Chapter 1

Sampling

Sampling is the process of transforming a continuous signal into a discrete one. In nature, most of the signals we measure (sound, light, ...) are defined over continuous domains (time, space, ...). In order to process them with computers we need to transform the continuous domain into a discrete one. This process is called sampling.

We need to study the following questions: what are the possible sampling patterns to discretize a signal? how can we characterize the lost of information? and how do we reduce artifacts?

Let's consider a 1D continuous time signal $f(t)$ and its sampled version $f[n] = f(nT_s)$, were T_s is the sampling period. Intuitively, it is clear that in this sampling process some information will get lost. If no information was lost, then we should be able to recover the continuous signal $f(t)$ from its sampled version $f[n]$ by doing some kind of interpolation. One could simply decrease T_s, which will results in a more accurate approximation of the continuous signal $f(t)$ at the expense of the amount of memory needed to store $f[n]$. Decreasing T_s will also result in an increase of the computational cost of processing the signal $f[n]$. Therefore, it is interesting choosing the appropriate T_s. Understanding the sampling process and how to reconstruct the continuous signal is important as it will allow us to find the optimal sampling parameters.

1.1 Sampling theorem

Let's first look at one example to get a sense of the type of issues that might arise when discretizing a signal. Figure 1.1.a shows one continuous signal with the form $f(t) = \cos(wt)$ with $w = 18\pi$. The period of this signal is $T = 1/9$ (there are 9 periods in the interval $t \in [0, 1]$). We now build a discrete signal $f[n] = f(nT_s)$ with $T_s = 1/11$ (there are 11 samples in the same interval $t \in [0, 1]$). This could seem enough because there are more samples than periods.

Figure 1.1.b shows $f[n]$. If we now want to reconstruct the original continuous signal from its samples $f[n]$ there are many possibilities as the samples do not constraint what happens between samples. Therefore we will need to make some assumptions about the continuous signal. In the absence of any other prior information, we will assume that the most likely signal is the slowest and smoothest signal (we will make this assumption more precise later). Figure 1.1.c shows the superposition of the original signal and the reconstructed one. Both signals perfectly pass through the same samples. Clearly the samples seem to correspond to a cosine function with a lower frequency (in this example $T = 1/2$) than the input (which had $T = 1/9$).

It is important to mention that there is nothing special on how the parameters have been chosen for this example. Many different parameter choices would have yielded the same qualitative behavior. This confusion of frequencies is called *aliasing*. We will show that for the reconstruction to match the input we need $T_s < 1/(2T)$. The *sampling theorem*

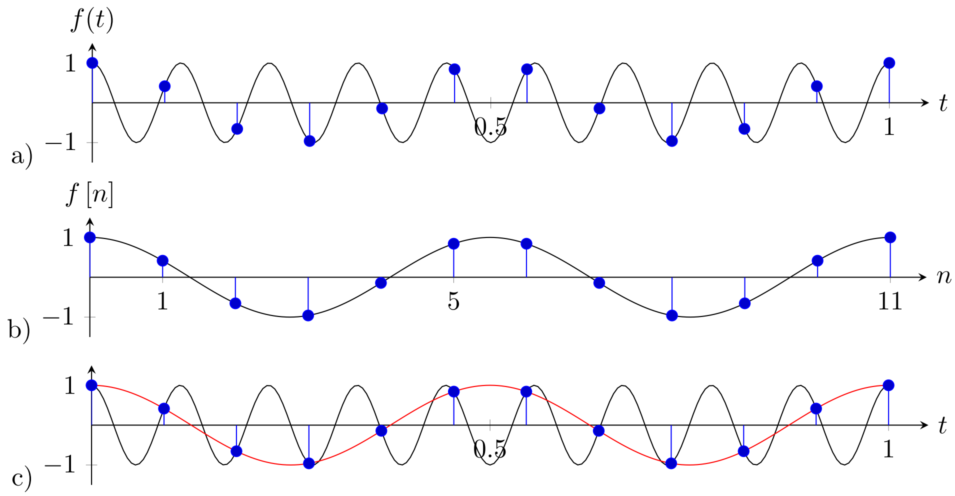

Figure 1.1: a) Continuous signal and its samples. b) Discrete signal and the reconstructed continuous signal by interpolation. c) Superposition of continuous signal (a) and its reconstructed approximation from the discrete samples from (b).

(also known as Nyquist theorem) states that for a signal to be perfectly reconstructed from it samples, the sampling frequency $f_s = 1/T_s$ has to be $f_s > 2f_{max}$ when f_{max} is the maximum frequency present in the input signal. You can check that our previous example did not satisfied the Nyquist condition.

One way of characterizing the sampling process is achieved by analyzing the relationship between the Fourier transform of the continuous and discrete signals. There are many ways of finding the relationship between the two Fourier transforms. Here we will describe the most common one.

Let's start writing a model of the sampling process by defining a special signal:

$$\hat{f}(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} f(nT_s)\delta(t - nT_s) = f(t)\delta_{T_s}(t) \tag{1.1}$$

where $\hat{f}(t)$ is a very special function that contains the same amount of information as the discrete signal $f[n]$ but that is defined over the continuous domain t. Note that $\hat{f}(t)$ is the product of two continuous signals. The first term is the continuous signal $f(t)$, the second term is a function composed of impulses placed at regular time instants $\delta(t - nT_s)$. The use of impulses is interesting because they are infinitely narrow in time, so the product of an impulse with a function is equivalent to taking just one sample of that function. Remember from the definition of $\delta(t)$ that $f(t)\delta(t - nT_s) = f(nT_s)\delta(t - nT_s)$. We define the impulse train (also called Dirac comb), $\delta_{T_s}(t)$, as the signal:

$$\delta_{T_s}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{1.2}$$

Although we will never directly work with the signal $\hat{f}(t)$, it is a convenient construction to understand how information is transformed during the sampling process. To see the interest of this construction, let's compute its Fourier transform. The continuous Fourier transform of $\hat{f}$ can be written as the convolution of the Fourier transforms of $f(t)$ and $\delta_{T_s}(t)$.

The Fourier transform of a Delta comb is:

$$\begin{aligned}\Delta_{T_s}(w) &= \int_{-\infty}^{\infty} \delta_{T_s}(t) \exp(-jwt)\, dw & (1.3)\\ &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t - nT_s) \exp(-jwt)\, dw & (1.4)\\ &= \sum_{n=-\infty}^{\infty} \exp(-jwnT_s) & (1.5)\\ &= \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta\left(w - k\frac{2\pi}{T_s}\right) & (1.6)\end{aligned}$$

It is honest to admit that the last step in this derivation is far from trivial. The Fourier transform of an impulse train is also an impulse train but with an displacement in frequency between impulses that grows when the spacing in time decreases.

Therefore, the continuous Fourier transform of $\hat{f}$ can be written as:

$$\widehat{F}(w) = F(w) \circ \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta\left(w - k\frac{2\pi}{T_s}\right) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} F\left(w - k\frac{2\pi}{T_s}\right) \tag{1.7}$$

where $F(w)$ is the Fourier transform of $f(t)$. This equation shows that $\widehat{F}(w)$ is build as an infinite sum of translated copies of $F(w)$. Each copies is centered on $k\frac{2\pi}{T_s}$. If T_s is small (i.e. if we sample very fast) then those copies will be far away from each other. But if we have few samples and T_s is large, those copies will get very close and will start mixing with each other. High frequency content in $F(w)$ will affect the low frequency content of $\widehat{F}(w)$, and this is exactly what produces aliasing. Figure 1.2 illustrates this. In this example, there is one band limited signal (i.e., there is a frequency, w_{max}, for which the magnitude of the Fourier transform is zero for all frequencies above w_{max}). First $T_s = 4$ seconds, in its FT we see replicates of the $F(w)$ centered around $\pi/2$. With $T_s = 8$ seconds, the replicates appear centered around $\pi/4$ and they start touching. $T_s = 8$ is slightly above the Nyquist's limit and some aliasing will exist. For $T_s = 16$ aliasing is severe and information will be lost making it impossible (without any additional prior information) to reconstruct the continuous function from its samples.

1.2 Reconstruction

If the copies do not touch, then we can see how it is possible to reconstruct the original continuous signal. We just need to apply a filter that has a constant gain for all the frequencies inside $w \in [-w_{max}, w_{max}]$, and 0 outside. The phase of the filter should be zero. This is:

$$H(w) = \begin{cases} \frac{T_s}{2\pi} & \text{if } w \in [-w_{max}, w_{max}] \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

One piece of bad news: for any time limited signal (i.e., a signal that is defined inside an interval $t \in [a, b]$ and it is zero outside) the Fourier transform is not band limited. In other words, a signal can not be simultaneously time limited and band limited. Anyway, when something is impossible, generally it is because it does not matter and it might just mean that it is not the right way of thinking about the problem. So let's not worry about it.

The impulse response of such a filter is:

$$h(t) = \frac{\sin(t)}{t} = sinc(t) \tag{1.9}$$

it is called the sinc function.

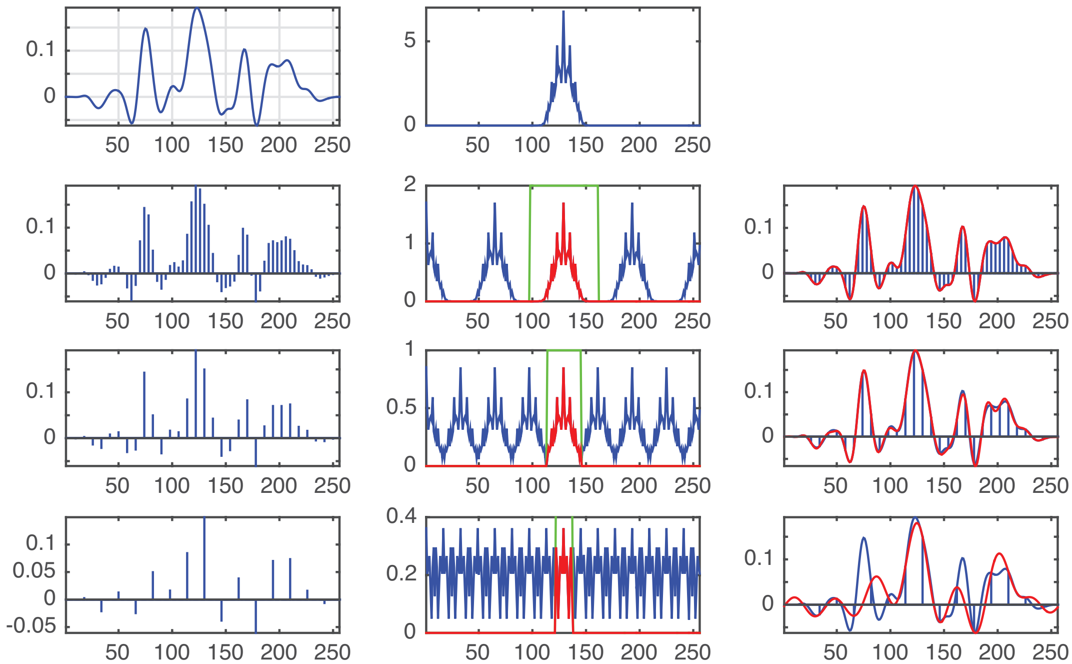

Figure 1.2: Aliasing examples. (a) - (f) Far left column: spatial sampling pattern. 2nd column: Fourier transform of that spatial pattern, revealing replication locations of the Fourier transform spectrum of the subsampled image. The subsampled image is shown in the 3rd column. Zeroing out all but the central replication of the image spectrum (far right), yields the interpolated images of the 4th column.

In fact, it is easy to show that, in the lack of any other prior information, this is the optimal reconstruction in terms of the L2 norm. This is:

$$sinc(t) = argmin_h \int \left(f(t) - \hat{f}(t) \circ h(t)\right)^2 dt = argmin_H \int \left(F(w) - \widehat{F(w)} H(w)\right)^2 dw \tag{1.10}$$

then the function, $\widetilde{f}(t)$, that better reconstructs the input signal from its samples is:

$$\widetilde{f}(t) = \hat{f}(t) \circ sinc(t) = \sum_{n=-\infty}^{\infty} f\left[nT_s\right] sinc\left(\frac{t - nT_s}{T_s}\right) \tag{1.11}$$

where $\widetilde{f}(t)$ is the reconstructed signal and $\hat{f}(t)$ is the sampled signal. One disadvantage of this reconstruction is that the *sinc* function has infinite support which means that to interpolate each instant, we need to linearly combine all the samples $f\left[nT_s\right]$. Sometimes it is better to have a local reconstruction that only depends on the nearby samples. Indeed, there are other possible reconstructions that are not optimal in terms of L2 norm, but that only require local computations: linear, bilinear, bicubic, splines, etc. All of them can be written as a linear convolution with a kernel $h(t)$. In the case of the linear interpolation, the kernel $h(t)$ is a triangle of width $2T_s$.

1.3 2D spatial sampling

Let's now analyze what happens when sampling 2D signals to form discrete images.

In 2D things get more interesting. If we have a continuous image $f(x, y)$ we can sample it using a rectangular grid as $f\left[n, m\right] = f\left(nT_x, mT_y\right)$. We can do a very similar analysis to the

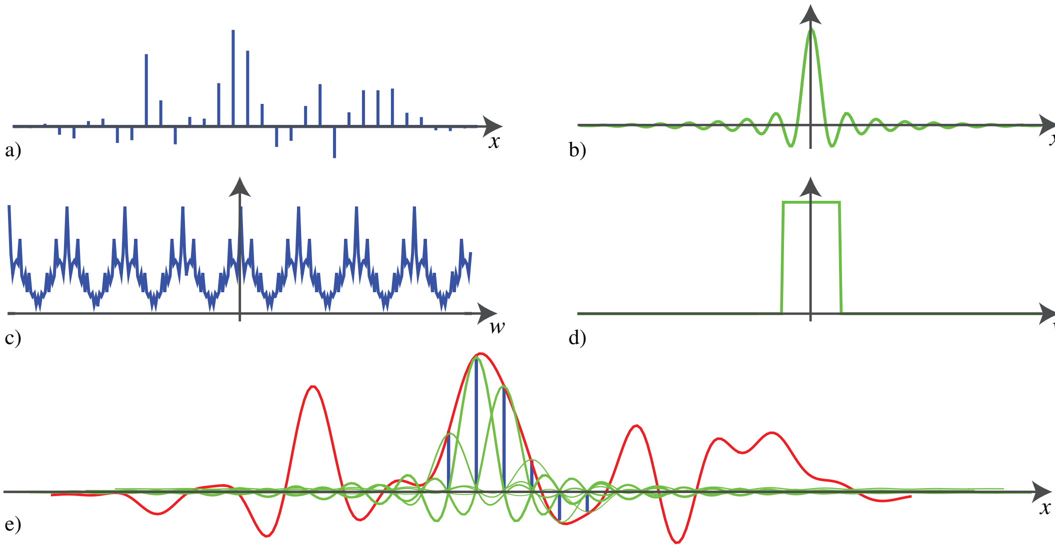

Figure 1.3: Reconstruction. a) Signal multiplied by a delta train. Each line corresponds to one impulse. The height of each impulse corresponds to the value of its integral. b) sinc function. c) magnitude of the Fourier transform of (a). d) Fourier transform of (b). The width of the box is set to cover just the central repetition of the FT shown in (c). e) Illustration of the reconstruction process. The sinc functions are scaled and shifted on top of each sample and then summed up (only six are shown). Note how the zero crossings coincide with the sample locations. The sum of all the sinc function corresponds to the red curve.

one we just did for the 1D case. But in 2D we can have more interesting sampling patterns. For instance, we could define the discrete image as:

$$f\left[n,m\right] = f\left(an + bm, cn + dm\right) \tag{1.12}$$

where a, b, c, d are constants. For instance, if $a = T, b = 0, c = 0, d = T$ then we will have a regular rectangular sampling. But we could have other patterns. For instance, if we set $a = T_1, b = -T_2/2, c = 0, d = T_2$ then we obtain an hexagonal sampling. So, now we can ask the following question: what is the optimal 2D sample arrangement given a fixed number of samples? The answer will require studying how aliasing will happen. What we want is to chose the sample arrangement that will allow the best reconstruction of the input continuous signal from a fixed number of samples. As we did with the 1D case, we can address this by studying the relationship between the Fourier transform of the continuous signal and the sampled one.

$$\widehat{f}(x,y) = f\left(x,y\right) \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta\left(x - an - bm, y - cn - dm\right) \tag{1.13}$$

Where the 2D delta train can be written using vector notation for the continuous spatial coordinates:

$$\delta_A(\vec{x}) = \sum_{\vec{n}\in\mathbb{Z}^2} \delta\left(\vec{x} - A\vec{n}\right) \tag{1.14}$$

where $\vec{x} = (x, y)^T$, $\vec{n} = (n, m)^T$, and A is the matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{1.15}$$

The continuous Fourier transform of this delta train can be done by applying a change in variables and then using a similar procedure as the one followed in the 1D case. The result is:

$$\Delta_A(\vec{w}) = \frac{(2\pi)^2}{|A|} \sum_{\vec{k} \in \mathbb{Z}^2} \delta\left(\vec{w} - 2\pi A^{-1}\vec{k}\right) \tag{1.16}$$

Therefore, the Fourier transform of the sampled signal $\widehat{f}(x, y)$ is:

$$\widehat{F}(\vec{w}) = \frac{(2\pi)^2}{|A|} \sum_{\vec{k} \in \mathbb{Z}^2} F\left(\vec{w} - 2\pi A^{-1}\vec{k}\right) \tag{1.17}$$

Remember that for 2×2 matrices the inverse is easy to write:

$$A^{-1} = \frac{1}{|A|} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{1.18}$$

We can now check what happens with different sampling strategies. For the 2D rectangular sampling, eq. 1.17, simplifies to:

$$\widehat{F}(w_x, w_y) = \frac{(2\pi)^2}{T^2} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} F\left(w_x - \frac{2\pi}{T}k_1, w_y - \frac{2\pi}{T}k_2\right) \tag{1.19}$$

This is similar to the 1D case. Figure 1.4 shows two different delta trains for two different sampling patterns and also their Fourier transforms. The region delimited by the green polygon shows the region of valid frequencies. If the input signal only has spectral content within that region, then there will be no aliasing. The optimal sampling will be the one that makes that region as large as possible for a fixed number of samples. The optimal sampling strategy is the regular hexagonal sampling. This is not the sampling used in computer vision as all images are always represented on a rectangular grid, but an hexagonal sampling achieves an increase of around 10% in resolution for the same amount of samples. In fact, the distribution of photoreceptors in the eye [**?**] are distributed on an hexagonal array as shown in figure 1.5. Working with convolutional filters defined over an hexagonal grid is more efficient and it can achieve better radial symmetry [**?**].

For all the examples and derivations in this book, we will be working always of a regular rectangular grid.

1.4 Aliasing and anti-aliasing filter

Sampling with the wrong frequency has interesting effects in 2D. Figure 1.6.a shows an example of a picture downsampled at different resolutions (412×512, 103×128, 52×64, and 26×32) and then reconstructed to the original resolution (412×512 pixels). For the figures, as we do not have access to the continuous image, we always work with sampled versions. But the original image is very high resolution and we can think of it as being the continuous image.

The images in Figure 1.6.a show the effects of aliasing. The stripes in the Zebra's body change orientation as we down sample them. And for the lowest resolution image, it is even hard to recognize the animal as being a zebra. Figure 1.6.b shows what happens with the image Fourier transform when we multiply it with the delta train (compare it with fig. 1.4). Figure 1.6.c shows the magnitude of the DFT of the sampled image (it corresponds to the region inside the green square in fig. 1.6.b). The DFT changes substantially, due to aliasing, from one resolution to the next one.

In order to reduce aliasing artifacts we need to filter the continuous signal with a low-pass filter in order to make it band-limited. Then we will be able to sample it avoiding high-spatial frequencies to interfere with the low-frequency content of the image. The anti-aliasing filter will not prevent from loosing the information contained in the high spatial frequencies. Figure 1.6.e shows the reconstructed images at different resolutions when an antialising filter is applied before sampling. Each resolution requires a different filter. The antialising filter can be a box filter like in eq. 1.8 with the support equal to the green region in Figure 1.6.b.

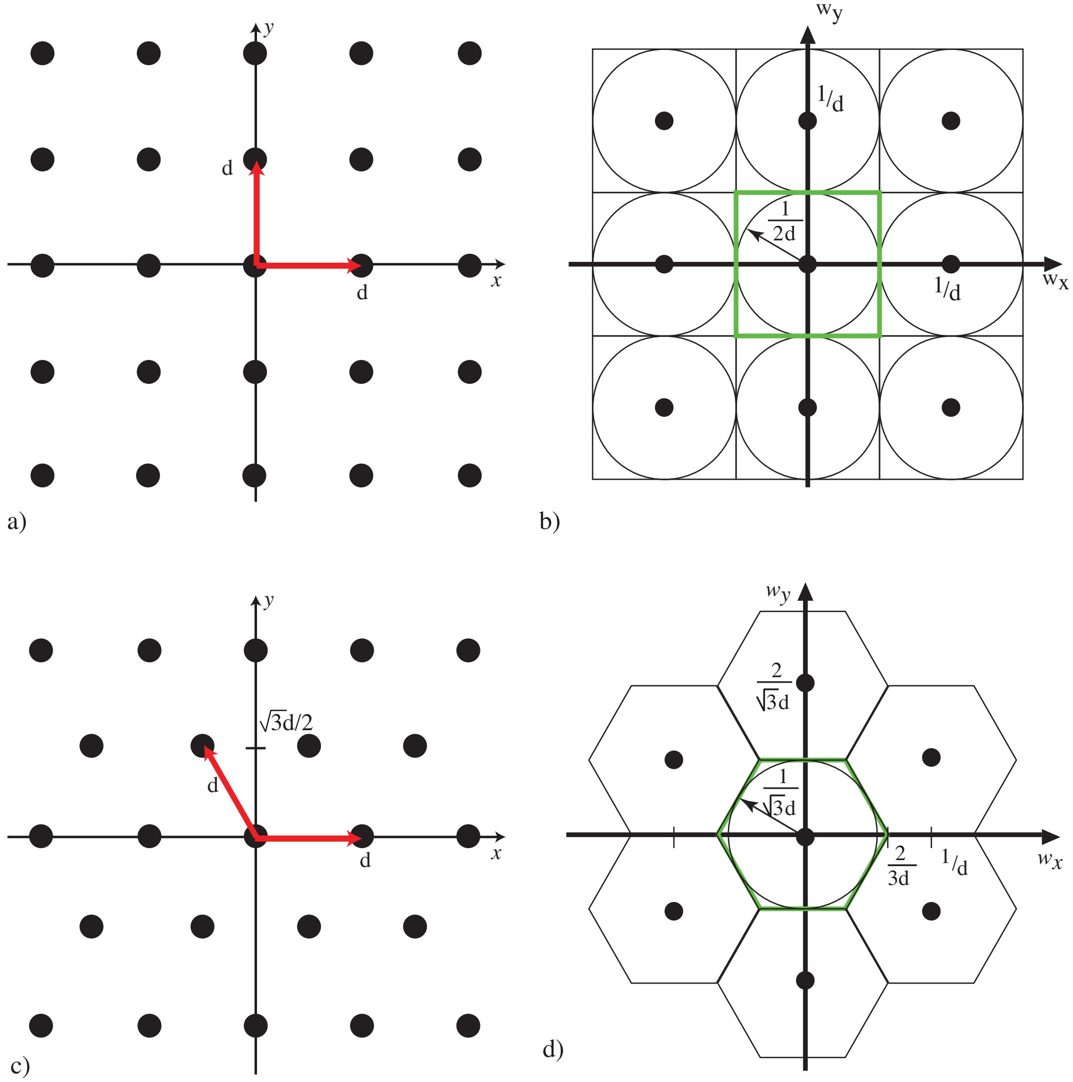

Figure 1.4: Sampling patterns.

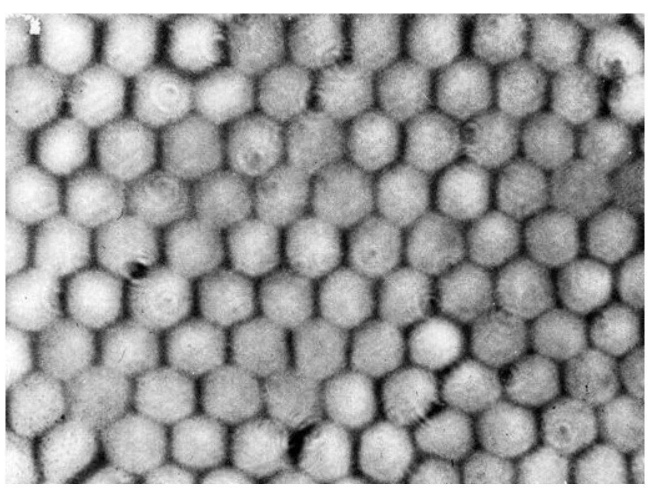

Figure 1.5: Distributions of cones in the fovea (cite source).

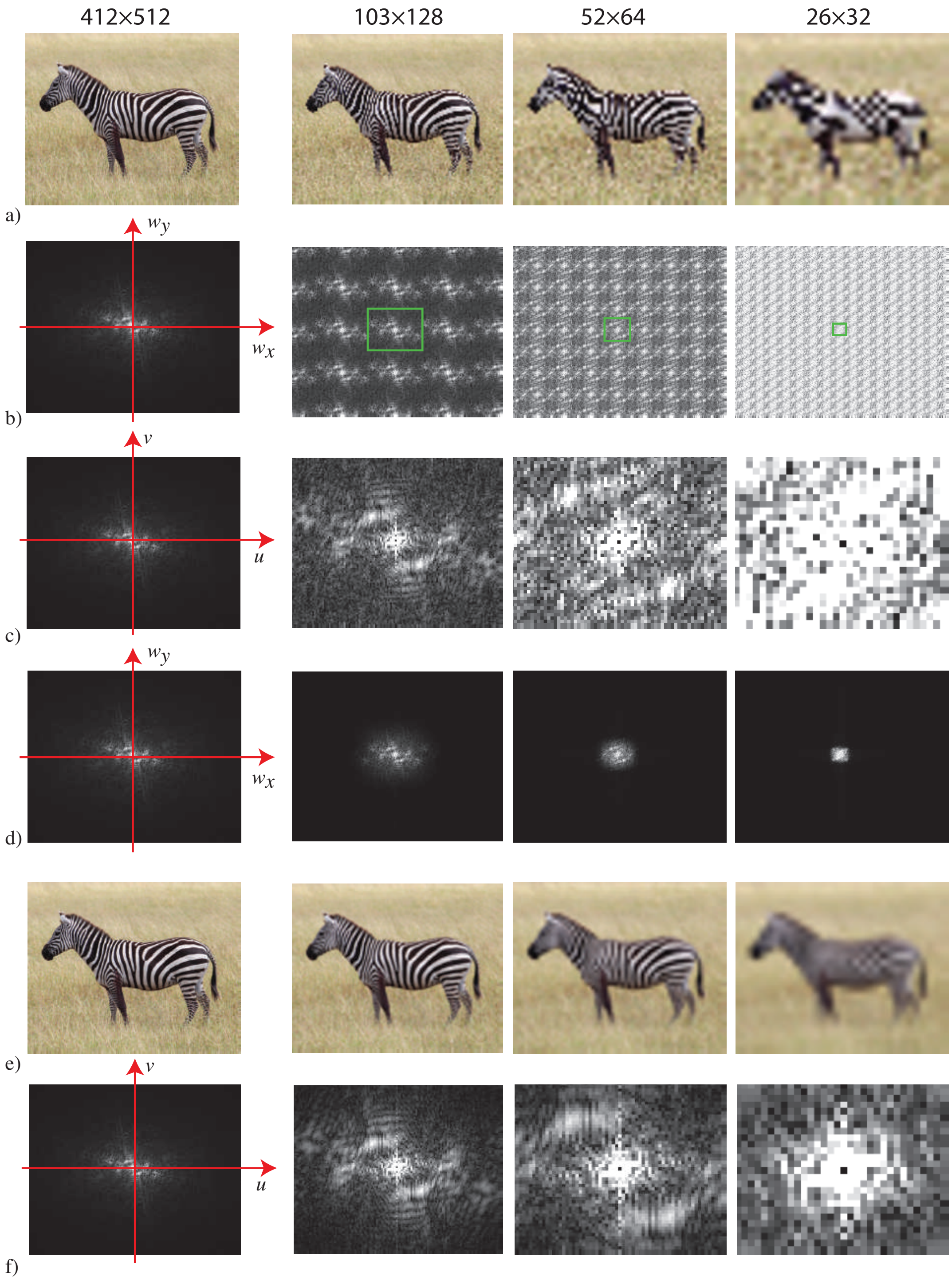

Figure 1.6: Aliasing and antialiasing filter. a) The zebra sampled with aliasing starts looking as a cow. b) Fourier transform of the continuous signal $f(x, y)$ multiplied by delta trains: $\widehat{f}(x, y)$, c) Discrete Fourier transform of the corresponding sampled signals, $f[n, m]$, and d) Fourier transform of the reconstructed signal. e) Sampled image after processing it with an antialiasing filter. f) Discrete Fourier transform of the corresponding antialiased sampled images, $f[n, m]$. Note that now the central part of the Fourier transform is not changing.