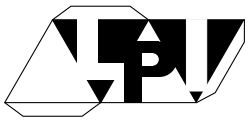
UNIVERSIDAD DE VALLADOLID

Master Universitario de Investigación
en Tecnologías de la Información y las Comunicaciones

2D SIGNAL PROCESSING



Laboratorio de
Procesado de Imagen
Santiago Aja Fernndez
E.T.S.I. Telecomunicacin
Valladolid, Febrero 2020

Image representation and 2D signal processing

Santiago Aja-Fernández[1], Gabriel Ramos-Llordén[2], and Paul A. Yushkevich[3]

[1] *Universidad de Valladolid, ETSI Telecomunicación, Valladolid, Spain*
[2] *Harvard Medical School, Psychiatry Neuroimaging Laboratory, Boston, MA*
[3] *University of Pennsylvania, Penn Image Computing and Science Laboratory, Philadelphia, PA*

1 Image representation

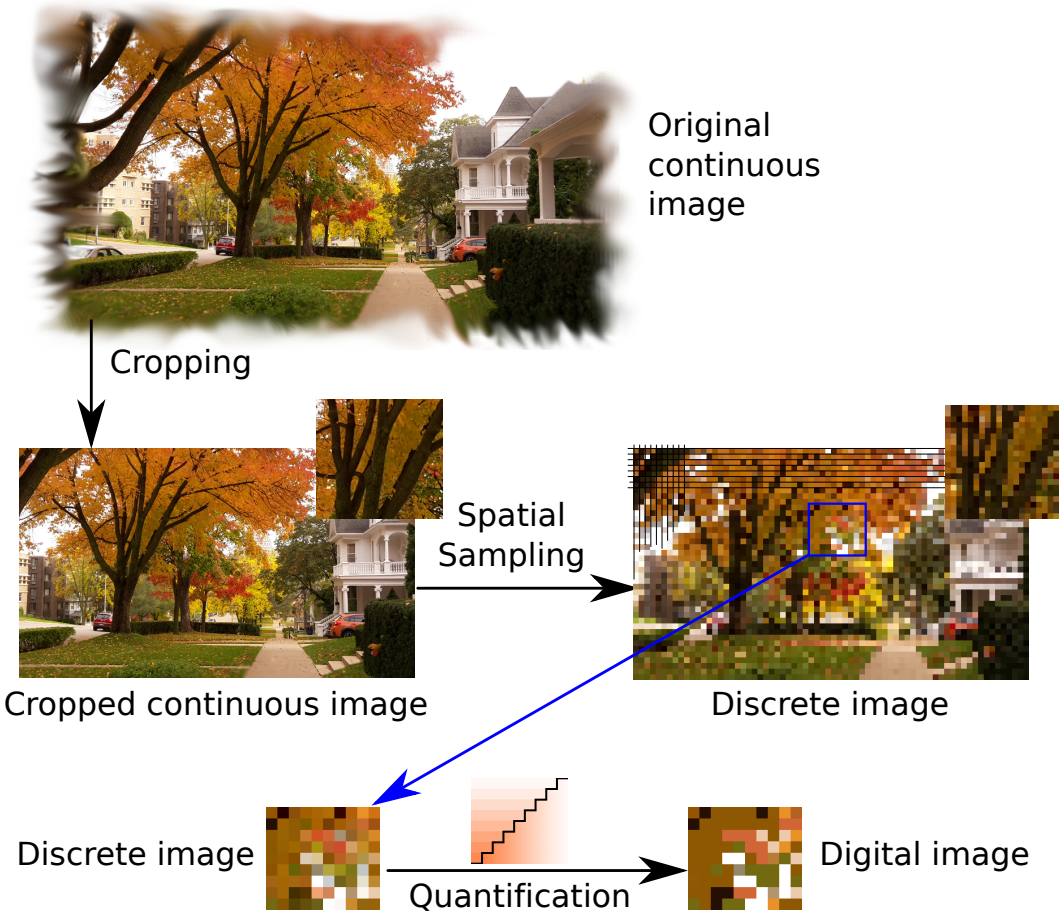


Figure 1: From continuous to digital image: the original continuous image, when acquired by a device is (1) limited in space; (2) spatially sampled and (3) quantized.

In medical imaging, when we talk about *images* we usually refer to digital images and, specifically to raster or bitmap images. A digital image is a 2D numerical representation of a picture of reality. See the example in Fig. 1: an ideal continuous image is captured by a device and it is transformed into a digital image through the following procedure:

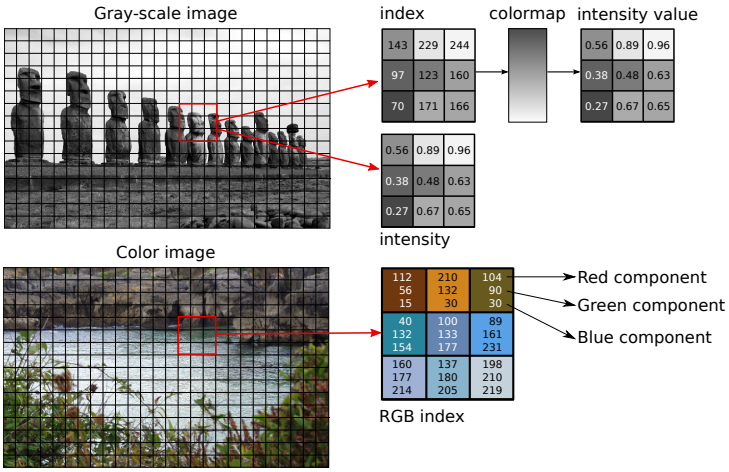


Figure 2: Digital image representation. Top: A gray scale image can store the intensity information or an index that points to the colormap with the intensity information. Bottom: Example of the representation of a color image using a RGB color component.

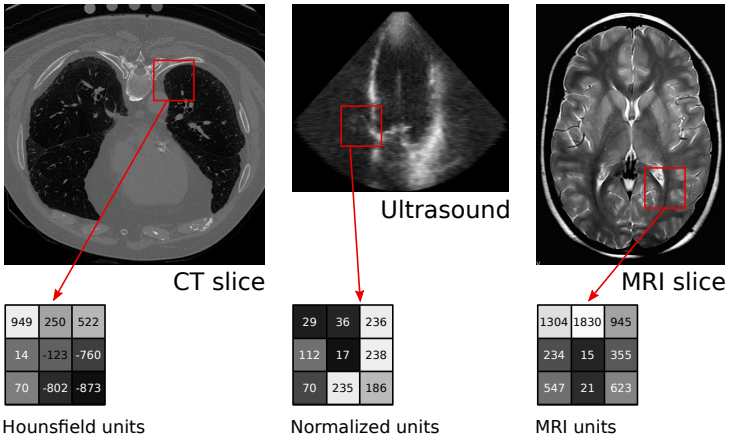


Figure 3: Example of the intensity values provided by three different modalities of medical images: Computer Tomography, ultrasound and magnetic resonance imaging.

1. The spatial extent that can be captured by the imaging device is limited.

2. The original image is sampled, in order to obtain a limited and discrete number of values. This is related with the resolution of the image. The basic unit of image, which usually corresponds to one sampled value, is known as *pixel*.

3. The range of values that a pixel can take is reduced to a discrete set. This procedure is known as quantification.

Although this procedure is describe in Fig. 1 for real-world images, it can be easily translated to the different modalities of medical imaging.

Information in bitmap images is stored inside every pixel, see Fig. 2. The value of that pixel can correspond to an intensity value, usually in the range $[0, 1]$, or it can be an index to a colormap, where the intensity values are stored. While both methods are equivalent for gray scale images, color images usually store indexed information, see Fig 2-bottom. If an image is coded using three color component (red, green, blue, for instance), a pixel stores the

indexes to the three color components. There may be images encoded with more than three comoponents. Indeed, multi-component data appear in magnetic resonance imaging, (diffusion MRI, functional MRI) as well as in microscopy. Nevertheless, in this chapter we focus on single-component images, which is yet the most traditional way to acquire images in the medical field.

In most medical imaging modalities, the information stored in a single pixel is related to the acquisition units of every specific modality, see Fig. 3 for illustration. The acquire data is usually given to the user as a digital image. However, the values from each pixel can vary: CT images provide units in the Hounsfield scale, where air areas have a value of -1000, water is 0 and bone from 200 to 3000. The scale is a linear transformation of the original attenuation coefficient measured in the scanner. It is quantitative and therefore it is totally related to the structure of the materials and tissues. In the example of the ultrasound image, in this case an ecocardiography image, the data provided by the vendor has been normalized in the range $[0 - 255]$, so that every pixel is represented by and integer number (one byte). Last, the MRI slice provided values are related to the scanner and acquisition features, so the values are not related to tissue properties. In this example, the information is given by the visual contrast between regions.

In this chapter we will consider the images as bidimensional signals, so we can use the whole signal processing theory upon them. The extension to 3D or multidimensional imaging is straightforward The original continuous image would be denoted $f(x, y)$, where $x, y \in \mathbb{R}$ are the spatial coordinates. The sampled (discrete) version of the image is defined as $f[m, n]$, where $n, m \in \mathbb{Z}$. Finally, the digital image is the quantized version of the sampled signal and it can represented as $\tilde{f}[m, n]$. However, for the sake of simplicity, digital images are usually simply pictured as $f[m, n]$, or even as $f(x, y)$ (in this case $x, y \in \mathbb{Z}$). In that case, the difference between continuous and digital image must be derived from context.

2 A quick tour on 1D signals and systems

Prior to the study of the representation of images as bidemensional signals it is important to be aware of the basics of signals processing and linear systems theory for one dimensional signal. If the reader is familiar with this concepts, they can just skip this section and go straight to the main text. For deeper knowledge about it, there are many great references books [9, 3, 1].

A **signal** can be seen as a function of one or more variables that carry information about some physical phenomenon and it can me measured, stored and processed. Signals usually present a variation pattern and it is precisely in that variation pattern where the information is hold. Signals are represented as a function on one or more variables. For instance, a temporal signal is represented as $x(t)$, where $t \in \mathcal{R}$. Many different classes of signals can be considered, but, according to the utility in image processing, we can consider the following:

1. **Continuous** vs. **discrete**: according to the nature of the independent variable.
2. **Analog** vs. **digital**: an analog signal is a continuos signal that can take any value. A digital signal is a discrete signal that can only take a discrete set of values.
3. **Deterministic** vs. **random**: in a deterministic signal, all the values are known. In a deterministic signal there is some uncertainty about the values of the signal.

The concept of signal is usually link to the concept of **system**. Although the word **system** can be used in many different context to denote many different processes, in signal processing theory a system is a process in which some signals are transformed into another. When a

signal arrives, the system can modify the input signal into an output signal or, alternatively the system can react to the input signal producing a specific behavior. Among the tag *system* we will gather the filters and processes we will define along the whole book.

2.1 Linear and Time-Invariant (LTI) Systems

A special class of systems are those that fulfill two features: linearity and time invariance.

Linearity: the relationship between the input and the output signals of the system is a linear map. If we define two signals $x_1(t)$ and $x_2(t)$ so that their outputs are $y_1(t) = T[x_1(t)]$ and $y_2(t) = T[x_2(t)]$, a linear combination of those inputs, $x(t) = \alpha_1 x_1(t) + \alpha_2 x_2(t)$ will produce the same linear combination of outputs: $y(t) = T[x(t)] = \alpha_1 y_1(t) + \alpha_2 y_2(t)$.

Time invariance: if the input is delayed an amount of time t_0, the output is delayed the same amount of time.

The systems that fulfill both properties are called Linear and Time-Invariant (LTI) systems and they are the base of many procedures in signal and image processing. One of the most relevant features of these systems is that they can characterize by their *impulse response function*, i.e., the output signal when the input is an impulse. For a continuous system, the impulse is the Dirac delta $\delta(t)$, and the impulse response is defined as $h(t) = T[\delta(t)]$. Analogously, for a discrete system, the impuls is the Kronecker delta $\delta[n]$ and therefore the impulse function in $h[n] = T[\delta[n]]$.

The impulse response completely characterize an LTI system, so that the output to any input signal can be calculated using the convolution operation. For a continuous system, the continuous convolution (or convolution integral) is defined as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau)\, d\tau, \tag{1}$$

where $x(t)$ and $y(t)$ are the input and output to the system. It is important to point out that the impulse response completely characterizes the system: if $h(t)$ is known, the output response from any input signal $x(t)$ can always be computed.

For a discrete system, the sum of convolution (/or discrete convolution) is used instead:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] = x[n] * h[n]. \tag{2}$$

The convolution is an operation that satisfies the associativity, commutativity and distributivity properties.

2.2 Representation of signals in terms of frequency components

In 1D signals, many interesting features are not visible in the temporal domain, but they can be easily analyzed using alternative representations. The most common and useful is the representation using the frequency components of the signals. For an continuous energy signal, the frequency representation is given by the Fourier Transform (FT)

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \qquad \text{(Analysis equation)} \tag{3}$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega \qquad \text{(Synthesis equation)} \tag{4}$$

The FT can also be alternatively defined in terms of frequency $f = \frac{\omega}{2\pi}$. In general terms, the FT of a signal $x(t)$ exists if $\int_{-\infty}^{\infty} |x(t)|^2 dt < \infty$, i.e. $x(t)$ is an energy signal. In a more general way, we can assure the convergence with the Dirichlet conditions.

Periodic signals are not energy signals, and therefore their FT cannot be computed using the integral in eq.(3). A representation based on deltas and the coefficients of the Fourier Series (FS) are used instead:

$$X(\omega) = \sum_{k=-\infty}^{\infty} 2\pi c_k \delta(\omega - k\omega_0) \tag{5}$$

where $\omega_0 = \frac{2\pi}{T}$ is the frequency of the periodic signal and c_k are the coefficients of the FS of such signal, that can be calculated as

$$c_k = \frac{1}{T} \int_{<T>} x(t) e^{-jk\frac{2\pi}{T}t} dt.$$

Typically we use a capital letter to represent the FT of a signal. It is also common to write:

$$X(\omega) = \mathfrak{F}\{x(t)\} \quad \text{or} \quad x(t) \overset{\mathfrak{F}}{\longleftrightarrow} X(\omega).$$

The representation of a signal in the frequency domains present very interesting advantages, for instance, the analysis of the signal in terms of bandwidths and energy bands. For a LTI system, there is an added value: the convolution $y(t) = x(t) * h(t)$ is computed as a multiplication in the Fourier domain: $Y(\omega) = X(\omega)H(\omega)$.

The frequency representation of a discrete signal is calculated via the Discrete-Time Fourier Transform (DT-FT), which is defined as

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \qquad \text{(Analysis equation)} \tag{6}$$

$$x[n] = \frac{1}{2\pi} \int_{<2\pi>} X(\Omega) e^{j\Omega n} d\Omega \qquad \text{(Synthesis equation)} \tag{7}$$

Note that the DT-FT is a continuous periodic signal with 2π period:

$$X(\Omega) = X(\Omega + 2\pi).$$

In order to have a discrete representation of the discrete signal $x[n]$, the discrete Fourier Transform (DFT) is used instead. The DFT of a finite sequence $x[n]$, with $n = 0, \cdots, N-1$, is a discrete signal $X[k]$, also finite and with the same number of samples:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k = 0, \cdots, N-1 \qquad \text{(Analysis equation)} \tag{8}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}, \quad n = 0, \cdots, N-1 \qquad \text{(Synthesis equation)} \tag{9}$$

The values of the DFT can be seen as samples of the DT-FT. If we call $X_d(\Omega)$ to the DF-FT of $x[n]$, then

$$\left.\begin{array}{l} x[n] \overset{\mathcal{TF}}{\longleftrightarrow} X_d(\Omega) \\ x[n] \overset{\mathcal{DFT}}{\longleftrightarrow} X[k] \end{array}\right\} X[k] = X_d\left(\frac{2\pi}{N}k\right)$$

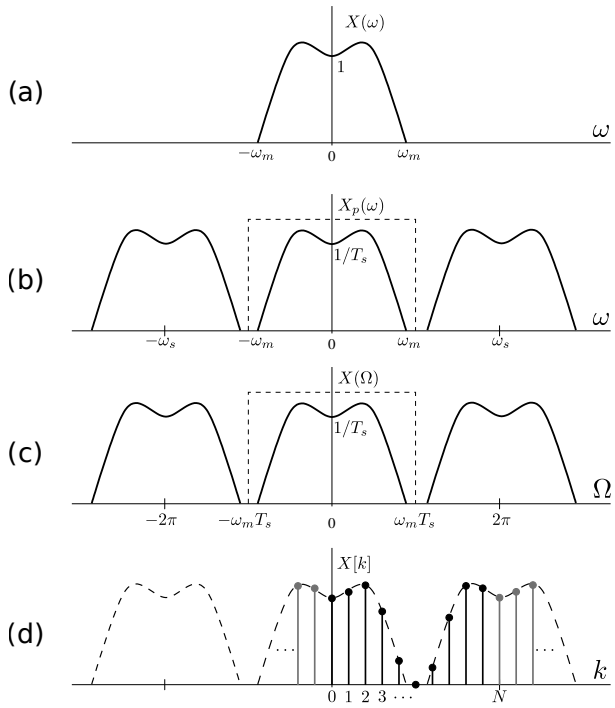


Figure 4: Relation of the different Fourier transforms for 1D signals. (a) TF of a continuous signal; (b) Continuous TF of a continuous representation of a sampled signal; (c) DT-FT, the FT of the discrete signal obtained by sampling the original signal $x(t)$ with a sampling rate T_s. It is a periodic expansion of the FT $X(\omega)$; (d) DFT, it corresponds to samples of the DT-FT $X(\omega)$.

The DFT has some interesting properties that must be taken into account, specially when working with 2D signals (images), as we will later see on this chapter. For instance, the use of the DFT assumes a periodic expanse of the signal and it Fourier representation:

$$x[n+N] = x[n] \quad X[k+N] = X[k].$$

This property has some influence over the limits of the signal, which only are a small set of points. However, when we change to 2D, it influences the whole border of an image and it can have some effects on the implementation of the algorithms.

2.3 Sampling and sampling theorem

In order to generate a discrete signal $x[n]$ from a continuous signal $x_c(t)$ it is necessary to take samples from the latter. That process is known as *sampling* and it is the first step in a digitalization procedure. There are many ways to sample a signal. In order to model it mathematically, it is common to define it as a sampling using an impulse-train sampling in which we can assure that

$$x[n] = x_c(n \cdot T_s),$$

i.e. the values of $x[n]$ are equally spaced samples of the original signal. Under certain conditions, a continuous time signal can be completely represented and recovered by these samples. The recovery conditions are given by the *sampling theorem* or *Nyquist's theorem*:

Let $x_c(t)$ be a bandwidth limited signal so that its FT fulfills that $X_c(\omega) = 0$ for $|\omega| > \omega_M$. Then, $x_c(t)$ can be recovered from its samples $x_c(n \cdot T_s)$ if

$$\omega_s > 2\omega_M,$$

where $omega_s$ is the sampling frequency defined as $\omega_s = \frac{2\pi}{T_s}$.

In order to mathematically model the effect of the sampling, an intermediate variable is often defined, called the *continuous sampled signal*, defined as

$$\begin{aligned} x_p(t) &= \sum_{n=-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s) \\ &= \sum_{k=-\infty}^{\infty} x[n]\delta(t - nT_s) \end{aligned}$$

The effect of sampling a signal in the temporal domain is translated in a replication of the FT of the original signal in the multiples of the sampling frequency. This way, the FT of $x_p(t)$ becomes

$$X_p(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_c(\omega - k\omega_s).$$

Equivalently, the DT-FT of $x[n]$ in terms of $x_c(t)$ can be written as

$$X(\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(\frac{\Omega - 2\pi k}{T_s}\right).$$

If the original signal is sampled fulfilling the Nyquist's criterion, we can write

$$X(\Omega) = \frac{1}{T_s} X\left(\frac{\Omega}{T_s}\right) \quad |\Omega| < \pi,$$

and the original signal can be (ideally) recovered using a low pass filtering.

3 Images as 2D signals

From a signal processing perspective, an image can be seen as a bidimensional signal that depends on two spacial components, width and height. The *ideal* original image (which actually does not exists), would be a continuous 2D signal defined as $f(x, y)$, where $x, y \in \mathbb{R}$ and

$$f : \mathbb{R}^2 \to \mathbb{R}.$$

In reality, the images we usually deal with are modified versions of the ideal continuous signal: the image is sampled, quantified and limited in space (the image is limited by a border). The discrete version of the image is $f[m, n]$, where $n, m \in \mathbb{Z}$ and

$$f : \mathbb{Z}^2 \to \mathbb{R}.$$

Finally, the digital image $\widehat{f}[m, n]$ is the quantized version of the discrete signal.

Digital and discrete images rely on a discrete grid as a coordinate system, see Fig. 5: Fig. 5-(a) shows a general coordinate system for a 2D signal $f[m, n]$. This system can be directly translated to an image coordinate system, Fig. 5-(b), where the origin is set to the center of the image. However, in order to avoid negative indexes, it is useful to set the origin in one of the corners of the image, see Fig. 5-(c).

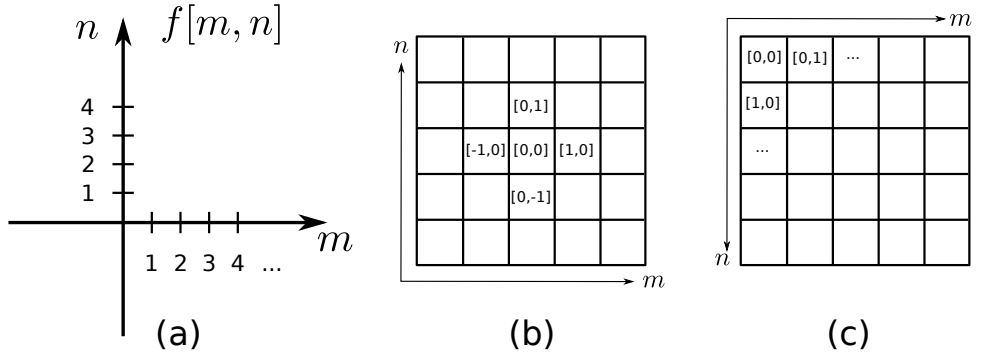


Figure 5: Representation of the coordinate system for a 2D discrete signal: (a) Coordinate system for 2D signals $f[m, n]$, with $n, m \in \mathbb{Z}$; (b) Coordinate systems for digital images considering the center of the image as the center of the space; (c) Alternative coordinate system using one corner of the image as the origin of the space.

3.1 Linear Space-Invariant (LSI) systems

The concept of signal is usually linked to the concept of a **system**. Although the word **system** can be used in many different contexts to denote many different processes, in signal/image processing theory a system is a process by which signals (eg. images) are transformed into other signals (eg. images). When an image arrives, the system can modify the input image into an output image or, alternatively the system can react to the input producing a specific behavior. Under the concept of *system*, we will include the various image processing filters defined throughout the book. Mathematically, a system is then a mapping $T[\cdot]$ that operates on an image $f(x, y)$ and produces an output image $g(x, y)$, which is given by $g(x, y) = T[f(x, y)]$. Among all the possible definitions for $T[\cdot]$, a special class of systems are those that fulfill two features, linearity and spatial invariance. Those concepts are described below.

Linearity: the relationship between the input and the output images is a linear map. That is, if we define $f_1(x, y)$ and $f_2(x, y)$ so that their outputs are $g_1(x, y) = T[f_1(x, y)]$ and $y_2(t) = T[f_2(x, y)]$, a linear combination of those inputs, $f(x, y) = \alpha_1 f_1(x, y) + \alpha_2 f_2(x, y)$ will produce the same linear combination of outputs: $g(x, y) = T[f(x, y)] = \alpha_1 g_1(x, y) + \alpha_2 g_1(x, y)$.

Spatial (shift) invariance: if the input is shifted by a given amount $[x_0, y_0]$, the output is shifted by the same amount: $g(x - x_0, y - y_0) = T[f_(x - x_0, y - y_0)]$. It is equivalent to say that the location of the origin of the coordinate system is irrelevant.

Those systems that fulfill both properties are known as Linear Space-Invariant (LSI) systems, and they are characterized by their unitary impulse response, $h(x, y) = T[\delta(x, y)]$ for continuous systems and $h[m, n] = T[\delta[m, n]]$ for discrete system, where $\delta(x, y)$ is the Dirac delta function in 2D and $\delta[m, n]$ its discrete counterpart.

This impulse response completely characterizes the system so that the output for any input image can be calculated. Sometimes systems are defined using continuous notation. That is why in what follows we will keep both notations in parallel. However, practical implementations nowadays are mostly done using discrete (digital) systems. The output of a continuous system to an input image $f(x, y)$ is given by the convolution with the impulse response of the system:

$$g(x, y) = f(x, y) * h(x, y) \tag{10}$$

$$= \int_{x'} \int_{y'} f(x', y') h(x - x', y - y') dx' dy'. \tag{11}$$

Similarly, for a discrete signal, the output is defined as:

$$\begin{aligned} g[m,n] &= f[m,n] * h[m,n] && (12) \\ &= \sum_{m'} \sum_{n'} f[m',n'] h[m-m', n-n']. && (13) \end{aligned}$$

In Fig. 6 an example of LSI filtering is depicted: discrete image $f[m,n]$ is processed using two different LSI systems via convolution.

Properties of the 2D convolution

The convolution defines a product that satisfies the following properties:

Commutativity: $f(x,y) * h(x,y) = h(x,y) * f(x,y)$.

Associativity: $f(x,y) * (h(x,y) * g(x,y)) = (f(x,y) * h(x,y)) * g(x,y)$.

Distributivity: $f(x,y) * (h(x,y) + g(x,y)) = f(x,y) * h(x,y) + f(x,y) * g(x,y)$.

Identity element: $f(x,y) * \delta(x,y) = f(x,y)$.

Differenciation: $(f(x,y) * h(x,y))' = f'(x,y) * h(x,y) = f(x,y) * h'(x,y)$.

Stability: a system defined by a impulse response $h(x,y)$ or $h[m,n]$ is stable if

$$\begin{aligned} \int_x \int_y |h(x,y)| dx dy &< \infty \\ \sum_m \sum_n |h[m,n]| &< \infty \end{aligned}$$

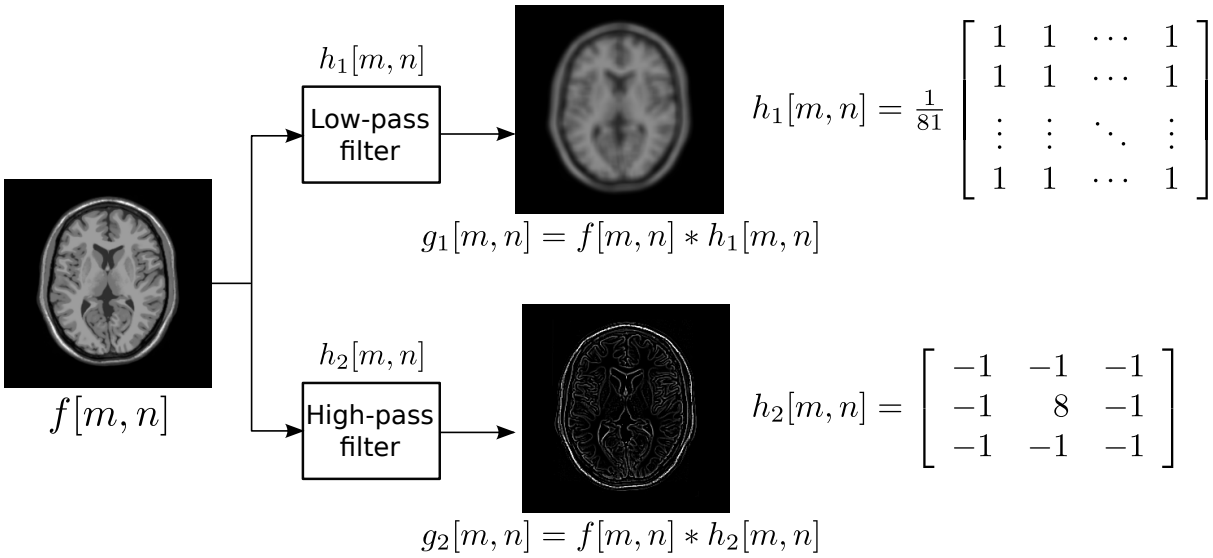


Figure 6: Example of image processing using LSI systems. Two different impulse responses have been considered: (1) $h_1[m,n]$ corresponds to a 9×9 low pass filter; (2) $h_2[m,n]$ is a high pass filter.

3.2 LCI systems

In practical situations, images are not infinite, but limited in space. Digital images are known to have a limited number of rows and columns. The limitation of a $M \times N$ discrete signal is equivalent to multiply an limitless image $f[m,m]$ by a rectangular function $r[m,n]$

$$f_d[m,m] = f[m,m] \cdot r[m,n] = \begin{cases} f[m,n] & 0 \leq m \leq M-1,\ 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

where

$$r[m,n] = \begin{cases} 1 & 0 \leq M-1,\ 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

If LSI systems are used to process signal $f_d[m,m]$, we must be aware that outside the borders of the image, the signal value is 0. As an effect of this area of zero-value, the different procedures may fail near the borders. In addition, in image processing, one of the requirements of the processing systems is that the output has the exact same size that the input. In an LSI system, if the input $f[m,n]$ is an $M_1 \times N_1$ signal, the impulse response is $M_2 \times N_2$, the output $g[m,n] = f[m,n] * h[m,n]$ will be a $(M_1 + M_2 - 1) \times (N_1 + N_2 - 1)$ signal, which is greater than the original size. If only *effective* values of the convolution are taken into account (those in which no pixels from the zero-values area are involved), the size would be smaller: $(M_1 - M_2 + 1) \times (N_1 - N_2 + 1)$.

In order to avoid these issues, a periodic extension of the image can also be assumed. A periodic expansion of the $M \times N$ discrete signal signal $f_d[m,m]$ is defined as

$$f_p[m + k_1 M, n + k_2 N] = f_p[m,n] = f[m,n], \quad (k_1, K_2, M, N) \in \mathcal{Z}.$$

This extension of the image assumes that all the operations are done in a space of periodic signals. The systems involved are no longer LSI but LCI: linear and circular-shift invariant. LCI systems are also completely characterized by the unitary impulse response, $h[m,n]$ and the output can be calculated using the circular discrete convolution

$$\begin{aligned} g[m,n] &= f_p[m,n] \circledast h[m,n] && (14) \\ &= \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} f_p[m',n'] h[(m-m')\text{modM}, (n-n')\text{modM}] && (15) \end{aligned}$$

It can also be seen as the periodic expansion of the discrete convolution. If

$$g_0[m,n] = f[m,n] * h[m,n]$$

then

$$g[m,n] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} g_0[m + kM, n + lN].$$

As we will see in the next section, LCI systems and the circular convolution naturally arise when dealing with the Discrete Fourier Transform.

In Fig. 7 we show the periodic expansion of two different images. Note that, in the first one, the periodic expansion creates discontinuities in the border of the image. This effect can produce undesired behaviors when using certain filtering procedures. However, note that in some medical images, like the MRI slice in Fig. 7, due to the black background present this is no longer an issue.

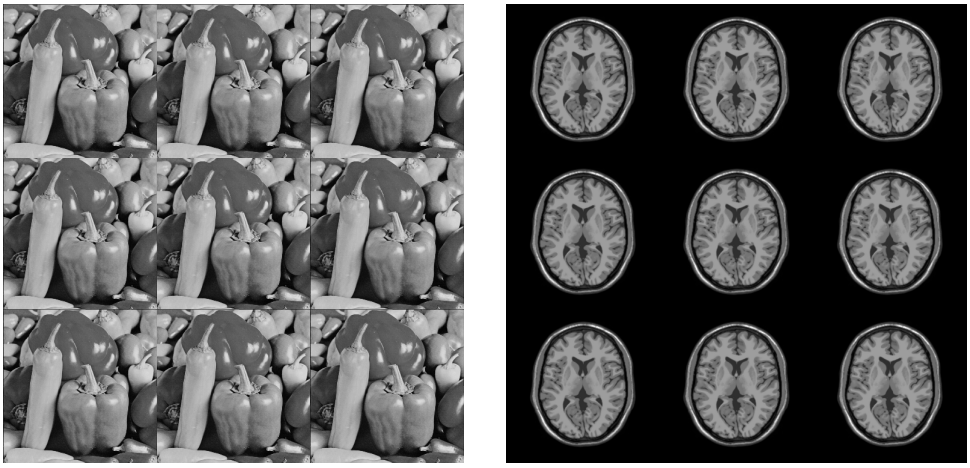


Figure 7: Periodic expansion of digital image $f[m, n]$. This expansion is used in LCI systems and implicitly assumed when DFT is used.

4 Frequency representation of 2D signals

One powerful in image processing is to use alternative representations of the signal that allow a better representation of certain features of the image. One of the most used transformation is the Fourier Transform, which is a representation of the distributions of frequencies within the image.

4.1 Fourier Transform of continuous signals

The 2D FT of a continuous signal $f(x, y)$ is define as

$$F(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)e^{-j2\pi(ux+vy)}dxdy \qquad \text{(Analysis equation)} \tag{16}$$

$$f(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u, v)e^{j2\pi(ux+vy)}dudv \qquad \text{(Synthesis equation)} \tag{17}$$

where u and v are the spatial frequency components. The FT can also be computed over angular frequencies $\omega_i = 2\pi\xi_i$ and therefore

$$f(x, y) = \left(\frac{1}{2\pi}\right)^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\omega_1, \omega_2)e^{j(\omega_1 x+\omega_2 y)}d\omega_1 d\omega_2 \tag{18}$$

For some applications it is useful to define the FT using polar coordinates:

$$F_p(\xi, \phi) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)e^{-j2\pi(\xi\cos\phi x+\xi\sin\phi y)}dxdy \tag{19}$$

$$= \int_{0}^{\infty}\int_{0}^{2\pi} f_p(\rho, \theta)e^{-j2\pi(\xi\cos\phi\rho\cos\theta+\xi\sin\phi\rho\sin\theta)}\rho d\rho d\theta \tag{20}$$

Properties of the FT: Some of the main properties of the 2D FT with special relevance for image processing are the following:

1. Uniqueness: if $f(x, y)$ is square-integrable, the FT is unique and reversible.

2. Linearity:

$$\alpha f(x,y) + \beta g(x,y) \overset{\mathfrak{F}}{\longleftrightarrow} \alpha F(u,v) + \beta F(u,v)$$

3. Shift:

$$f(x - x_0, y - y_0) \overset{\mathfrak{F}}{\longleftrightarrow} e^{j2\pi(x_0 u + y_0 v)} F(u,v).$$

4. Separability: the 2D FT is equivalent to two consecutive 1D FT in normal directions:

$$F(u,v) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x,y) e^{-j2\pi v y} dy \right] e^{-j2\pi u x} dx.$$

5. Convolution: the FT of a convolution is a product.

$$g(x,y) = f(x,y) * h(x,y) \overset{\mathfrak{F}}{\longleftrightarrow} G(u,v) = F(u,v)H(u,v)$$

6. Rotation: the rotation of $f(x,y)$ implies a rotation of its FT, see Fig 8:

$$\mathfrak{F}\left[f_p(\rho, \theta + \alpha)\right] = F_p(\xi, \phi + \alpha)$$

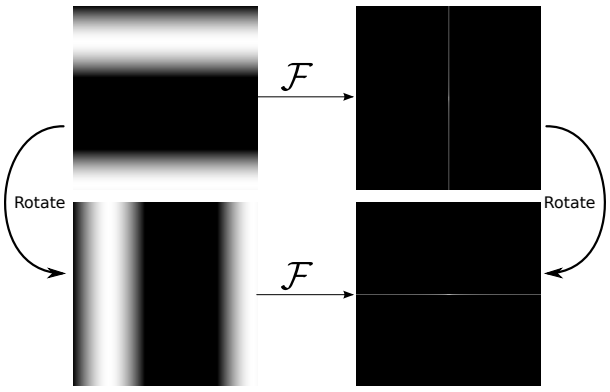


Figure 8: The rotation of an image implies the rotation of its FT.

4.2 Discrete-Space Fourier Transform

For a discrete signal $f[m,n]$ we can define the 2D FT as

$$F(\Omega_1, \Omega_2) = \sum_m \sum_n f[m,n] e^{-j(\Omega_1 m + \Omega_2 n)} \qquad \text{(Analysis equation)}$$

$$x[m,n] = \left(\frac{1}{2\pi}\right)^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} F(\Omega_1, \Omega_2) e^{j(\Omega_1 m + \Omega_2 n)} d\Omega_1 d\Omega_2 \qquad \text{(Synthesis equation)}$$

We call this transform the Discrete-space Fourier Transform (DS-FT). Note that the DS-FT is a periodic continuous signal, with period 2π in both frequency directions.

The computational alternative to the DS-FT is the 2D DFT, where the signal and its transform are both discrete.

4.3 2D Discrete Fourier Transform

Let us assume that $f[m,n]$ is a $M \times N$ finite discrete image. The 2D DFT of that signal is computed as

$$F[k_1,k_2] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f[m,n]e^{-j\frac{2\pi}{N}(k_1 m+k_2 n)} \qquad \text{(Analysis equation)} \tag{21}$$

$$f[m,n] = \frac{1}{N^2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} F[k_1,k_2]e^{j\frac{2\pi}{N}(k_1 m+k_2 n)} \qquad \text{(Synthesis equation)} \tag{22}$$

Properties of the 2D DFT:

1. Periodicity: The DFT assumes a periodic extension of image $f[m,n]$ so that $f[m+M,n+N] = f[m,n]$, and a periodic expansion of the DFT: $F[u+M,v+N] = F[u,v]$.
2. The values of the DFT are samples of the DS-FT:
$$\left.\begin{array}{l} f[m,n] \overset{\mathcal{FS}-\mathcal{FT}}{\longleftrightarrow} F_d(\Omega_1,\Omega_2) \\ f[m,n] \overset{\mathcal{DFT}}{\longleftrightarrow} F[k_1,k_2] \end{array}\right\} F[k_1,k_2] = F_d\left(\frac{2\pi}{M}k_1, \frac{2\pi}{N}k_2\right).$$
3. The DFT can be implemented using fast computational solutions, like the Fast Fourier Transform (FFT), with a considerable reduction in the number of operations.
4. Conjugate Symmetry: $F[k_1,k_2] = F^*[M-k_1, N-k_2]$.
5. (Circular) convolution: the DFT of the circular convolution is a product:
$$f[m,n] \circledast h[m,n] \overset{\mathcal{DFT}}{\longleftrightarrow} F[k_1,k_2]\cdot F[k_1,k_2].$$
6. Product:
$$f[m,n]\cdot h[m,n] \overset{\mathcal{DFT}}{\longleftrightarrow} F[k_1,k_2] \circledast F[k_1,k_2].$$

Note that the LCI systems previously defined implicitly arise when working with DFT. When dealing with limited discrete signals and DFT, we are assuming the periodic expansion of the images that makes the traditional convolution unfit for the problem.

4.4 Discrete cosine transform

One of the problems of the DFT is that the periodicity that it implies creates non-natural transitions in images. In order to obtain a better behavior in the borders, the Discrete cosine transform (DCT) can alternatively be used. The DCT of an $N \times N$ image $f[m,n]$ is equivalent to the DFT of an extension of the image into a $2N \times 2N$ (see Fig. 9), $f_e[m,n]$.

$$\text{DCT}\{f[m,n]\} \quad = \quad \text{DFT}\{f_e[m,n]\} = F[k_1,k_2]$$

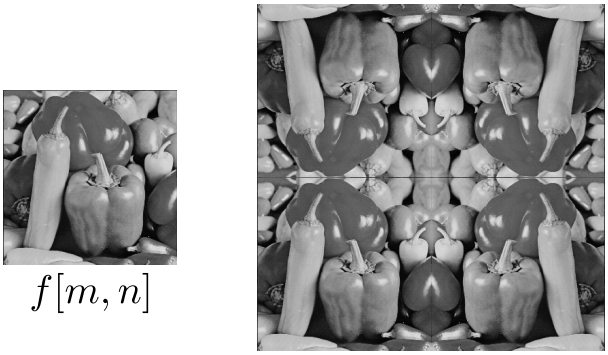
The DCT is a separable linear transformation. For a 2D signal is defined as

$$F[k_1,k_2] \quad = \quad \alpha(k_1)\alpha(k_2)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f[m,n]\cos\left(\frac{(2m+1)k_1\pi}{2M}\right)\cos\left(\frac{(2n+1)k_2\pi}{2N}\right) \tag{23}$$

(Analysis equation)

$$f[m,m] \quad = \quad \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} \alpha(k_1)\alpha(k_2)F[k_1,k_2]\cos\left(\frac{(2m+1)k_1\pi}{2M}\right)\cos\left(\frac{(2n+1)k_2\pi}{2N}\right) \tag{24}$$

(Synthesis equation)



$f[m,n]$ $\qquad$ $f_e[m,n]$

Figure 9: Symmetric expansion of an image $f[m,n]$ implicitly assumed by the DCT. The discontinuities in the border of the image are reduced with respect to the periodic expansion in Fig. 7.

where

$$\alpha(k_1) = \begin{cases} \frac{1}{\sqrt{M}} & \text{if } k_1 = 0 \\ \sqrt{\frac{2}{M}} & \text{if } k_1 = 1, \cdots, M-1 \end{cases}$$

$$\alpha(k_2) = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } k_2 = 0 \\ \sqrt{\frac{2}{N}} & \text{if } k_2 = 1, \cdots, M-1 \end{cases}$$

5 Image Sampling

5.1 Introduction

So far, medical images have been modeled as 2D or 3D continuous functions, and algorithms for medical image analysis are often conceived for such a continuous representation. Nevertheless, in practice, modern medical image processing and storing require images to be available in digital form, that is, as arrays of finite lengths of binary words [4]. Transforming a continuous image into a digitized one comprises (at least) two steps: image sampling and image quantization. Both processes are described in this chapter, with a special emphasis on the former step. For simplicity, we will focus on the 2D-case, being the extension to 3D straightforward.

Image sampling can be seen as a discretization of a continuous image in the spatial domain. That is, given a continuous image $f(x,y)$ with $(x,y) \in \mathbb{R}^2$, the output of image sampling is a two dimensional array of points $f[m,n]$ with $(m,n) \in \mathbb{Z}^2$, where each point represents a particular sample $f(x,y)$, i.e.,

$$f[m,n] = f(m \cdot \Delta_x, n \cdot \Delta_y), \tag{25}$$

Δ_x and Δ_y are the so-called sampling rates in the x and y direction. Points $(m,n) \in \mathbb{Z}^2$ are often referred as pixels in 2D and voxels in the 3D case.

The fundamental question in image sampling is the following: Given samples $f[m,n]$ of $f(x,y)$, *is it possible to exactly reconstruct* $f(x,y)$*, by solely using information from* $f(m,n)$*?* If that is so, *which conditions should the image* $f(x,y)$ *fulfill? How should we then choose the values of* Δ_x *and* Δ_y*?* The answer to these questions are provided by a branch of mathematical analysis called sampling theory.

5.2 Basics on 2D sampling theory

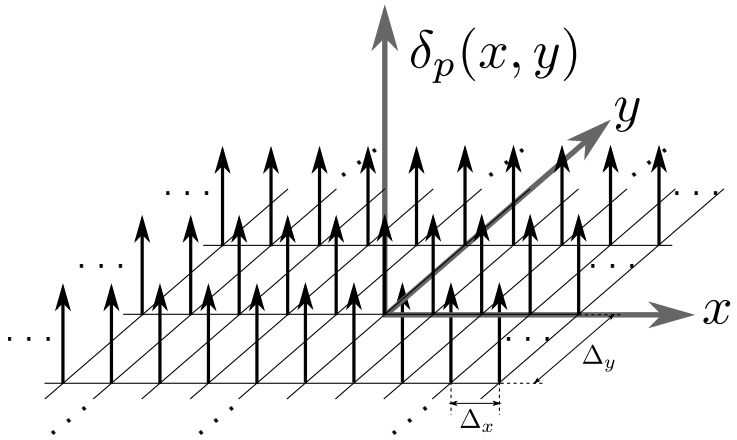


Figure 10: Two-dimensional infinite array of Dirac delta functions.

Let us consider an ideal image sampling function, which is a two-dimensional infinite array of Dirac delta functions located on a rectangular grid with spacing Δ_x and Δ_y, i.e. (10),

$$\delta_p(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - m\Delta_x, y - n\Delta_y). \tag{26}$$

Its Fourier transform is another train of Dirac functions with spacing $1/\Delta_x$ and $1/\Delta_y$,

$$\hat{\delta}_p(u,v) = \frac{1}{\Delta_x \Delta_y} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(u - m/\Delta_x, v - n/\Delta_y). \tag{27}$$

The sampled image is thus defined as the product of the original function and the sampling function:

$$\begin{aligned} f_p(x,y) &= f(x,y) \cdot \delta_p(x,y) && (28)\\ &= \sum_m \sum_n f(m\Delta_x, n\Delta_y)\delta(x - m\Delta_x, y - n\Delta_y) \\ &= \sum_m \sum_m f[m,n]\delta(x - m\Delta_x, y - n\Delta_y), && (29) \end{aligned}$$

and its Fourier transform is given by

$$\begin{aligned} F_p(u,v) &= F(u,v) * \hat{\delta}_p(u,v) && (30)\\ &= \frac{1}{\Delta_x \Delta_y} \sum_m \sum_n F(u,v) * \delta(u - m/\Delta_x, v - n/\Delta_y) \\ &= \frac{1}{\Delta_x \Delta_y} \sum_m \sum_n F(u - m/\Delta_x, v - n/\Delta_y). && (31) \end{aligned}$$

Important to note, $F_p(u,v)$ contains infinite overlapped replicas of the original spectrum of $f(x,y)$, that is $F(u,v)$, so in principle, it is impossible to recover $F(u,v)$ and hence $f(x,y)$ from $F_p(u,v)$. However, let us assume that image $f(x,y)$ is band-limited, that is, $F(u,v)$ has finite support. As an illustration we assume that the support of $F(u,v)$ is rectangular, i.e., it is zero

for $|u| > B_x$, $|v| > B_y$. We will further assume that $2B_x \leq 1/\Delta_x$, $2B_y \leq 1/\Delta_y$. If that is so, it is straightforward to prove that the replicas of $F(\xi_x, \xi_y)$ in eq. (31) do not overlap, and hence,

$$F(u,v) = \Delta_x \Delta_y F_p(u,v) \Pi\left(\frac{u}{2B_x}\right) \Pi\left(\frac{v}{2B_y}\right), \tag{32}$$

where $\Pi\left(\frac{u}{2B_x}\right)$ and $\Pi\left(\frac{v}{2B_y}\right)$ are rectangular functions with width $2B_x$ and $2B_y$, respectively. Inverse Fourier transforming eq. (32) yields

$$f(x,y) = r_x r_y f_s(x,y) * \operatorname{sinc}(2B_x x) \operatorname{sinc}(2B_y y) \tag{33}$$

where $r_x \leq 1$ and $r_y \leq 1$ are the sampling rate parameters, defined as

$$r_x = \frac{2B_x}{1/\Delta_x}, \quad r_y = \frac{2B_y}{1/\Delta_y}. \tag{34}$$

Substituting eq. (29) into eq. (33), we arrive at the following expression:

$$f(x,y) = r_x r_y \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[m,n] \operatorname{sinc}(2B_x x - r_x m) \operatorname{sinc}(2B_y y - r_y n). \tag{35}$$

If $1/\Delta_x$ and $1/\Delta_y$ are chosen to be equal to $2B_x$ and $2B_y$, then $r_x = 1$ and $r_y = 1$, and eq. (35) becomes the celebrated cardinal series, also called Whittaker-Shannon-Kotelnikov sampling theorem formula, which can be found in any basic book of image processing. Frequencies $2B_x$ and $2B_y$ are called the Nyquist frequencies, or Nyquist rate. If $1/\Delta_x$ and $1/\Delta_y$ are higher than the Nyquist frequencies, i.e., sampling rate parameters are strictly smaller than one, we are in the case of *oversampling*. In both situations, exact reconstruction is possible with the formula eq. (35). If $1/\Delta_x$ and $1/\Delta_y$ are smaller than the Nyquist frequencies, the replicas in eq. (31) do overlap. As a result, the derivation that we followed to arrive at the expression in eq. (35) is not valid. This case is referred as *undersampling*. Reconstruction is no longer possible in this situation. Those three scenarios are illustrated in Fig. 11.

In summary, a band-limited image $f(x,y)$ with rectangular bandwidth $2B_x$ and $2B_y$ can be reconstructed without error provided $2B_x\Delta_x \leq 1$ and $2B_y\Delta_y \leq 1$. This condition is called the Nyquist condition.

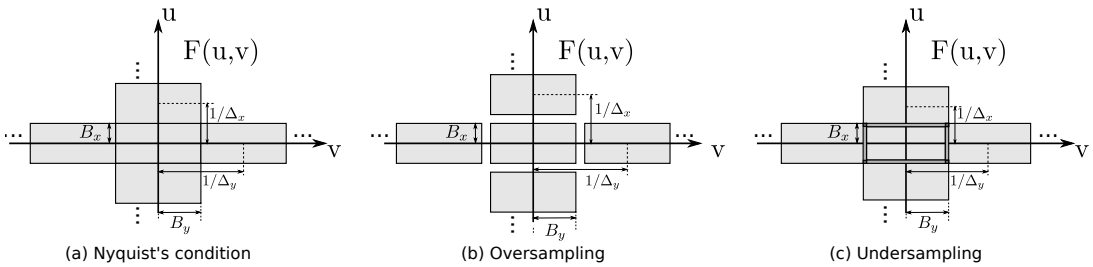


Figure 11: Three scenarios depending on the choice of Δ_x and Δ_y. The Nyquist's regime (a) and oversampling (b) allows exact reconstruction whereas it is no longer possible when under sampling occurs (c).

5.2.1 Inexact reconstruction

Though exact reconstruction is possible if the Nyquist conditions hold, it should be noted that, in practice, real-world images are barely bandlimited. Moreover, instead of an infinite set of samples, the number of image samples is always finite. These two cases violate the assumptions we have made above. Therefore, eq .(35) is not directly applicable. We are then in a scenario where inexact reconstruction is the best we can achieve. This two type of errors are described below.

1. **Aliasing:** When images are not bandlimited, the Nyquist conditions never hold. In this undersampling scenario, spurious spatial frequency components will be introduced into the reconstruction. This effect is called aliasing. The effects of aliasing in an actual image are shown in Fig. 12. In this example, the starting point is a a sampled image of a brickwall (Fig. 12-(a)). The original continuous image is assumed to be sampled fulfilling the Nyquist condition. To mimic the effect of sampling with a different sampling step, we decimate the discrete image by a factor of five and ten respectively. Images are then zoomed in at the same size of the original image. Observe that when the sampling frequency is reduced a factor of five, the particular pattern of the brickwall is still noticeable, whereas it is impossible to distinguish when the sampling frequency is reduced ten-fold. In the latter case, the Nyquist conditions do not hold, an therefore aliased frequencies are present in the image. This type of artificial low-frequency spatial components are known in the image jargon as Moiré patterns.

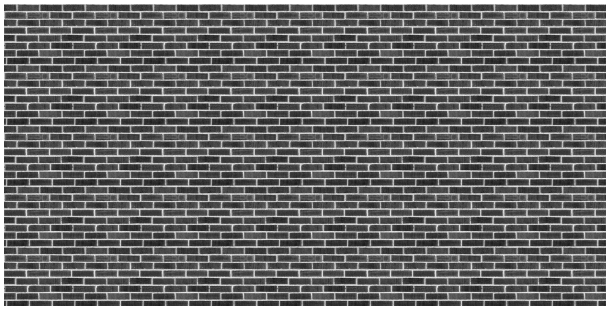
   Aliasing errors can be substantially reduced by low-pass filtering the image before sampling. This attenuates the spectral foldover that appears when replicas of the sampled image overlap. In Figure, the same image was sampled before it is first low-pass filtered and thus sampled with a sampling frequency reduced by a factor of ten.

   Obviously, attenuating high spatial frequencies to avoid undersampling introduces a loss of information in the sampled image. As a result, there is always a trade-off between sampled image resolution and aliasing error. In a practical design, consideration must be given to the choice of the low-pass filter as well as the degree of aliasing error that is acceptable. It may be the case that small aliasing error is preferable instead of a substantial loss of resolution. For a more elaborated discussion, the reader is referred to [10].

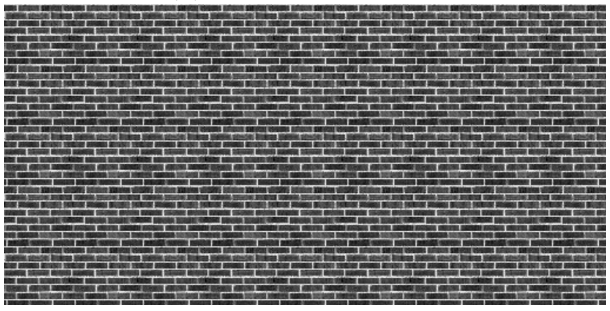
2. **Truncation error:** Even if Nyquist conditions hold, exact reconstruction is not possible, since the number of samples is finite and thus Eq. (35) is unrealizable. A typical line of action is then to calculate expression Eq. (35) with a finite number of samples N, thereby truncating the summand. This has the effect of loss of resolution, manifested in terms of blurring.

In the previous example, it has been assumed that the finite support of $F(u, v)$ is rectangular. Clearly, the most efficient sampling scheme is achieved when $2B_x\Delta_x = 1$ and $2B_y\Delta_y = 1$, since replicas do not overlap and all the space (u, v) is filled in. This gives the biggest Δ_x and Δ_y that we may choose to reconstruct the sampled image. A smaller Δ_x and Δ_y, i.e., turning in more relaxed sampling conditions, will necessarily end up in aliasing. This is so because the replicas of $F(u, v)$ are maximally packed.

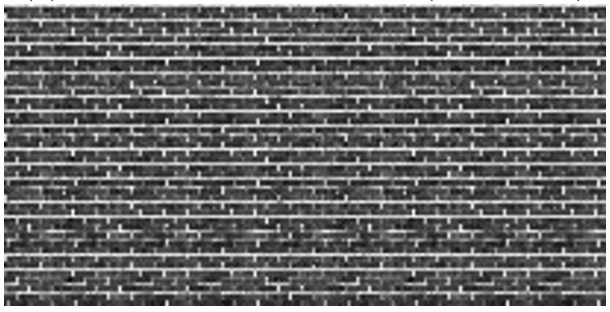
It should be noted, though, that continuous images that are band-limited can have frequency support with any arbitrary shape. Let us assume, for example, that the support of $F(u, v)$ is circular, that is, $F(u, v)$ is zero for $u^2 + v^2 > r^2$, being r the radius. Clearly, if $2r\Delta_x < 1$ and $2r\Delta_y < 1$, the replicas of $F(u, v)$ do not overlap (Fig. 14), and reconstruction is theoretically possible with the same formula as Eq. (35). Nevertheless, the space (u, v) is not completely



(a) Continuous image



(b) Sampled by a factor of five (No aliasing)



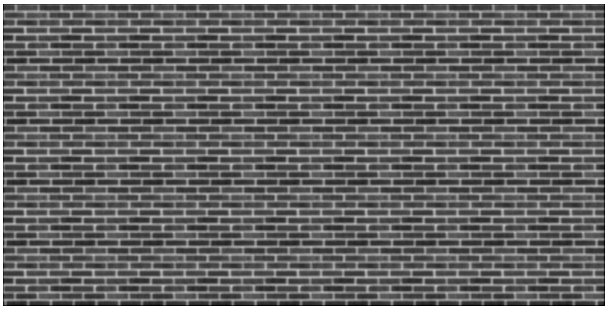
(c) Sampled by a factor of ten (Aliasing)

Figure 12: A continuous image of a brickwall (a) is sampled at a factor of 5x, respecting Nyquist conditions. When sampling frequency is reduced a factor of ten, aliasing occurs, and is manifested in the image domain as artificial texture patterns (Moiré patterns).

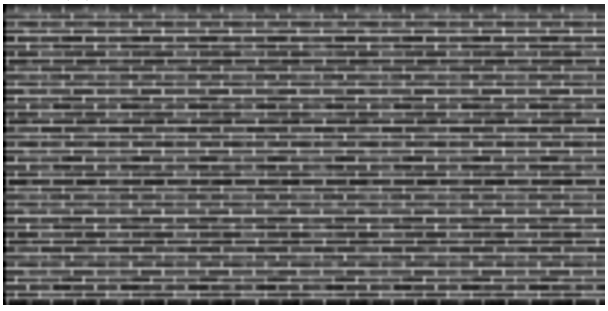
filled in, and therefore the reader may wonder whether it would be possible to relax the sampling condition by trying to fill in the spectral gaps that are shown in Fig. 16.

Evidently, it is impossible to fill in the space (u, v) with a circular spectral support if a rectangular sampling scheme is implemented since replicas are repeated in a rectangular fashion. The branch of geometry that studies how to arrange circles in such a way no overlapping occurs is named circle packing. Since completely packing is impossible, the interesting question is to find the arrangement that obtains the maximum proportion of space covered. Translating in terms of image sampling, the optimal arrangement determines the optimal sampling scheme. It is well known that in a two-dimensional Euclidean space, the optimal arrangement for a circle is an hexagonal tiling. The sampling scheme that produces this type of tiling for continuous image with circular spectral support is named hexagonal sampling.

In what follows, we will generalize the sampling theory presented above for any non-rectangular sampling scheme with arbitrary spectral support. Then, we will illustrate the general case with both rectangular and circular support, and the concept of Nyquist density will be introduced. Let us start by defining a generalized version of the two-dimensional array



(a) Continuous image low-pass filtered



(b) Sampled by a factor of ten (No aliasing)

Figure 13: When continuous image is low-pass filtered (a), high-frequency components are left out, and the effect of aliasing is severely reduced when reducing the sampling frequency a factor of ten (b).

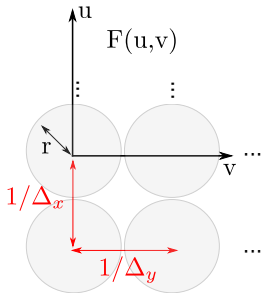


Figure 14: Rectangular sampling of an image with circular spectral support.

of Dirac delta functions of Eq. (26):

$$\delta_p(\boldsymbol{x}) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(\boldsymbol{x} - \boldsymbol{Q} \begin{pmatrix} m \\ n \end{pmatrix}) \tag{36}$$

with $\boldsymbol{x} = (x, y)^T$, $\boldsymbol{Q} = (\boldsymbol{q}_1, \boldsymbol{q}_2)$ the so-called sampling matrix and the two-dimensional column vectors $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, being the sampling directions. Observe that, indeed, this is a generalization of Eq. (26) since the latter is a special case of Eq. (36) with $\boldsymbol{q}_1 = (\Delta_x, 0)^T$ and $\boldsymbol{q}_2 = (0, \Delta_y)^T$.

The Fourier transform of $\delta_p(\boldsymbol{x})$ is [8]

$$\hat{\delta}_p(\boldsymbol{u}) = |\boldsymbol{P}| \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(\boldsymbol{u} - \boldsymbol{P} \begin{pmatrix} m \\ n \end{pmatrix}) \tag{37}$$

with $\boldsymbol{u} = (u, v)^T$, $\boldsymbol{P} = \boldsymbol{Q}^{-T}$, and $|\boldsymbol{P}|$ the determinant of $\boldsymbol{P}$. Let us assume as well that the continuous image $f(\boldsymbol{x})$ is bandlimited in the sense of $F(\boldsymbol{u}) = 0$ if $\boldsymbol{u} \notin \mathcal{C}$, where $F(\boldsymbol{u})$ is its

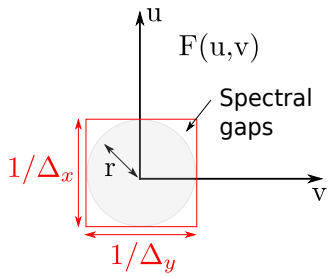


Figure 15: Spectral gaps that appear with rectangular sampling and circular spectral support.

Fourier transform and $\mathcal{C}$ its spectral support. Multiplying $f(\boldsymbol{x})$ by $\delta_p(\boldsymbol{x})$ and then applying the Fourier transform, it is possible to prove that

$$F_p(\boldsymbol{u}) = |\boldsymbol{P}| \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F(\boldsymbol{u} - \boldsymbol{P} \begin{pmatrix} m \\ n \end{pmatrix}). \tag{38}$$

Therefore, $F(\boldsymbol{u})$ is replicated in the $\boldsymbol{u} = (u, v)^T$ space on tiles with a periodicity matrix $\boldsymbol{P} = (\boldsymbol{p}_1, \boldsymbol{p}_2)$, where vectors $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ determine the directions of replication of the spectral support $\mathcal{C}$.

Similar to Eq. (32), we can write

$$F(\boldsymbol{u}) = \frac{F_p(\boldsymbol{u}) \Pi_{\mathcal{C}}(\boldsymbol{u})}{|\boldsymbol{P}|}, \tag{39}$$

with $\Pi_{\mathcal{C}}(\boldsymbol{u})$ defined as

$$\Pi_{\mathcal{C}}(\boldsymbol{u}) = \begin{cases} 1 \text{ if } \boldsymbol{u} \in \mathcal{C} \\ \\ 0 \text{ otherwise.} \end{cases} \tag{40}$$

Inverse Fourier transforming Eq.(39) gives

$$f(\boldsymbol{x}) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[m, n] s_{\mathcal{C}}(\boldsymbol{x} - \boldsymbol{Q} \begin{pmatrix} m \\ n \end{pmatrix}) \tag{41}$$

with $f[m, n] = f(\boldsymbol{Q} \begin{pmatrix} m \\ n \end{pmatrix})$ and $s_{\mathcal{C}}(\boldsymbol{x})$ given by

$$s_{\mathcal{C}}(\boldsymbol{x}) = |\boldsymbol{Q}| \int_{\boldsymbol{u} \in \mathcal{C}} e^{-2\pi i \langle \boldsymbol{u}, \boldsymbol{x} \rangle} d\boldsymbol{u}. \tag{42}$$

5.3 Nyquist sampling density

A sampling scheme is efficient when the number of samples per unit area is kept low. The number of samples per unit area, or, sampling density, is defined by means of the inverse of the area of the parallelogram that is defined by vectors $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. Such an area is given by $|\boldsymbol{Q}|$, thus, the sampling density SD is defined as [8]

$$\text{SD} = \frac{1}{|\boldsymbol{Q}|} = |\boldsymbol{P}| \text{ samples/unit area.} \tag{43}$$

Given a particular image $f(\boldsymbol{x})$ with spectral support $\mathcal{C}$, the minimum value of SD that can be achieved with a given sampling scheme without aliasing is known as the *Nyquist sampling*

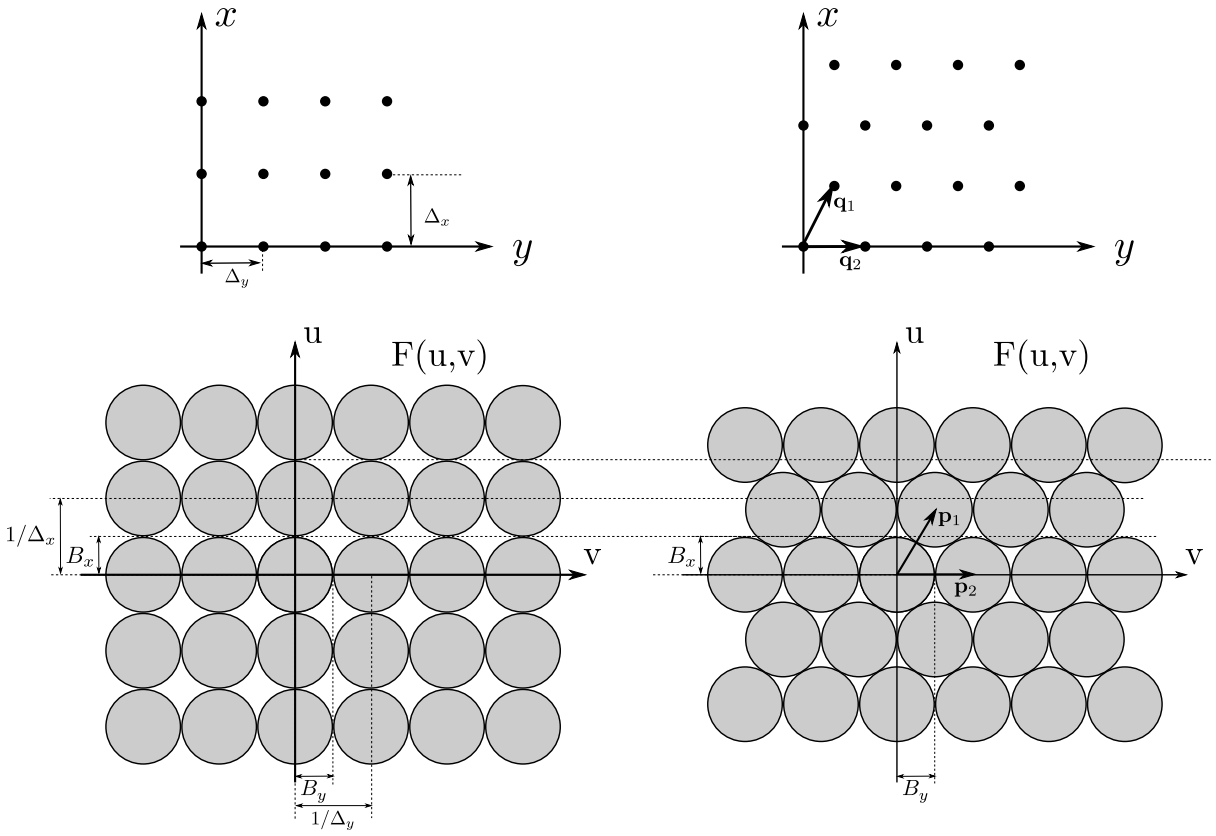


Figure 16: Comparative between rectangular sampling (left) and hexagonal sampling (right). Note that circles are maximally packaged with hexagonal sampling.

density. There may exist different sampling schemes that achieve *Nyquist sampling density*. If the spectral support $\mathcal{C}$ is rectangular, we have already stated that rectangular sampling achieves the *Nyquist sampling density* if $2B_x\Delta_x = 1$ and $2B_y\Delta_y = 1$. This choice is not unique and other different tilings can be shown to reach the minimum possible density.

In case of circular $\mathcal{C}$, the gain of using hexagonal sampling over rectangular sampling can be substantial. Indeed, if $\mathcal{C}$ is circular with radius r, and rectangular sampling is used with $\Delta_x = \frac{1}{2r}$ and $\Delta_y = \frac{1}{2r}$, then

$$\boldsymbol{Q} = \begin{pmatrix} \Delta_x & 0 \\ 0 & \Delta_y \end{pmatrix} = \begin{pmatrix} \frac{1}{2r} & 0 \\ 0 & \frac{1}{2r} \end{pmatrix} \tag{44}$$

and

$$\boldsymbol{P} = \begin{pmatrix} \frac{1}{\Delta_x} & 0 \\ 0 & \frac{1}{\Delta_y} \end{pmatrix} = \begin{pmatrix} 2r & 0 \\ 0 & 2r \end{pmatrix} \tag{45}$$

given a value of SD of

$$\mathrm{SD}_{\mathrm{rect}} = 4r^2. \tag{46}$$

If hexagonal sampling is applied, which is defined through the following sampling matrix

$$\boldsymbol{Q} = \begin{pmatrix} T & -T \\ \frac{T}{\sqrt{3}} & \frac{T}{\sqrt{3}} \end{pmatrix} \tag{47}$$

yielding

$$\boldsymbol{P} = \begin{pmatrix} \frac{1}{2T} & -\frac{1}{2T} \\ \frac{\sqrt{3}}{2T} & \frac{\sqrt{3}}{2T} \end{pmatrix} \tag{48}$$

with $T = \frac{1}{2r}$, then the sampling density can be shown to be

$$\mathrm{SD}_{\mathrm{hex}} = 2\sqrt{3}r^2. \tag{49}$$

The reduction of sampling density with respect of rectangular sampling is

$$r = \frac{\mathrm{SD}_{\mathrm{hex}}}{\mathrm{SD}_{\mathrm{rect}}} = \frac{2\sqrt{3}r^2}{4r^2} = \frac{\sqrt{3}}{2} = 0.866. \tag{50}$$

That is, hexagonal sampling reduces the sampling density by 13.4% over rectangular sampling. For a given value r there is no other sampling scheme that gives a SD such that

$$\mathrm{SD} \leq \mathrm{SD}_{\mathrm{hex}}. \tag{51}$$

Thus $\mathrm{SD}_{\mathrm{hex}}$ is the Nyquist sampling density.

6 Image interpolation

At the beginning of Section 5, it was mentioned that sampling theory provides the conditions and rules to choose the sampling settings in order to exactly reconstruct any point of an image $f(x, y)$ from a discrete set of points $f[m, n]$. Such a question is intimately linked to interpolation theory. Interpolation is the task of constructing new points of a given mathematical function from knowledge of a discrete set of points. Image interpolation is thus a subfield of mathematical analysis which deals with the generation of new pixel intensities from a given set of values.

In most ot the cases, any interpolation method can be modeled as [5]

$$\hat{f}(x, y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[m, n] h\left(\frac{x - m\Delta_x}{\Delta_x}, \frac{y - n\Delta_y}{\Delta_y}\right), \tag{52}$$

where $h(x, y)$ is the interpolation kernel. Observe that $\hat{f}(x, y) \neq f(x, y)$, except for very singular cases, where $h(x, y)$ is then called an ideal interpolation kernel. For example, kernel $h(x, y)$ defined as (Eq. (35))

$$h(x, y) = r_x r_y \operatorname{sinc}(r_x x) \operatorname{sinc}(r_y y) \tag{53}$$

is an ideal interpolation kernel for images with spectral support confined to a rectangle with width and height B_x and B_y, respectively. As we have seen, ideal interpolators are impractical since the conditions imposed on the image may not hold, and perhaps more importantly, ideal interpolators do not have limited spatial support, thus, infinite samples are required.

There are multiple options to define $h(x, y)$, however, interpolation methods are required to fulfill a certain list of conditions, which ultimately determine the shape of $h(x, y)$. Here, we just mention two of them [6]

1. *Image should not be modified if it is interpolated in the same grid.*
   Mathematically, this means that $\hat{f}(m\Delta_x, n\Delta_y) = f[m, n]$ for every m and n. The reader can easily check that such a condition is guaranteed if $h(0, 0) = 1$ and $h(m, n) = 0$ for $|m| = 1, 2, ...$ and $|n| = 1, 2, ...$ Observe that Eq. (53) meets this condition, as the sinc function is null when evaluated at integer points, except at zero, whose value is one.

2. *Direct current (DC) preservation: DC-constant interpolator.*
   It is desirable that the DC component of the image is not amplified during the interpolation process. Mathematically, this condition is guaranteed if

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(d_x + m, d_y + n) = 1 \tag{54}$$

for any displacement $0 \leq d_x, d_y \leq 1$. This condition is also known as the partition of unity condition. It can be demonstrated that $H(0,0) = 1$ and that $H(u,v)$ is zero for $|u| = 1, 2, ...$ and $|v| = 1, 2, ...$, are necessary conditions for the partition of unity condition to hold.

6.1 Typical interpolator kernels

In this section, we briefly describe the most common interpolation methods in medical imaging. Most of the kernels $h(x,y)$ are assumed to be separable, that is, $h(x,y) = h_x(x)h_y(y)$. Hence, we focus on the 1D case, i.e., the shape of $h_x(x)$ or $h_y(y)$. For ease of notation, we call this 1D interpolation kernel as $h(x)$, no matter it refers to the x or y coordinate.

6.1.1 Windowed sinc

Implementation of the ideal sinc interpolator requires infinite number of samples. Hence, a straightforward approach is truncating the summand. Truncation of the summand is equivalent to multiplication of the sinc function with a rectangular function. In the frequency domain, this is equivalent to the convolution with a sinc function. As a result, truncation of the ideal interpolator produces ringing effects in the frequency domain. The rectangular function is a just a particular example of window or tapered function, which is a mathematical function that is zero outside of a given interval. More elaborated window functions can be considered to implement the sinc interpolator. Such window functions produce significant better results. An example of window function is the Blackman-Harris window function. For $N = 6$ samples, the Blackman-Harris window function $w(x)$ is given by

$$w(x) = \begin{cases} 0.42323 + 0.49755\cos\left(2\pi\frac{x}{N}\right) + 0.07922\cos\left(2\pi\frac{2x}{N}\right) & 0 \leq |x| \leq N/2 \\ 0 \quad \text{otherwise.} \end{cases} \tag{55}$$

A windowed sinc interpolator with the previous window can be shown to be a DC-constant interpolator. The profile of the filter is shown in Fig. 17.

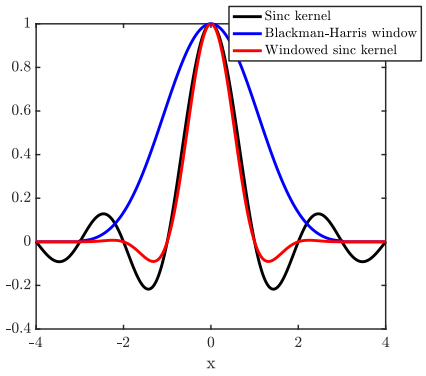


Figure 17: Windowed sinc kernel with Blackman-Harris window ($N = 6$)

Other typical windows functions are Hamming, Hanning or Lanczos.

6.1.2 Nearest Neighbor interpolation

The nearest neighbor interpolation is probably the simplest interpolation one can consider. The value $\hat{f}(x,y)$ at point (x,y) is given by the sample $f[m,n]$ whose associated discrete point (m,n) is the closest to (x,y). The way closeness is determined is given by the distance that is chosen.

In the conventional nearest neighbor interpolation, the l_1 norm or Manhattan distance is used. With this distance, the closest point (m, n) is simply the discrete point whose component m and n is the nearest integer to x and y, respectively. Nearest neighbor interpolation, which is a DC-constant interpolator and fullfills property 1 of Section 6, corresponds to the kernel $h(x)$ (Fig. 18) defined as

$$h(x) = \begin{cases} 1 & 0 \leq |x| < 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{56}$$

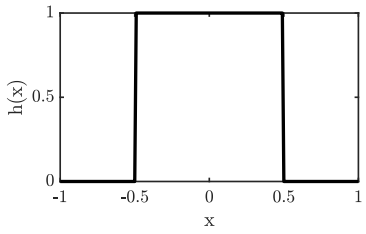


Figure 18: Kernel of nearest neighbor interpolation

Strong aliasing and blurring effects are prominent with nearest neighbor interpolation.

6.1.3 Linear interpolation

In a linear interpolation scheme, the values of direct neighbors (x and y) are weighted by their distance (absolute value) to the opposite point of interpolation [6]. The associated kernel $h(x)$ is a triangular function (Fig. 19) :

$$h(x) = \begin{cases} 1 - |x| & 0 \leq |x| < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{57}$$

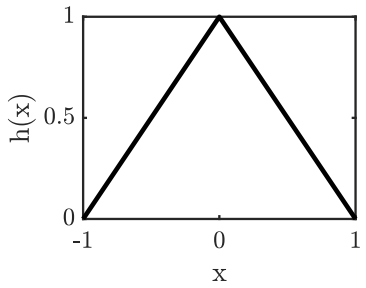


Figure 19: Kernel of linear interpolation

Linear interpolation is a DC-constant interpolation method and obviously meets $h(m) = 0$ for $|m| = 1, 2, ...$ Linear interpolation corresponds to a low-pass filter in the frequency domain. Consequently, it attenuates the high-frequency components, manifesting in blurring effects in the image domain. In the 2D case, bilinear interpolation is the common nomenclature.

6.1.4 Cubic interpolation

Cubic interpolation is defined through the following piece-wise cubic polynomial (Fig. 20) :

$$h(x) = \begin{cases} 1/2|x|^3 - |x|^2 + 2/3 & 0 \leq |x| < 1 \\ -1/6|x|^3 + |x|^2 - 2|x| + 4/3 & 1 \leq |x| < 2 \\ 0 & \text{otherwise.} \end{cases} \tag{58}$$

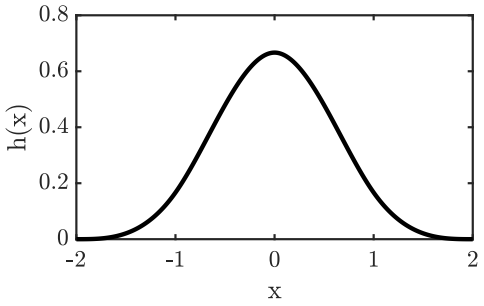


Figure 20: Kernel of cubic interpolation

Note that though cubic interpolation is DC-constant kernel, it does not meet the first property of interpolation kernels. Indeed, $h(0) \neq 1$ and $h(m) \neq 0$ $|m| = 1, 2, ...$ Consequently, cubic interpolation displays severe blurring effects. Nevertheless, it presents a favorable stop band response, and allows the attenuation of unwanted high-frequency noise in the image [6].

7 Image quantization

The pixels intensities are represented by bits. Since the number of bits are limited, the continuous range of intensity should be quantized, that is, should be transformed into a discrete set. This process is called image quantization. Thus, a quantizer transforms the continuous variable that corresponds to the intensity, let us say s, into a discrete variable. Such a discrete variable can only take a finite set of numbers $\{r_1, r_2, ..., r_L\}$. A quantizer is determined by the number of levels, that is, the discrete set of numbers that the variable can take, as well as by the decision levels. Given L decision levels $\{t_k, k = 1, ..., L+1\}$, the quantizer rule is defined as follows: if s lies in the interval $[t_k, t_{k+1})$, the quantized value is r_k. An example of a quantizer is shown in Fig. 21.

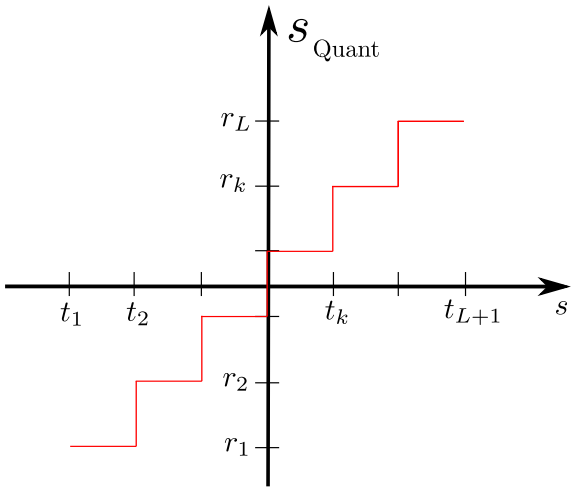


Figure 21: An image quantizer.

If the intervals $[t_k, t_{k+1})$ have the same length, the quantizer is said to be uniform. Clearly, an image quantizer with non-uniform intervals may have a better performance. Indeed, if intensity values u are likely to lie in a given range, it has sense to design a quantizer with much more decision levels /intervals in that region. As the error is proportional to the length of the intervals, the error is reduced in the selected region. If the probability density function

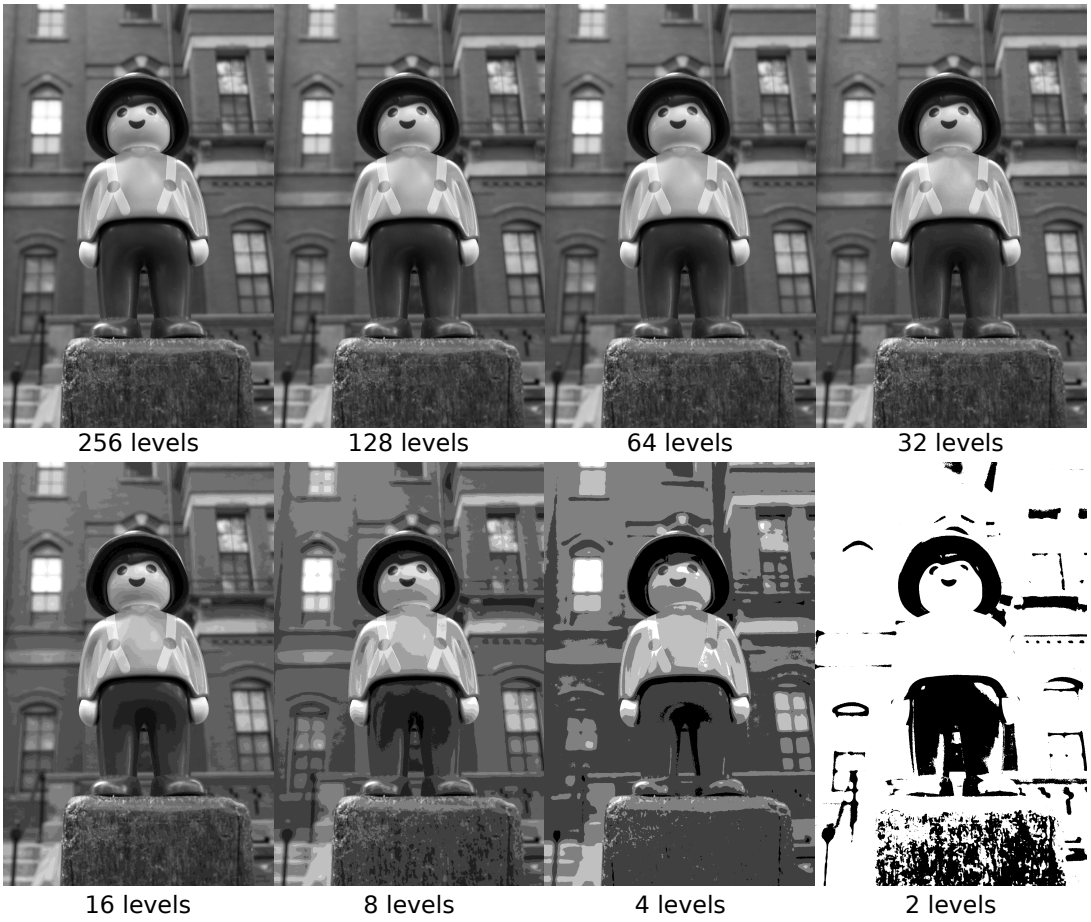


Figure 22: Example of quantization of a digital image. The same image is shown with different quantization levels, from 256 (8 bits) to 2 (1 bit).

of the random variable u is known, the optimal quantizer in terms of Mean Square Error can be derived analytically. This image quantizer is called Lloyd-Max quantizer. An example of a quantized image is shown in Fig. 22. For a more comprehensive treatment of image quantizers, we refer the reader to the excellent references [4].

References

[1] Beerends, R., ter Morsche, H., van den Berg, J., van de Vrie, E.: Fourier and Laplace Transforms. Cambridge University Press, 7th edn. (2003)

[2] Gonzalez, R.C., Woods, R.E.: Digital image processing. Pearson, 4th edn. (2018)

[3] Haykin, S., Veen, B.V.: Signals and Systems. Wiley, 2nd edn. (2002)

[4] Jain, A.K.: Fundamentals of digital image processing. Englewood Cliffs, NJ: Prentice Hall, (1989)

[5] Keys, R.: Cubic convolution interpolation for digital image processing. IEEE transactions on acoustics, speech, and signal processing **29**(6), 1153–1160 (1981)

[6] Lehmann, T.M., Gonner, C., Spitzer, K.: Survey: Interpolation methods in medical image processing. IEEE transactions on medical imaging **18**(11), 1049–1075 (1999)

[7] Lim, J.S.: Two-dimensional signal and image processing. Prentice Hall, Englewood Cliffs, NJ (1990)

[8] Marks, R.J.: Handbook of Fourier analysis & its applications. Oxford University Press (2009)

[9] Oppenheim, A.V., Willsky, A.S., Nawab, S.H.: Signals & Systems. Prentice-Hall, Inc, Upper Saddle River, NJ, USA (1996)

[10] Pratt, W.K.: Digital Image Processing. Wiley, New York, NY, USA, 3rd edn. (1991)

[11] Tolstov, G.P.: Fourier Series. Dover Publications Inc., Mineola, NY (1977)