# Uniform Sampling and Reconstruction

January 21, 2000

## 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 things 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 Transformation. The

basics of Fourier Transformation will be discussed and we will see how Fourier Transformation helps us doing sampling and reconstruction.

The second part of this paper mainly deals with reconstruction. As Fourier Transformation and the Convolution Theorem is already explained we will mainly concentrate on special techniques to improve our results like super-sampling.

### 1.1 Signals and Functions

The main thing we deal with inhere are signals represented by functions. These functions are mainly continuous as are most signals which come from real world. The functions we get out of these real world signals are defined by values at certain positions. Functions normally are infinite.

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 and
2. they are infinite.

The problems we deal with in here are quite old: Since the digital world was invented 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 invention of telegraph system some hundred years ago. At that time Morse code was used to encode messages in a reasonable way. The system had quite big advantages: it was less expensive than any system before. Thinking about drums, riders, anything used before. And the advantages of digitizing things are still the same: 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 quite good chance to influence which information will be lost in order to keep the information itself use-able.

Taking things back to computer graphics a common example might be drawing a line in a screen matrix or processing a monitor scan-line and things like that. 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 displaying and computation possible. Maybe this is the right place to state that 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 an incredible 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 interval - the reason why this is called Uniform Sampling - and take samples there. This sounds quite easy but it is not. Just taking samples very often

leads to big errors in results just because we leave out big 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 have a look at an example.

## 2.1 The basic sampling process

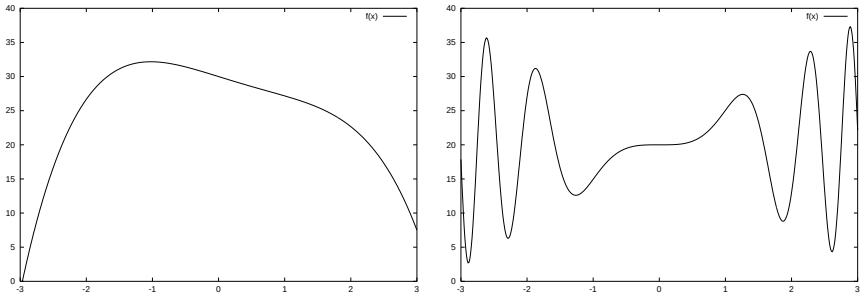
First we introduce two functions we would like to have a closer look to:

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

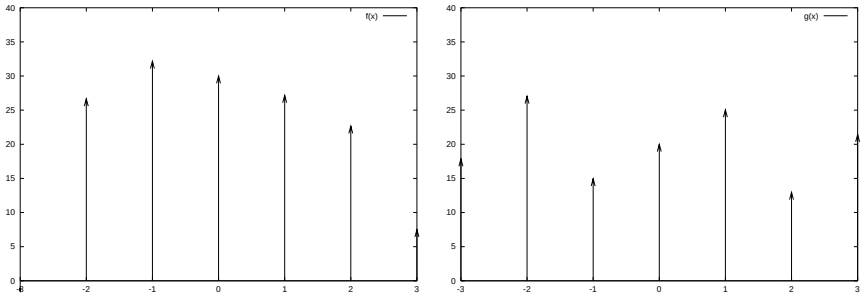
and

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

There is nothing strange about these two functions and we are not going too deeply into mathematical details. Having a look at the plotted functions allows us to see that $f(x)$ varies less over time than $g(x)$ does. The variation of a function is named its frequency. Functions may consist of many different frequencies. Going back to the above mentioned difference between $f(x)$ and $g(x)$, this is - as we will see - a quite important fact because it determines our sampling frequency. We will get back to this later.




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 2.




Now we have two sampled functions from which we may guess the original function. If we do so with $f(x)$ we can at least very easily imagine that we might come quite close to the original function whereas doing so with $g(x)$ results in a completely different function. The reason for this - not so difficult to guess

- is that we took too few samples. In other words we have to increase the sampling density for being able to better reconstruct this function. This raises the question for the best sampling frequency. Generally speaking it turned out that sampling works best with sampling at two times the highest frequency of the function. This is called Nyquist Frequency $F_{N(f)} = 2u$. We will hear about this later when talking about reconstruction.

### 2.1.1 Band limited Functions

Thinking about this we can state that a function sampled at her Nyquist Frequency is good reconstructable. Things might get worse if there is a small part within this function which has a very high frequency and the rest of this function is in a quite normal range. If this happens we are in the worst possible position: We can either do a non-uniform sampling which means we do not take our samples on a regular base or we sample the whole function at this very high frequency which could mean an incredible amount of useless data. To solve this problem for uniform sampling we normally wish to get a function which does not contain too high frequencies. To take care of this, normally some kind of low-pass filter is applied to the function with the purpose to band limit our function or in other words just to limit frequencies.

A band limited function is a function which has

1. its frequencies in the interval $[-u; u]$ where
2. $u$ is the highest frequency within the function.

Too low frequencies may also cause problems at reconstruction time. Going back to the problems with $g(x)$ the phenomenon we are confronted with is called aliasing. Basically this means that proper reconstruction is not possible any more. As Jim Blinn stated in his papers about sampling and reconstruction sampling is an information destroying process.

### 2.1.2 Aliasing

Aliasing happens because of the reduction of continuous information to periodic samples trying to provide the same information. A very common example for this are movies. Continuous movement 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 back. Its exactly the same thing happening. The reason for this is the too low frame rate in movies. A proper solution would be to increase frame rate. In practice nobody would do that because the main information of the result is not directly influenced. There are also possibilities where this is not the case.

To sum it up at sampling time we have to care for sampling frequency which should be 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 Low-Pass filter

We already stated above that a filter to cut off too high and too low frequencies should be applied. We already introduced the term Low-Pass filter for this. Now we will go more into detail: The interesting question is what a Low-Pass filter does and how we can apply it to our function.

### 2.2.1 Convolution and Convolution Theorem

To filter one function with another function we can use Convolution. The convolved function is a sliding weighted average of one function using the other for providing the weights. That is exactly what we need for merging our functions an average using both functions.

The interesting thing about Convolution is that if two functions in spatial domain are convoluted their representation in frequency domain is multiplied and vice versa. This is stated by the Convolution theorem. So practical use for this is quite obvious. Seen from the point of view of performance, the transforming process and the multiplication there is much cheaper than doing a real convolution in spatial domain. This is true in 2D for a filter kernel from size 15x15 and upwards. Below this border costs of computation are low anyway.

### 2.2.2 Fourier Transformation

It is possible to do transformation of our functions from spatial domain to frequency domain where they are not aperiodic and continuous as they mostly are in spatial domain but a sum of sine and cosine waves. This helps us a lot with computing these functions. This transformation process can easily be done with Fourier Transformation which is also used for restoring the original signal in spatial domain. Important to notice is that Fourier Transformation - short FT - does not harm our functions in any way. They are just seen from a different point of view.

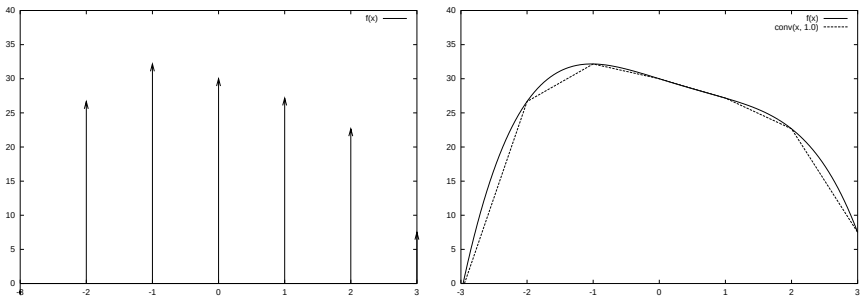
So FT is just used as a tool to switch from spatial to frequency domain and vice versa. We do not necessarily need FT to do this job. There also exists Hartley transform for example which could do exactly the same for us. FT is just more commonly used.

# 3 Reconstruction

After sampling we normally have sampled functions represented by values at certain positions. At uniform sampling these certain positions are in regular intervals. To make our function useable again we have to reconstruct it again. Basically this means transform the digitized signal back into an analog one.

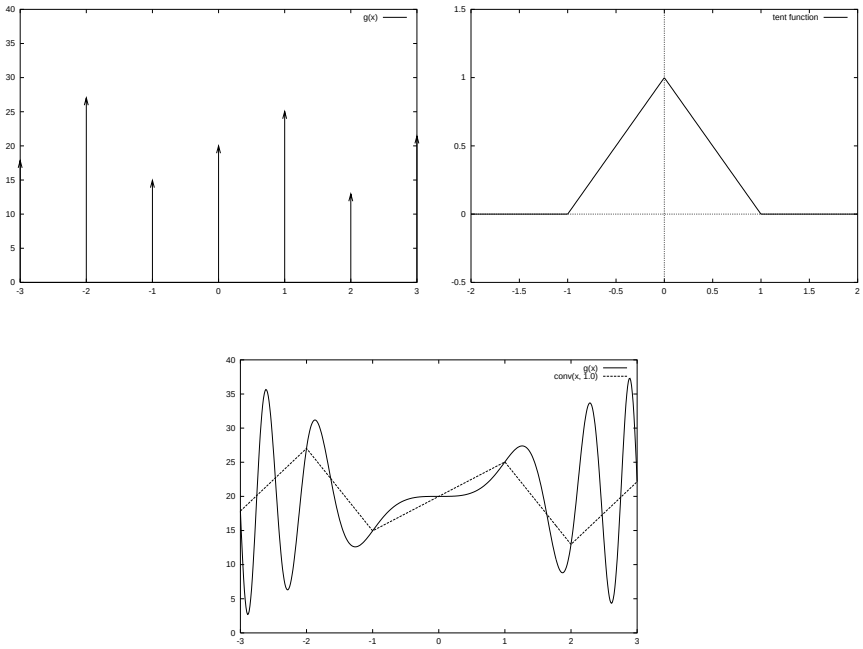
Thinking back to high school times whenever we had to draw a function in maths we first calculated some values and then tried to draw the function. This is exactly the same process. First sampling, computing some values at certain positions mostly on a regular interval then draw the points and connect them. So can we do here:

## 3.1 Linear Interpolation




This method is called Linear Interpolation. The original function is also displayed just to make life easier. The result looks quite promising. The exact way of Linear Interpolation is just to apply a so called Tent filter to our samples. The Tent and the samples are convolved and Linear Interpolation in between the values is done. The Tent filter has a value of 1 at zero and is going down straight forward to zero at position 1 and -1.

Convolving the second function $g(x)$ with the Tent filter results in the following:




Here we have a really rough approximation compared to the above one. The reason for this is the too low sampling frequency. The side effects are called aliases.

As our sampling distance was 1 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

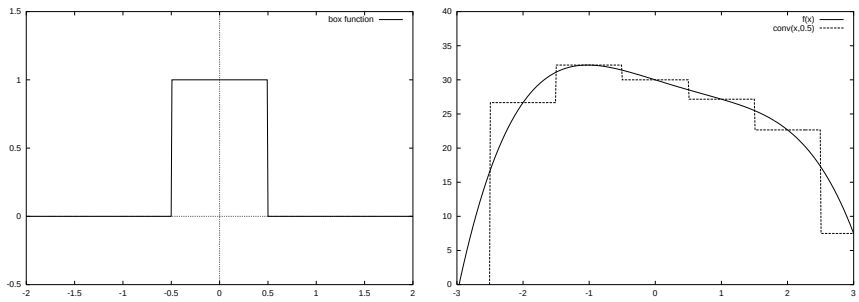
be achieved analytically or by transferring both functions to frequency domain multiplying them and transferring the result back.

As the results are quite good for the properly sampled function $g(x)$ we will now focus on improving our results. Obviously the result depends on the reconstruction filter therefore we have to think about improving it. Other possibilities are nearly all functions that have a value of 1 at position zero and go down to zero on both sides. The function should also be symmetric about the origin.

## 3.2 Nearest Neighbor

Maybe the easiest way of reconstructing a function from its samples is doing it as if you would do it with Lego pieces: You just have the chance of building one height for a certain distance. The result looks like a stair-step function.

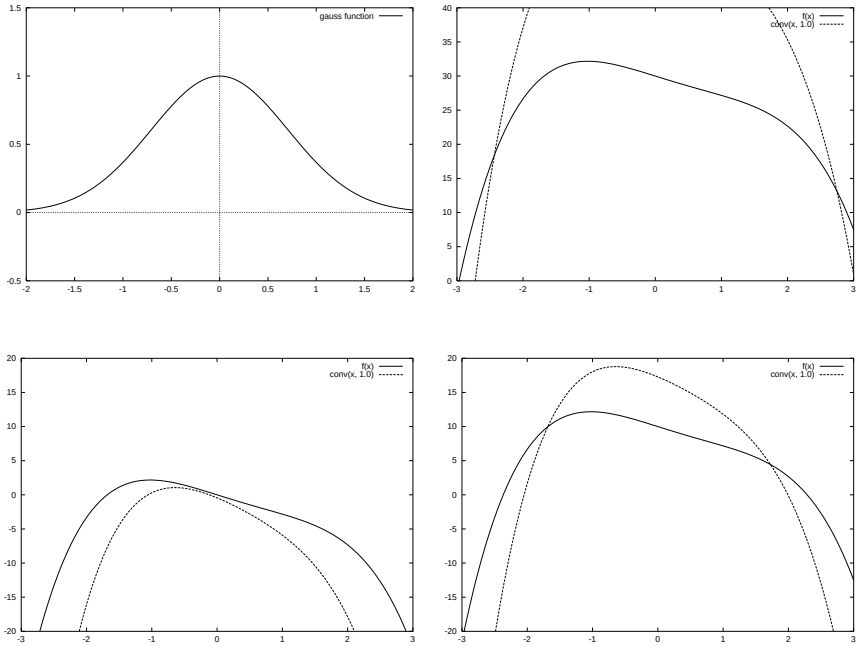
To imagine what happens there: The area around one sampling point on the x-axis has half of the sampling distance on both sides the value of this sampling point. It looks like the following:




The filter used for this is called Box filter. It is 1 for the half sampling distance left and right of the origin and zero elsewhere. Maybe this is a good place to mention that a 1 as y-axis value means 100% influence on the reconstructed value and zero means no influence at all. Why a reconstruction filter should not be larger than on the x-axis than one sampling distance can easily be seen with the Gauss function used as a filter.

## 3.3 Approximation

The Gauss function converges in infinity to zero. So it is not completely zero outside half sampling distance around the origin. We will see what this does to our sampled function. In this example I will shift the function on the y-axis just to show what the non-zero values outside half the sampling distance do to our results.




The only difference between the pictures is a shift of the original function on y-axis. For the above mentioned reasons the Gauss function can never be used as a good reconstruction filter just because the only thing it does is an approximation.

But there is still the question for a perfect reconstruction filter.

## 3.4 The 1D Sampling Theorem

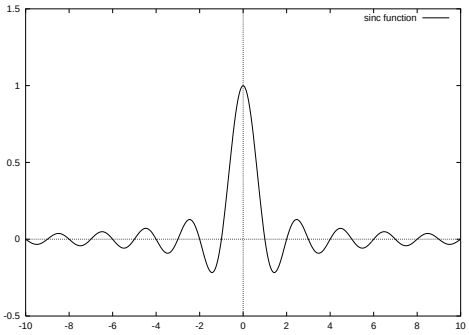
Theoretically a function can perfectly reconstructed by using the sinc function.

$sinc = \frac{\sin(\pi x)}{\pi x}$

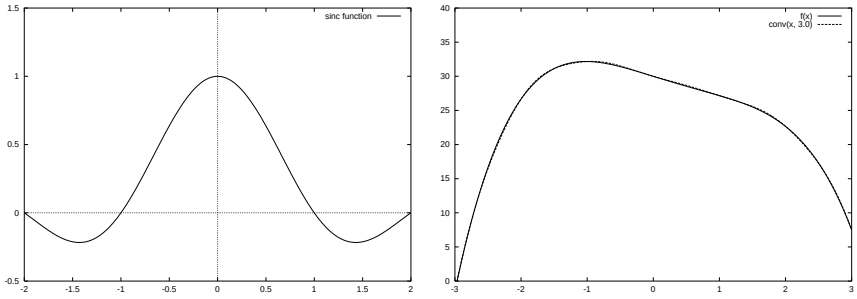
If we have a band limited function sampled with its Nyquist frequency we can perfectly restore it by using the sinc filter in the following way:

$reconstruction(x) = sinc\left[\frac{u}{2\pi}(x - nX)\right]$

$X$ specifies the sampling distance and $u > 2\pi$. The main problem with the usage of the sinc filter is that it is infinite. Just cutting the edges leads to incredible reconstruction errors. So this might not be the best idea. One possible technique is the usage of a window. This basically means just to use a part of the sinc filter at the most relevant passage around the origin. The sinc filter is also very good just because it is at every sampling position zero except the one in the origin.




We will just have a short look at an example for this windowing technique in combination with the sinc filter: We use a Kaiser window with window width 3 and the Theussl constant $t = 9.28$. The result looks nearly perfect, only at some parts we may notice the difference between the original function and the reconstruction.




Up to now we found 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. Only difference is that every dimension has its own bandwidth and Nyquist frequency. So we have 2 or 3 different sampling distances.

## 3.5 2D Sampling Theorem

We can think about the result of 2D sampling as a not necessarily symmetric grid and each point of interception as one pair of $(x/y)$ samples. As we can now very easily imagine we have two different sampling rates and therefore different sampling distances. For stating something like the 1D Sampling Theorem we just have to care for one more different function to be reconstructed. But the definition of the 2D sampling theorem is straight forward:

$reconstruction(x, y) = sinc\left[\frac{u_x}{2\pi}(x - X)\right] sinc\left[\frac{u_y}{2\pi}(y - Y)\right]$

$u_x$ and $u_y$ are band widths in both dimensions as $X$ and $Y$ are the different sampling intervals. As soon as we work in 2 dimensions we can easily find out that a function, once sampled with the wrong distance may start overlapping at reconstruction time and therefore be irrecoverable lost. "Digital Image Synthe-

sis" by Andrew S. Glassner has some nice examples for this. Basically its the same aliasing effect as seen at the first effort on reconstructing $g(x)$.

### 3.6 Further Improvements

We will not always be in a position to reconstruct with perfect filters, therefore we will never get perfect results out of our work. But there are some "Low-cost tricks" like super sampling which make life easier.

First at reconstruction time we never ever completely reconstruct a function. What we do is we reconstruct only the necessary parts of a function. This means whenever we want create a picture out of our 2D samples we treat the y value for example as a color value and are able to reconstruct the function only at the points we essentially need for our reconstructed picture because this is the only way it makes sense.

Super-sampling works the same way but sample positions are taken for example 4 times for one pixel and then an average is computed. This is used to improve quality. Also in 3D reconstruction this is an excellent way of improving quality. The essential factors for this kind of image processing are time and costs.

## 4 Summary

As we have seen, sampling an image is a very critical process. Any 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 perfectly be restored by using the perfect, infinite sinc filter. Reconstruction errors can be reduced to a minimum by super-sampling.

Still the question of cost of computing versus quality remains.