled just above the Nyquist frequency. An example of sampling at the Nyquist frequency is given in Figure 4. Here we can see that the frequency of the signal is captured correctly, but the amplitude depends on the sampling positions. Sampling has to be above the Nyquist frequency to fully recover the analog signal.

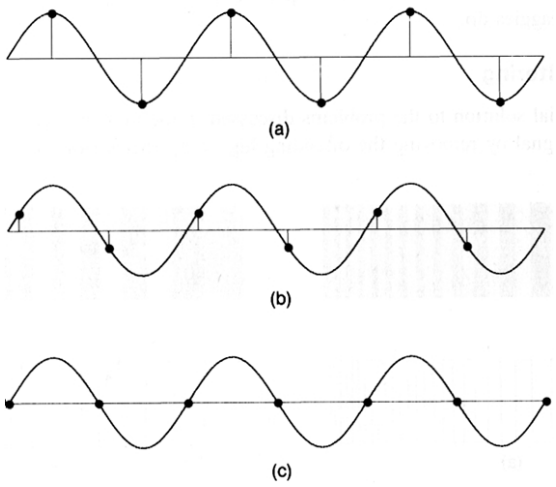


Figure 4: Sampling at the Nyquist frequency (a) at peaks, (b) between peaks, (c) at zero crossings. (Taken from [1])

An example of sampling below the Nyquist frequency is given in Figure 5. Here we can see that the frequency of the signal is captured incorrectly like we are sampling a lower-frequency signal. We will discuss the Nyquist frequency later when talking about reconstruction.

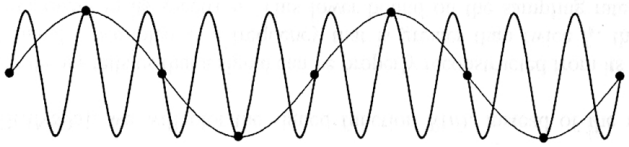


Figure 5: Sampling below the Nyquist frequency. The reconstructed signal has a lower frequency than the original signal.(Taken from [1])

2.1.1 Aliasing

The phenomenon of high frequencies masquerading as low frequencies in the reconstructed signal is known as *aliasing*: The high-frequency components appear as though they were actually lower-frequency components. Aliasing happens because of the reduction of continuous information to periodic samples trying to provide the same information. A common example for this is a movie where a continuous movement is reduced to 24 frames per second. Imagine a western movie with a stagecoach seen from the side. At a certain speed wheels seem to rotate backwards.

To sum it up at sampling time we have to care for sampling frequency which should be at least just above the Nyquist Frequency. Due to this fact it is a good idea to apply a low-pass filter before starting the sampling process. This will normally reduce the Nyquist Frequency and therefore make sampling less expensive to compute. Also the amount of data resulting from the sampling will be reduced.

2.2 The Sampling Process in Computer Graphics

Sampling processes are wide-spread in computer graphics. A *continuous signal* is defined at a continuum of positions in space: a *discrete* signal is defined at a set of discrete points in space. Before scan conversion, the projection of 3D objects onto the view plane may be treated as a continuous 2D signal whose value at each infinitesimal point in the plane indicates the intensity at that point. In contrast, the array of pixel values in the graphics system's frame buffer is a discrete 2D signal whose value is defined only at the positions in the array. The rendering algorithms must determine the intensities of the finite number of pixels in the array, so that they best represent the continuous 2D signal defined by the projection. Now lets consider some examples of sampling used in computer graphics.

2.2.1 Point Sampling

A straightforward way to select each pixel's value is known as *point sampling*. In point sampling, we select one point for each pixel, evaluate the original signal at this point, and assign its value to the pixel. The points that we select are typically arranged in a regular grid, as shown in Figure 6.

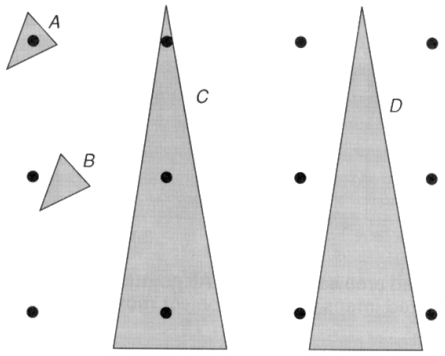


Figure 6: Point-sampling process. Samples are shown as dots. (Taken from [1])

Some important features of the signal, however, may be missed (objects B and D in Figure 6). Sampling at a higher rate (*supersampling*), we can generate images with more pixels representing each portion of the picture.

2.2.2 Area Sampling

The problem of objects falling between samples and being missed suggests another approach: integrating the signal over the area of a square centered on each grid point and selecting its average intensity as that of the pixel. This technique is called *unweighted area sampling*. Unweighted area sampling has drawbacks: a small black object wholly contained inside of one of the pixels and surrounded by a white background may move freely inside the pixel, and for each position the value computed for the pixel remains the same. Thus, the object causes the image to change only when it crosses pixel boundaries. The object's contribution to the pixel's intensity can be *weighted* by its distance from the pixel's center: the farther away it is, the less it should contribute. This technique is called *weighted area sampling*. This allows us to assign different weights to different parts of the pixel.

2.3 Sampling Theory

Sampling theory provides an elegant mathematical framework to describe the relationship between a continuous signal and its samples. So far, we have considered signals in the *spatial domain*; that is, we have represented each of them as a plot of intensity against spatial position. A signal may also be considered in the *frequency domain*; that is, we may represent it as a sum of sine waves, possibly offset from each other (the offset is called *phase shift*), and having different frequencies and amplitudes. Each sine wave represents a component of the signal's *frequency spectrum*. We sum these components in the spatial domain by summing their values at each point in space. An example of periodic and nonperiodic signals represented as the sum of phase-shifted sine waves

whose frequencies are integral multiples (harmonics) of the signal's fundamental frequency is given in Figure 7.

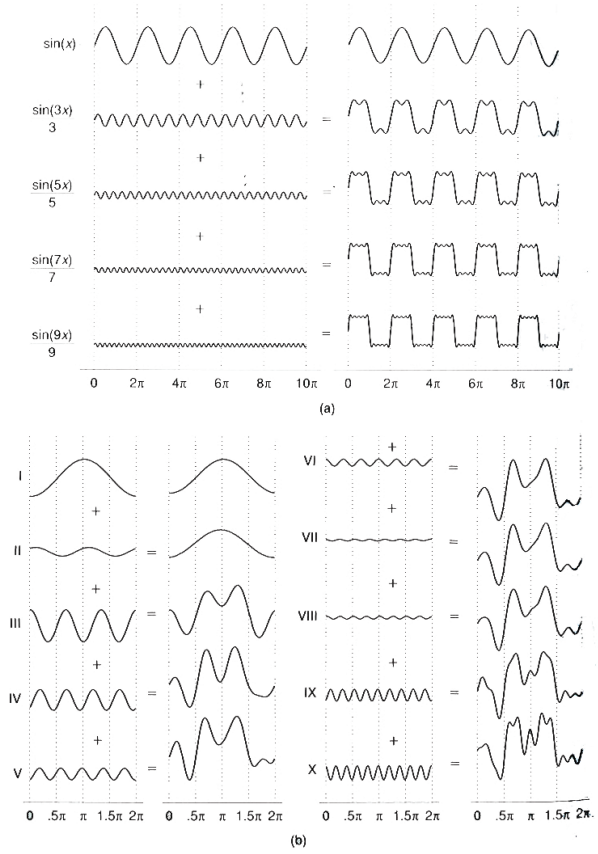


Figure 7: A signal in the spatial domain is the sum of phase shifted sines. Each component is shown with its effect on the signal shown at its right. (Taken from [1])

Determining which sine waves must be used to represent a particular signal is the central topic of *Fourier analysis* [2]. The *Fourier transform* (FT) is used as a tool to switch from spatial to frequency domain and vice versa. In the frequency domain a function is considered to be a sum of sine and cosine waves. The FT does not change the function in any way. It is just seen from a different point of view. The FT of f is called $F(u)$, whose argument u represents frequency. The value $F(u)$, for each frequency u, tells how much (i.e., the amplitude phase shift) of the frequency u appears in the original signal $f(x)$. The function $F(u)$ is therefore called the *representation of f (the signal) in the frequency domain*; $f(x)$ is called the representation of the signal in the spatial domain. The FT of

a continuous, integrable signal $f(x)$ is defined by

$$F(u) = \int_{-\infty}^{+\infty} f(x)[\cos 2\pi ux - i \sin 2\pi ux]dx,$$

where $i = \sqrt{-1}$ and u represents the frequency of a sine and cosine pair. For each u, the value of $F(u)$ is a complex number $R(u) + iI(u)$. This is a clever way of encoding the *phase shift* and *amplitude* of the frequency u component of the signal. The amplitude (or *magnitude*) of $F(u)$ is defined by

$$|F(u)| = \sqrt{R^2(u) + I^2(u)},$$

and the phase shift (or the *phase angle*) is given by

$$\phi(u) = \tan^{-1}[\frac{I(u)}{R(u)}].$$

The *inverse FT (IFT)* transforms the signal from frequency to spatial domain

$$f(x) = \int_{-\infty}^{+\infty} F(u)[\cos 2\pi ux + i \sin 2\pi ux]du.$$

The *discrete version of the FT* is

$$F(u) = \sum_{0 \le x \le N-1} f(x)[\cos(2\pi ux/N) - i \sin(2\pi ux/N)], 0 \le u \le N-1,$$

and the *inverse discrete FT* is

$$f(x) = \frac{1}{N} \sum_{0 \le u \le N-1} F(u)[\cos(2\pi ux/N) + i \sin(2\pi ux/N)], 0 \le x \le N-1.$$

By choosing a sufficiently high sampling rate for the discrete FT, a good approximation to the behavior of the continuous FT is obtained for most signals. The discrete FT may also be computed more efficiently by using a clever reformulation known as the *fast FT* [3].

The FT is not the only way to transform the signal from spatial to frequency domain and vice versa. Alternatives are the *Hartley transform* and the *wavelet transform*. The FT is the most commonly used frequency transform.

2.4 Low-Pass Filtering

If we could create a new signal by removing problematic high frequencies from the original signal, then the new signal could be reconstructed exactly from a finite number of samples. The more high frequencies we remove, the lower the sampling frequency is needed, but the less the signal resembles the original one. This process is known as *bandwidth limiting* or *band limiting* the signal. It is also known as *low-pass filtering*. The interesting question is what a low-pass filter does and how we can apply it to a function.

2.4.1 Convolution and Convolution Theorem

To filter one function with another function we use *convolution*. The convolved function is a sliding weighted average of one function using the other one for providing the weights. The convolution of two signals $f(x)$ and $g(x)$, written as $f(x)^* g(x)$, is a new signal $h(x)$ defined as follows. The value of $h(x)$ at each point is the integral of the product of $f(x)$ with the filter function $g(x)$ shifted such that its origin is at that point. So, $h(x)$ is a weighted average of the neighborhood around each point of the signal $f(x)$, weighted by filter $g(x)$ centered at the point. The size of the neighborhood (or filter's *support*) is the size of the domain where the filter $g(x)$ is nonzero.

$$h(x) = f(x)^* g(x) = \int_{-\infty}^{+\infty} f(\tau) g(x - \tau) d\tau.$$

The filter function $g(x)$ is often called the *convolution kernel* or *filter kernel*. Two examples of convolution are given in Figure 8.

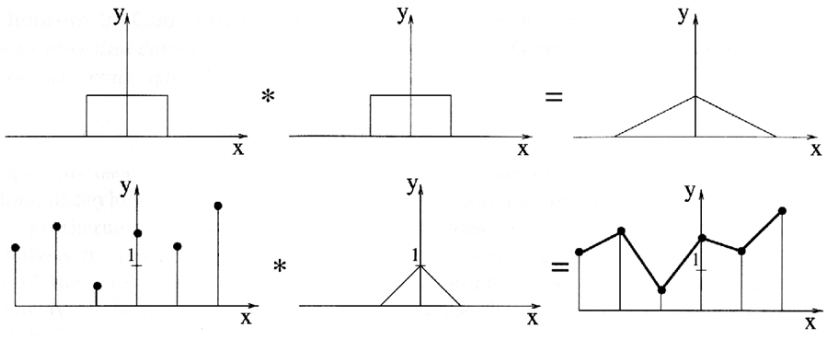


Figure 8: Examples of convolution.

If two functions in spatial domain are convolved their representations in frequency domain are multiplied and vice versa. This is stated by the *convolution theorem*. The FT of the convolution of two functions is equivalent to the product of the FTs of both input signals, and vice versa.

$$f_1 * f_2 \equiv F_1 F_2$$

$$F_1 * F_2 \equiv f_1 f_2$$

Seen from the point of view of performance, the FT and the multiplication in frequency domain can be much cheaper than doing the convolution in spatial domain. The perfect low-pass filter in the frequency domain is multiplication with a box filter, which corresponds to a convolution with the *sinc* function in the spatial domain.

$$sinc = \begin{cases} \frac{sin(\pi x)}{\pi x} \text{ if } x \neq 0 \\ 1 \text{ if } x = 0 \end{cases}$$

But the *sinc* function is infinite (this filter has infinite support). Therefore we use a truncated version of the *sinc* or other not perfect filter functions with finite support for the filtering in the spatial domain. Common filter function representations are shown in Figure 9.

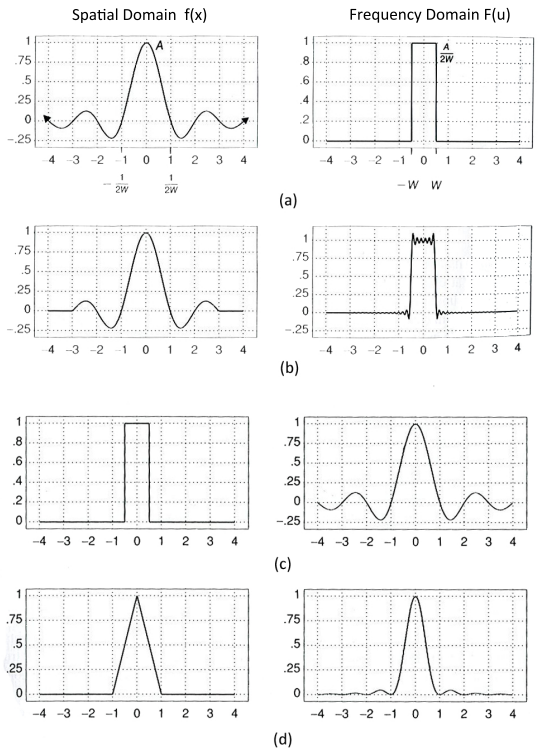


Figure 9: (a) Sinc in spatial domain is box in frequency domain. (b) Truncated sinc in spatial domain is ringing box in frequency domain. (c) Box in spatial domain is sinc in frequency domain. (d) Tent in spatial domain is $sinc^2$ in frequency domain. (Taken from [1])

3 Reconstruction

After sampling is done, we have sampled functions represented by values at certain positions. In uniform sampling these positions are located at regular intervals. To make a function continuous we have to reconstruct it again. Basically this means to convert the digitized signal back into an analog one.

Corresponding to the sampling theory, the frequency spectrum of the sampled functions look like that of the original signal, replicated at multiples of the sampling frequency f_s. Lets see how it works. Sampling a signal corresponds to multiplying it in the spatial domain by a *comb* function (as shown

in Figure 10(a)). The comb function has a value of 1 in the sample points and a value of 0 everywhere else. In frequency domain sampling corresponds to a convolution with the FT of a comb function. The FT of a comb function is just another comb with teeth at multiples of f_s (see Figure 10(b)). The height of the teeth is f_s in cycles per pixel. So the FT of a $comb(f_s)$ function results in a $comb(\frac{1}{f_s})$ function. Convolution of the FT of the comb function and the FT of an original signal will replicate the original spectrum (see Figure 10(d)). All the replications of the original spectrum (in the middle) are also called the *shadow spectra*. The higher f_s is the larger is the spacing between the spectra. When f_s approaches infinity there will be a single spectrum only. Selecting a sufficiently high f_s allows getting the replicated spectra which are located far apart from each other. Sampling below the Nyquist frequency means that the teeth of the $comb(f_s)$ are too far apart and the teeth of the $comb(\frac{1}{f_s})$ are too close together. Therefore the resulting shadow spectra will overlap, which will cause aliasing artifacts. This makes error-free reconstruction not possible.

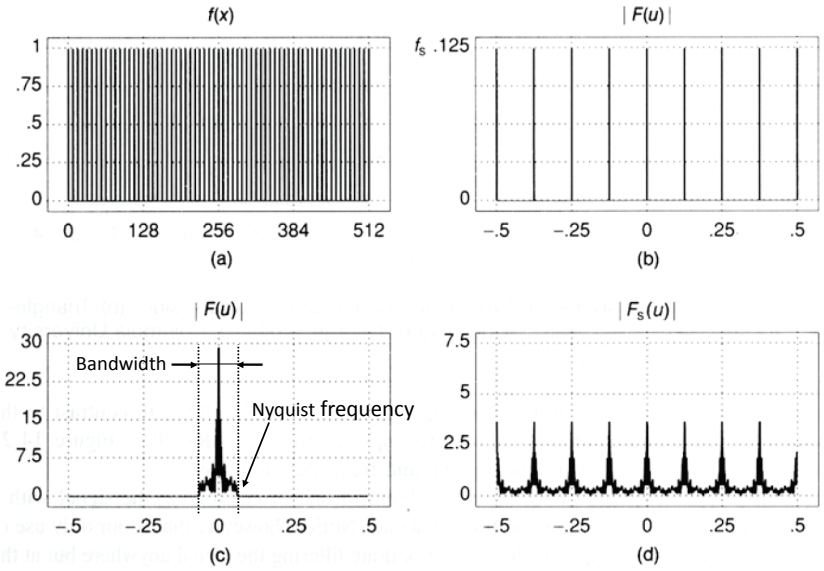


Figure 10: (a) Comb function in spatial domain. (b) Comb function in frequency domain. (c) a FT of an original signal (d) Convolution of the FT of the comb function and FT of an original signal. The spectrum of the sampled signal is replicated. (Taken from [1])

The sampled signal at a finite sampling frequency has an infinite frequency spectrum. In frequency domain reconstruction means: remove the shadow spectra and retain only the original spectrum. Multiplying the spectrum with the box filter in the frequency domain can be used for this purpose. This corresponds to convolution of the sampled signal with a sinc function in the spatial

domain. An example showing the influence of adequate and inadequate sampling rates on the reconstruction results is shown in Figure 11(a). If the sampling frequency is too low, the shadow spectra are overlaped (see Figure 11(b)). The reconstruction then fails to remove overlapping spectra and separate them from the original spectrum. This results in erroneous high frequency contributions to the reconstruction result.

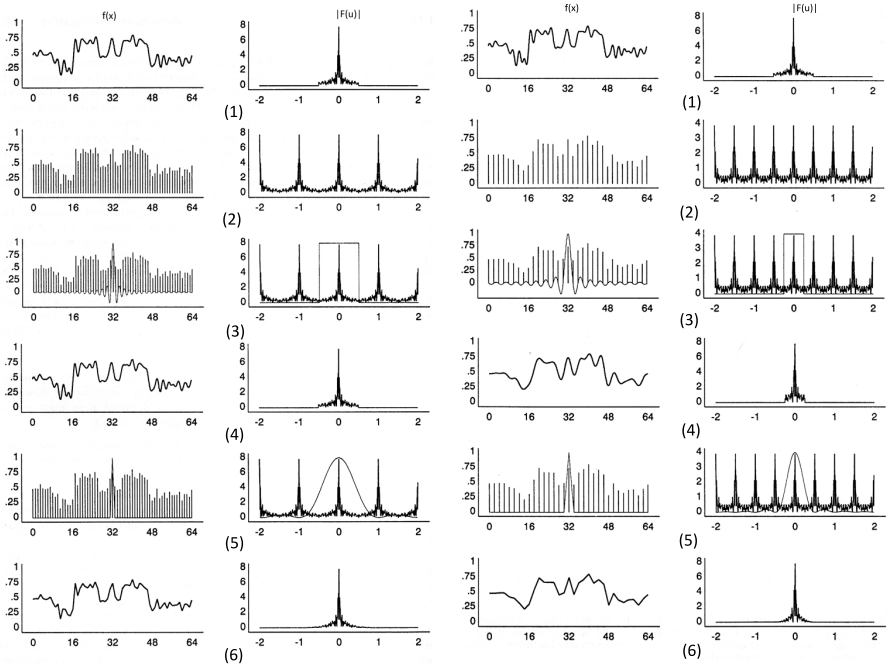


Figure 11: (1) Shows the original signal (2) Sampled signal. (3) Sampled signal before reconstruction using sinc filter. (4) Signal reconstructed with sinc. (5) Sampled signal before reconstruction with tent filter. (6) Signal reconstructed with tent filter. (Taken from [1])

If we have a band limited function sampled with its Nyquist frequency, then this function can be perfectly reconstructed by using the *sinc* function. The main problem with the usage of the *sinc* filter is that it is infinite. Just truncating the *sinc* leads to strong reconstruction errors. Another technique is the usage of a windowed *sinc* function. This basically means to use just a part of the sinc filter at the most relevant region around the origin.

So far we discussed the perfect reconstruction filter *sinc* and stated that a complete reconstruction of a function from its samples is possible if a band limited function is sampled at its Nyquist frequency which is two times its bandwidth. We did all this for the sake of simplicity in 1D. But exactly the same can be done in 2D and 3D. The only difference is that every dimension

has its own bandwidth and Nyquist frequency. So we have 2 or 3 different sampling distances.

Linear interpolation means that the discrete samples are connected by line segments. In our terminology this means that the discrete signal is convolved with a *tent filter*. The tent filter has a value of 1 at zero and is going down to zero at position 1 and -l. The result of convolving the discretized function $f_2(x)$ with the tent filter is given in Figure 12. The reason for this is the too low sampling

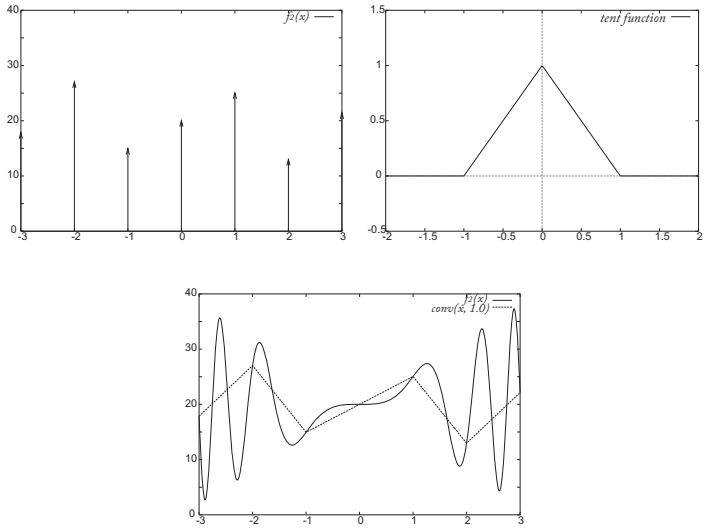


Figure 12: Linear interpolation of $f_2(x)$ samples.

frequency. The side effects are called aliases. We can think of the reconstruction process as described in the section about convolution: The tent is shifted in both directions and a weighted average of both neighbor values is computed. This can either be achieved analytically or by transforming both functions to the frequency domain, multiplying them and transforming the result back.

Nearest neighbor interpolation means that at an arbitrary position x the intensity value of the closest sample point is taken as reconstructed value. Nearest neighbor interpolation produces a piecewise constant (discontinuous) result. In our terminology this means that the discrete singal is convolved with the box filter. An example of the nearest neighbor interpolation is given in Figure 13.

To summarize: The non-ideal reconstruction (i.e., using not the sinc in spatial domain and the box filter in frequency domain) causes two types of aliasing in the resonctruction:

- the original spectrum is not exactly recovered (e.g., high frequencies are dampened)
- shadow spectra not fully eliminated (erroneous high frequency contribu-

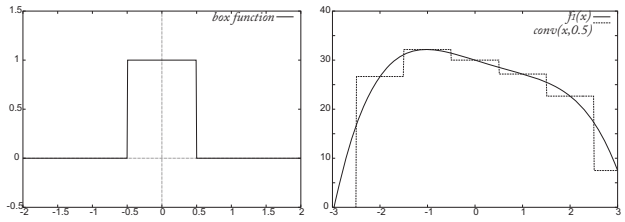


Figure 13: Nearest neighbor interpolation of $f_1(x)$ samples.

tions to the result)

4 Summary

Sampling a signal or image is a critical process. Errors done by sampling at wrong frequencies can never again be corrected. Errors due to under-sampling are called aliasing. Band-limited signals sampled at their Nyquist frequency can be perfectly restored by using the ideal *sinc* filter.

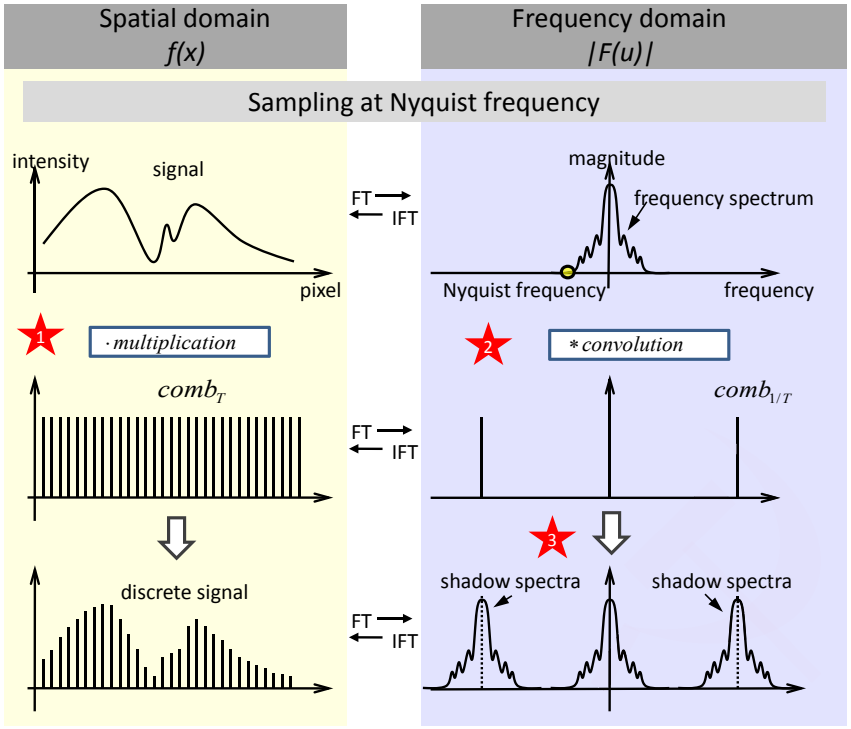
References

[1] Foley J. D., Van Dam A., Feiner S. K. and Hughes J. S., *Computer Graphics: Principles and Practice, Chapter: Aliasing and Antialiasing*, Addison-Wesley Publishing Company, 2nd edition, 1990

[2] Gonzales, R., and P. Wintz, *Digital Image Processing*, second edition, Addison Wesley, Reading, MA, 1987

[3] Brigham, E. O., *The Fast Fourier Transform*, Prentice-Hall, Englewood Cliffs, NJ, 1974

A Sampling At the Nyquist Frequency

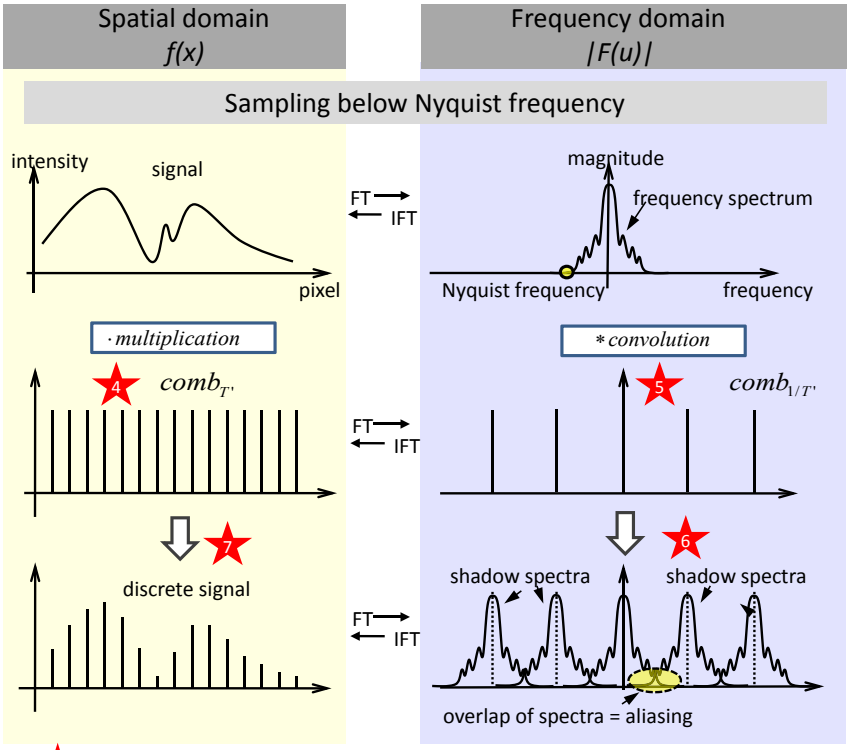


1 Sampling is multiplication with *comb* function in spatial domain

2 In frequency domain this corresponds to a convolution with the related *comb* function $comb_{1/T}$

3 Convolution result are replicated frequency spectra

B Sampling Below the Nyquist Frequency



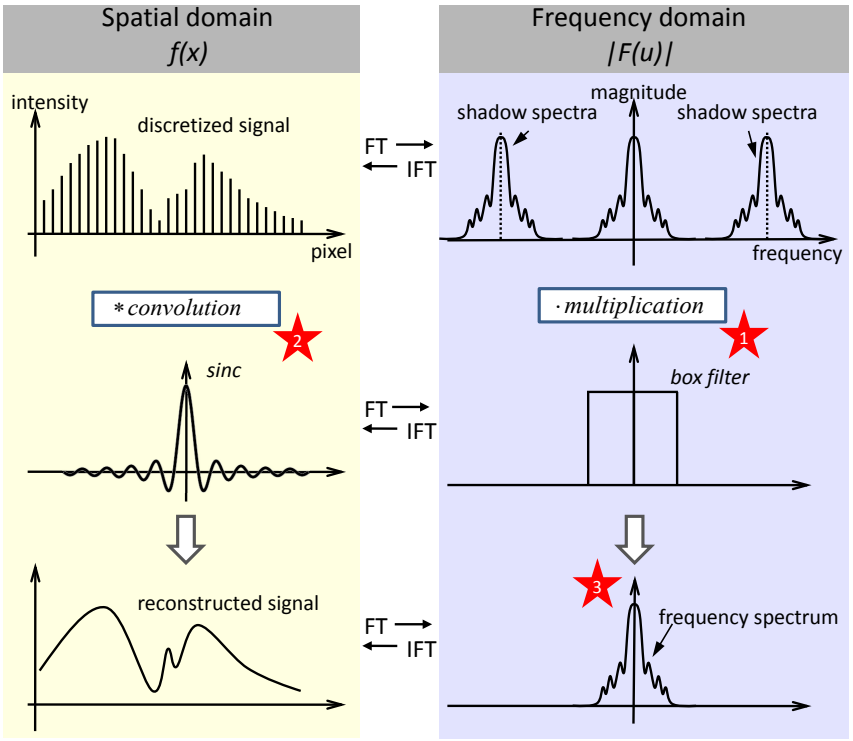
4 Sampling below Nyquist frequency means teeth of $comb_{T'}$ are too far apart

5 Teeth of $comb_{1/T'}$ are too close together

6 Resulting shadow spectra overlap → aliasing results

7 Aliased discrete signal: error free reconstruction not possible

C Reconstruction



1 Reconstruction means: eliminate shadow spectra, retain original spectrum → multiplication with box filter

2 In spatial domain this corresponds to a convolution with the *sinc* filter (ideal reconstruction)

3 Original spectrum is recovered

Non-ideal reconstruction (i.e., using not the *sinc* in spatial domain and the box filter in frequency domain) causes two types of aliasing in the reconstruction

- original spectrum not exactly recovered (e. g., high frequencies are dampened)
- shadow spectra not fully eliminated (erroneous high frequency contributions to the result)