Basic fundamentals of image processing

Having presented the physical foundations that give rise to the formation of radiological images, this chapter will introduce the initial basic concepts and the mathematical formulation necessary for an approach to digital image processing. Thus, the reader will be introduced to the initial concepts needed to understand the representation and basic operations of digital image processing.

Introduction to image processing

Image analysis and processing is performed by computers due to the complexity and number of calculations required. Therefore, while the mathematical formulation necessary for its implementation dates back several centuries, the real possibility of using it daily in clinical practice has only become possible in recent decades, thanks to advances in hardware technologies.

The proliferation of new equipment capable of performing millions of operations per second and its extension to everyday life and to all types of users has made digital image analysis and processing a major field of study. Today, this technology is even incorporated into all types of household equipment, such as digital cameras, scanners, and cell phones, among others.

Historically, the use of radiographic images for clinical diagnosis dates back almost entirely to the discovery of X-rays in 1895 (Roentgen). Even functional images derived from the emission of photons (X-rays) by radionuclides are already more ÿthan 90 years old (Heavesy & Seaborg, 1924). However, images acquired from X-ray films or directly from correct processing have not fully exploited their potential until the in vivo , so your incorporation of technology that allows them to be digitized.

The main reason for this "late appearance" of image processing has been due to the hardware requirements for both image processing and image representation in high-performance graphics systems.
In parallel with this development, the development of algorithms for processing has followed technological advances, achieving a high degree of sophistication and image manipulation in near real time.

The current variety of techniques, algorithms, and software and hardware developments used in digital image processing is beyond the scope of any single course. These courses leverage techniques initially developed based on foundational concepts for image analysis and incorporate a wide variety of concepts and notions from physics and mathematics, such as entropy and metrics.

This chapter will introduce the first notions and concepts for addressing the study of digital image processing, including image reading and representation formats, modification operations, transformations on tones and colors, and the generation of effects on regions of an image.

The interest of the following study can be condensed into two main objectives: a) to achieve a considerable improvement in image quality for specialist interpretation, and/or b) to obtain specific information for processing by means of calculation and analysis systems.

Of interest in this course will be the images produced by the interaction of ionizing radiation with matter for medical use, that is, those acquired by detectors.

X-rays that have[1] passed through - or originated from - a patient's biological tissue, forming a two-dimensional (2D) or three-dimensional (3D) image.

Image format and digital representation

Today, images constitute a language in themselves. Depending on different cultural factors, images are used to convey messages, symbols, and various types of information. Therefore, it is necessary to have a support for the digital representation of images that can then be modified to either modify the visual and symbolic content or obtain necessary information.

Bands in digital images

To acquire an image remotely, there must be some type of interaction between the object to be observed and the detector. In digital imaging, the different types of detectors depend on the type of electromagnetic radiation they are capable of detecting. Just as the information that can be obtained from an object also depends on the interaction of this radiation with the object. This particularity gives rise to the concept of "bands," where the electromagnetic spectrum is divided based on the types of interaction of the radiation with matter (see Fig. [fig2-1]), which defines everything from the objects to be analyzed to the detectors and materials that can be used.

All objects absorb, reflect, or emit energy quanta depending on their wavelength, intensity, and type of radiation. This type of radiation is defined by its physical properties within the electromagnetic spectrum. The human eye, on the other hand, is only capable of detecting electromagnetic energy in the visible light spectrum, while for

X-rays, ultraviolet, infrared or microwave radiation, construction is necessary of detectors that can collect this information, either digitally or analogically, to can be quantified and analyzed.

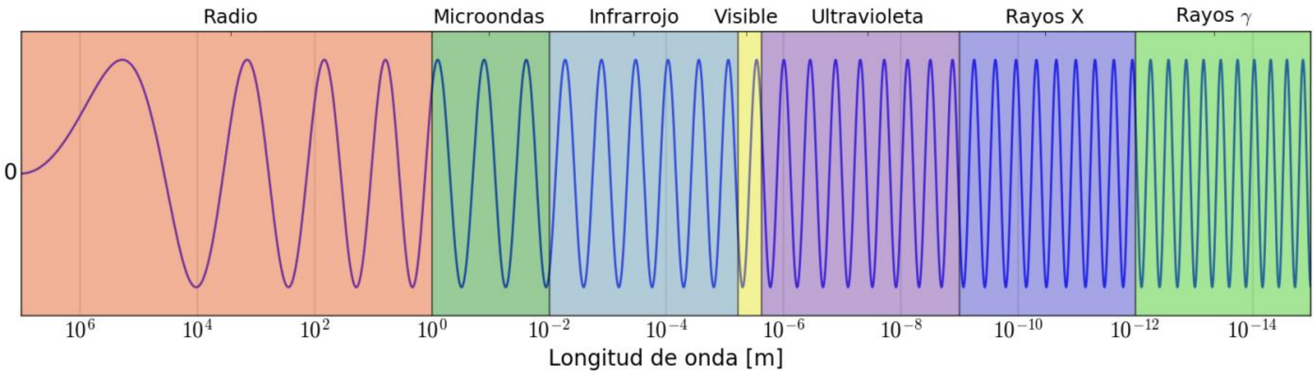

Figure 1: Qualitative diagram of the electromagnetic spectrum

Digital representation: bitmaps

A Bitmap is an elementary way to represent digital images as information in the hardware , specifically the memory, of a computer. It basically consists of forming arrays of elements (vectors, matrices, tensors) ordered in specific ways. In In general, for the typical case of 2D images, sorting is performed by rows of array elements (image pixels) assigning each one a value that determines "the color" in that position.

In the case of grayscale images, the value of the matrix element is a scalar; while for color images the value of each matrix element is a vector of three coordinates, each of which specifies "the degree of influence" of the colors red (Red "R"), green (Green "G") and blue (Blue "B"), so it is called RGB representation). There are other modes of color representation, such as CMYK (cyan, magenta, yellow and black).

Typically scales (which determine "dynamic ranges") are used in bits, and 2N denotes bits. Nÿ That is, for the most common 8-bit case, the scale is [0, 255], given that range is customarily defined as [0, 2 ÿ 1] the N .

The typical use of 8-bit is based mainly on two reasons. First, studies Biometrics show that the human eye is not sensitive enough to differentiate more of 256 intensity levels for a given color. In addition, the range of values for the Array elements determine the storage capacity requirements on the computer.

So for images in shades of gray, known as "one band" the range For the values of matrix elements (scalars) are color images, the [0, 255], while for values of matrix elements (3-coordinate vectors) assume values in ([0, 255], [0, 255], [0, 255]). However, it is also common to find normalized representations for color images, i.e. matrix elements in ([0, 1], [0, 1], [0, 1]) to determine RGB colors.

All colors in the visible range can be represented as RGB combinations, varying from black (0, 0, 0) to the target (255, 255, 255). Therefore, an RGB image is represented by a two-dimensional array of pixels, each encoded in 3 bytes being able to assume 256^3 different values of vector combinations, i.e. 16.8 millions of different colors, approximately.

Digital representation: vector images

Vector images are made up of defined outlines and fills. mathematically, vectorially, by means of equations that perfectly describe each illustration. In this way, it is possible to implement a process scaling without loss of quality. The that is typical in scaling, production, or reproduction on devices. By Hence, the importance of maintaining invariability. This characteristic is particularly relevance in cases where the illustrations contain marked areas with curved contours, since pixelation would imply a loss of resolution, as indicated in the figure.

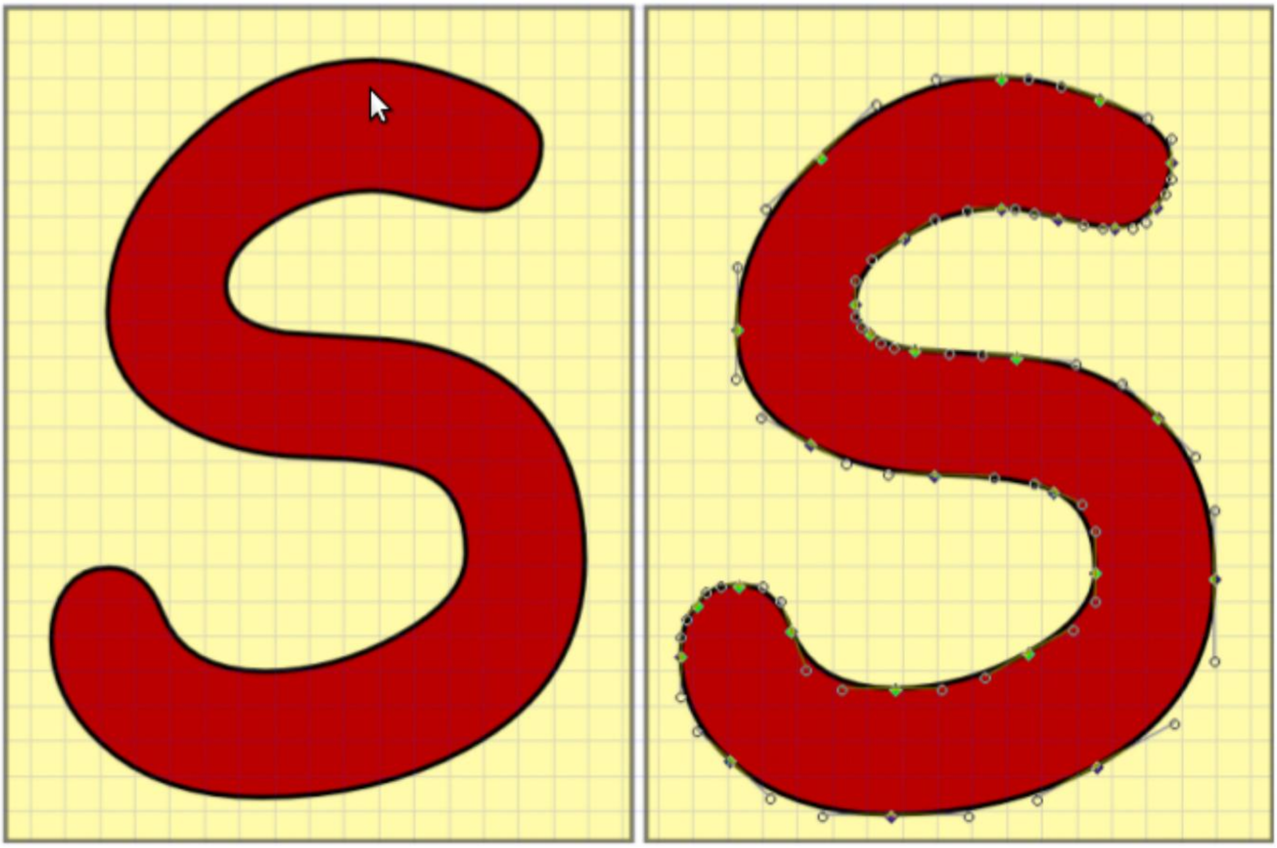

Figure 2: Image in vector representation (left) and in pixelated bitmap (right).

Modifying colors in images

It is possible to quantify the difference between two colors (in digital representation, trio values). RGB vector) calculating the distance, according to some type of metric, Euclidean for example, between the vectors that represent them.

The color represented by the vector C1 (R1, G1, B1) and the color represented by C2 (R2, G2, B2). So, in vector space, the distance given by: D(C1, C2) among these is

$$D(C1, C2) = \sqrt{(R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2}$$

For the particular case of single band images (gray tones) you can apply the same methodology described for RGB images with the simplification associated with the fact that that in color space, vectors in the vector direction represent the (1, 1, 1) different shades of gray.

Therefore, there is the equivalence that for any RGB type pixel (1, 1, 1) (R, G, B) if you know it projected onto the surface, the contribution of each shade of gray is obtained. Then, has:

$$Proy \cdot (R, G, B) \cdot (1, 1, 1) = R + G + B = |V| \, |\hat{n}| \cos(\theta)$$

where Proy is the projection, it is the vector that forms the point in the space of V (R, G, B) coordinates of the trio (vector representation), is the projection inverse $\hat{n}$ (1, 1, 1) forks the angle it forms with V $\hat{n}$.

From here it can be seen that $Proj = \frac{R+G+B}{\sqrt{3}}$ and it must be ensured that this value does not exceed 255, so it is usual to renormalize to obtain $Proj = \frac{R+G+B}{3}$

As an illustration of the general concepts presented on representations vector-bitmap, a very simple application case is proposed. If the objective in the edge detection of shapes in an image to obtain the resulting bitmap to highlight the black-and-white edges, you can proceed as follows: Move within the image comparing the color pixel to pixel of each one with its neighbor on the right and its neighbor below. Then, the following control (criterion) is performed: if when comparing results in a very large difference ("very large" is a parameter [2] or set of parameters pre-defined by the user, or automated in more elaborate cases) the pixel into consideration is part of the border and is assigned the color white, otherwise it is assigns the color black.

Histogram of an image

Given the digital representation of an image by means of the arrangement of rows by NM columns determine a matrix $M \times N$, in which the digital representation of bitmap will be given by the distribution function NM $f(m, n)$, for $n \in [0, N - 1]$ $m \in [0, M - 1]$, typically and are powers of 2, as already stated.

Machine Translated by Google

The histogram of an image $h(i)$, commonly referred to "image enhancement" pixels either "image characterization" It is a vector that accounts for the amount of within the image with a certain element value. That is, for a -bit image, we have: a

$$h(i) ÿ \sum_{m=0}^{Mÿ1} \sum_{n=0}^{Nÿ1} ÿ(f(n, m) ÿ i) ÿi ÿ [0, 2^a ÿ 1]$$

One of the generic techniques, which then diversifies into a very varied amount of specific processing methodologies, is the convolution method. Let (0, 0) $w(k, l)$ a arrangement $2 \times K + 1, 2 \times L + 1$, centered on the "origin" that coincides with $w(k, l)$ pixel central part of the image. It can be considered as a convolution mode as a kernel that applied to the image $f(n, m)$ result:

$$g(m, n) ÿ w(k, l) ÿ f(m, n) = \sum_{k=ÿK}^{K} \sum_{l=ÿL}^{L} w(k, l) ÿ f(m ÿ k, n ÿ l)$$

From this definition, a large number of specific methods can be introduced, Among which the transforms stand out, such as Fourier, Laplace, Radon, etc.

Image resolution

A priori , This concept has different meanings depending on the context in which it is used and could be defined, in a generic way, as the ability to represent or perceive details of an image. This is a concept present throughout the entire digital process, from capture or generation to representation, and affects (conditions) subsequent processing.

A useful definition is: the resolution of an image is the amount a typical measurement pixels that describe it. And is in terms of representation as well as the "pixels per inch" (ppi). Therefore, the quality of the size of the image depend on the resolution, which determines in turn the memory requirements for the graphic file to be generated.

Resolution, image size and file size

The three concepts are closely related and mutually dependent, although they are They refer to different characteristics and confusion should be avoided.

The size of an image is its actual dimensions in terms of width and height once printed, while file size refers to the amount of physical memory necessary to store the information of the digitalized image on any medium storage computing.

Certainly, the resolution of the image strongly conditions these two concepts, since The amount of pixels the scanned image is fixed and therefore as the size of the image increases, image resolution is reduced and vice versa.

As an example: doubling the resolution of a scanned image from 50 ppi to 100 ppi, the image size is reduced to a quarter of the original while dividing the resolution by 2. That is, it goes from 300 ppi to 150 ppi, obtaining an image with double of the original dimensions that represent four times its surface.

Reducing the image resolution while maintaining its size causes the elimination of pixels . So, a less accurate representation (description) of the image is obtained, so such as sharper color transitions. The file size that generates an image digitized is proportional, as expected, to the resolution, therefore, varying it implies modify the file size in the same way.

Contrast in an image

Conceptually, increasing or decreasing the contrast in an image consists, basically and Visually, by increasing or decreasing the slope of the straight line with a 45 degree slope degrees representing the grays (taking care not to exceed the limits 0-255) between input and output , as shown in figure [Fig2_2].

The transformation corresponding to the change in contrast is:

$$VO(m, n) = (VI(m, n)\ ÿ\ 2^{Yÿ1})\ \tan ÿ + 2^{Yÿ1}$$

where is AND the scale in bits, and are the ALIVE values of (m, n) valued at pixel input and output , respectively ÿ ; and the angle corresponds to the properties of the linear transformation of contrasts, specifically the slope (figure [Fig2_2]).

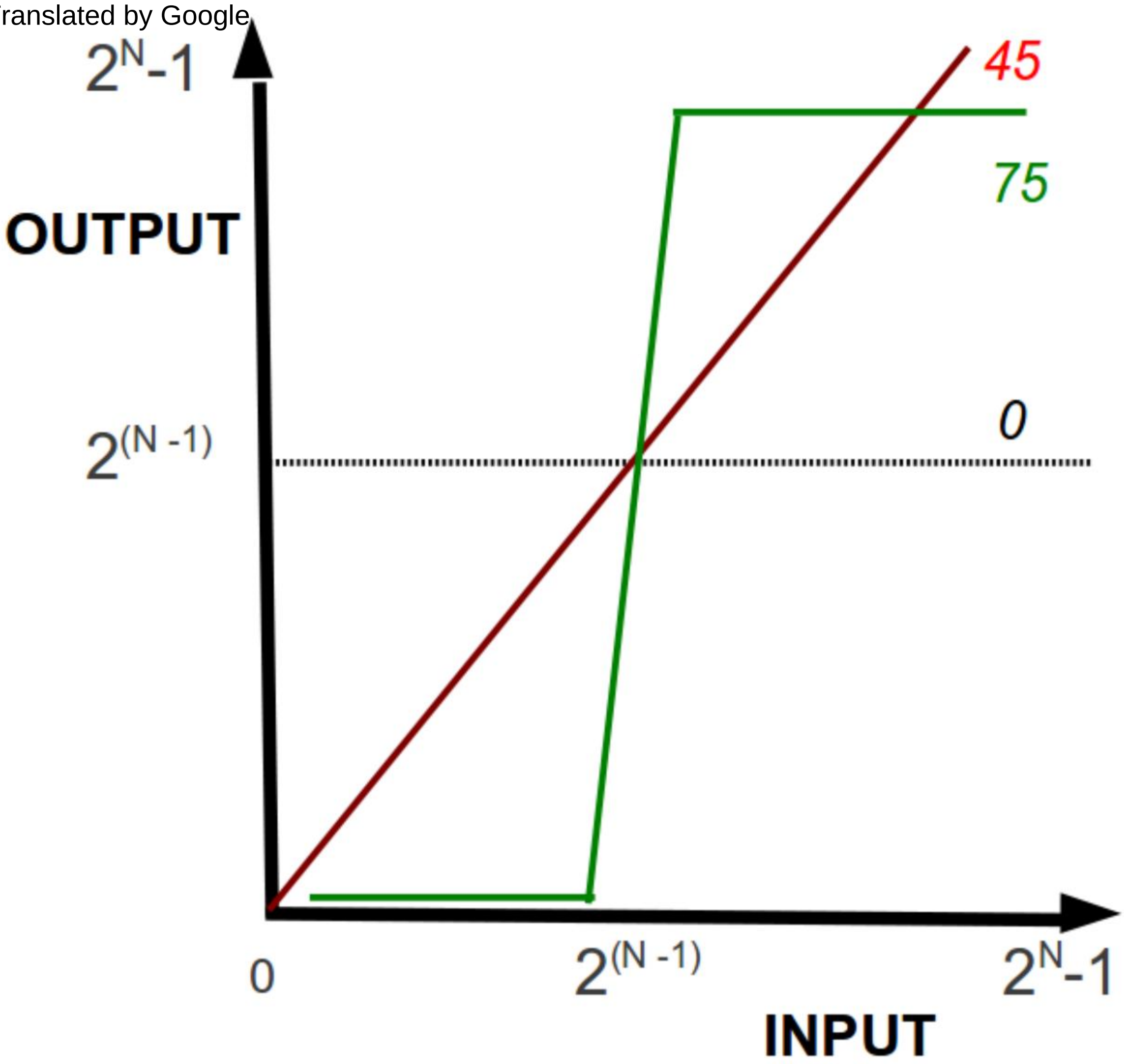

Figure 3: Representation of the changein input contrast between output. and

Physical link to the source of images

Images generated by electromagnetic radiation can be classified in generic mode according to the order of highest to lowest frequency.

Good heavens ÿ
nuclear medicine, astronomy observations.

X-ray
medical diagnostics and industry (quality control).

Ultraviolet band
Industrial inspection and biological microscopy.

Visible and infrared band
Various applications, photography.

Microwave
radar.

Radio waves
medicine (MRI) and some applications in astronomy.

Modifying an image

A property input can be modified in different ways, depending on the image/s that is/are modified.

In particular, some of the most frequent modifications are considered below.

Modifying colors or shades: ÿ correction

There is a wide variety of techniques and criteria for modifying the colors of an image. One of the most used and simple methodologies is the ÿ correction. , defined from:

$$VO(m, n) = (2^N ÿ 1) \left(\frac{VI(m, n)}{2^N ÿ 1}\right)^{\frac{1}{ÿ}}$$

where the index assumes ÿ values ÿ R .

Therefore, it turns out:

- For = ÿ 1 there is no correction.
- For values of there ÿ > 1 is a large correction in contrast for small values of color of input while a small correction in contrast for high values. The Brightness increases further for intermediate color values of ÿ < 1 input .
- For values of there is a small correction in contrast for low values of color of input , while a large correction in contrast for high values. The brightness decreases more for intermediate values of the color of input .

Image modification: inversion (flip)

Basically, this modification consists of a transformation that produces a "movement" of the column mn , column and row m (nmax ÿ n) + 1 row to , for nmax as the dimension in the direction of n .

That is to say,

$$Vflip(m, n) = VI (m,(nmax ÿ n) + 1)$$

where Vflip is the output matrix that corresponds to the investment transformation.

Image modification: reflection (mirror)

Basically, this modification consists of a transformation that produces a "movement" of row nmn , column to the row and column (mmax ÿ m) + 1 , for mmax as the dimension in the direction of m .

That is to say,

$$V_{mirror}(m, n) = V_I(m, (n_{max} - n) + 1)$$

where V_{mirror} is the output matrix corresponding to the reflection transformation.

Image modification: interpolation

From a sampling different from input intensity values at points can be estimated those points where the value is known. Among other techniques, the following stand out: methods of re-sampling .

Thus, different criteria are used to determine the values $VO(k, l)$ for pixels (k, l) where the input VI is not known:

- Nearest neighbor interpolation.
- Bilinear interpolation.
- Bicubic interpolation.

The nearest neighbor interpolation technique (based Nearest neighbor interpolation input) this on superimposing the 2D array onto the output 2D (k, l) array) calculating the value for the pixels according to known values $VI (i, j)$, using an average (which can (quantified in different ways) of the equidistant nearest neighbors. However, It can be seen that this technique has some undesirable effects.

The linear interpolation technique considers the 4 for the pixels closest to $V (k, l)$ interpolation. An average is taken between these 4 values to determine the value unknown of the pixel (k, l) . The image . output It is "softer" than in the case of the technique Nearest neighbor interpolation However, it can cause the image to appear somewhat "fuzzy."

So, the starting values are: pixels (k, l) , for which it is not known $VI (k, l)$ are obtained at

$$VO(k, l) = (1 - \alpha)(1 - \beta)\, VI (i, j) + \alpha(1 - \beta)\, VI (i + 1, j) +$$
$$(\alpha \beta - 1)\, VI (i, j + 1) + \alpha \beta\, VI (i + 1, j + 1)$$

where $\alpha = k - i$, $\beta = l - j$, $i = \mathrm{floor}(k)$ and $j = \mathrm{floor}(l)$ [3].

For its part, the bicubic interpolation technique is the most common interpolation algorithm. used. Consider the 16 pixels closest to each pixel (k, l) whose values must be determined by interpolation. It locally approximates the value (the gray level) in the image original by means of a bicubic polynomial surface. It turns out to be, of the chemical techniques described, the optimum when considering the balance between computation time and performance .

The implementation of this method can be carried out by processing the block $B(k, l)$, focused on the pixel (k, l) , whose dimensions correspond to the dimensions of the mask (16 pixels in a 5 × 5 arrangement):

$$B(k, l) = ÿ (k ÿÿ(l ÿ q^{(k,l)}_{i,j} k^{ÿ})^{ÿ} l^{ÿ})^{j}$$

i=0 j=0, 3 3

$$k^{ÿ}, ÿ [k ÿ 2, k + 2] \& l^{ÿ} ÿ [l ÿ 2, l + 2]$$

where the coefficients $q_{i,j}$ must be determined. Or,

$$VO(k, l) = h(k)\ h(l)$$

where the interpolation function h is defined, piecewise, as follows:

$$h(p)\ ÿ 1 ÿ |p| +^{2} |p ÿ|p|_{l}^{3} < 1$$
$$h(p)\ ÿ 4 ÿ 8 |p| +^{2} 5 |p ÿ |p^{2} ÿ1 ÿ |p^{3}| < 2$$
$$h(p)\ ÿ 0 ÿp ÿ 2$$

Qualitative comparison of performance of algorithms interpolation

- Nearest neighbor interpolation: The position error is, at most, average and is noticeable on pixel , objects with straight boundaries where a jump effect appears. after this transformation.
- Linear Interpolation: Generates a slight decrease in resolution due to blurring (blurring) intrinsic to the average value calculation mode, but decreases the effect of jump presented by the nearest neighbor algorithm.
- Bicubic Interpolation: It does not present the problem of the jump effect while generating a minor blurring .

Basic relationships between pixels

The most immediate basic relationship between pixels is the distance D between two pixels (m, n) and $(m^{ÿ}, n^{ÿ})$.

The axioms for defining a metric or distance function are as follows: pixels D require the

- $D(kk^{ÿ}, l l^{ÿ})\ ÿ 0$ with $D(kk^{ÿ}, ll ÿ) = 0 ÿ k = k^{ÿ}\ l = l^{ÿ}$
- $D(kk^{ÿ}, l l^{ÿ}) = D(k, k^{ÿ} l l^{ÿ})$
- $D(kk^{ÿ}, l l^{ÿ})\ ÿ D(k, k^{ÿ} s_{s}) + D(s, l l^{ÿ} s^{ÿ})$

Based on these conditions, various metrics can be defined. These include:

$$D(kk^{ÿ}, ll ÿ)\ ÿ (k ÿÿkÿÿÿÿÿÿÿ)^{2} ÿÿÿÿÿ(l ÿÿÿl)^{ÿ 2} ÿ$$

$$D_{4}(k, l)\ ÿ |k ÿ l^{ÿ}| + |l ÿ | k\ k^{ÿ}\ l^{ÿ}$$

\begin{aligned}
D_{8}(k \; k', l \; l') \equiv \max \left(\lvert k - k' \rvert , \lvert l - l' \rvert \right)
\label{EqXXXIV}\end{aligned}

The Euclidean definitions, D4 and D8 for the distance between pixels does not depend on adjacencies but exclusively of spatial coordinates (k, l) .

It can be seen from the definitions of the metrics that the condition D(kk ÿ ,ll ÿ) ÿ. R determines a circle centered on (k, l) for the Euclidean meter, a rhombus for the meter D4 and a square for the metric D8 .

Operators on images

To operate on images, tools based on operations can be used. linear algebra matrix operations and oriented array operations f(m, n) pixel to pixels . H it's a arbitrary operator on an image whose matrix representation is if it satisfies:

$$H [f(m, n)] = g(m, n)$$

Furthermore, H it is a linear operator if:

ÿjfj ÿ aj fj
H [ÿ (m, n)] = H [(m, n)] j
j

An important application of the linearity properties of operators on images is the description of images r(m, n) g(m, n) as "original" contribution (f(m,n)) and noise (random):

$$g(m, n) = f(m, n) + r(m, n)$$

The noise image is of the correlated random if the values of pixels of r(m, n) They are not random type and with expectation 0.

Averaging noisy NTot images random the average image is obtained given by: ÿgÿ

ÿgÿ(m, n) = 1 / NTot ÿ (j=1 to NTot) gj (m, n)

The application of the central limit theorem establishes that the average image ÿgÿ(m, n) ÿ f(m, n) ("original" image) for NTot ÿ ÿ .

Another useful application of linear operators is mask subtraction [4] M(m, n) f(m, n) to the original image :

$$g(m, n) = f(m, n) ÿ M(m, n)$$

Adding and differentiating images

To illustrate the operations, 8-bit images are used.

Therefore, the values of the image resulting from the addition of two images vary in $[0, 510]$. While the image values resulting from the difference of two images vary in $[ÿ255, 255]$.

The adequacy (f_A) of the image values resulting from the addition/difference of two images are done as follows:

$$f_A(m, n) = \text{round}\left[(2^N ÿ 1)\frac{f(m, n) ÿ \min[f(m, n)]}{\max[f(m, n) ÿ \min f(m, n)]}\right]$$

For N-bit type images.

Operations on pixels

The introduction of spatial operations carried out on the image values allows: pixels of

- Operations of a pixel.
- Neighborhood Operations.
- Geometric transformations.

Operations of a pixel

The value of a is modified pixel individually in the original image $f(m, n)$, giving as a result $g(m, n)$ given by:

$$g(m, n) = T(f(m, n))$$

so that the image value is modified by the transformation T. This concept is applies, for example, to determine "the negative"

Neighborhood operations

Sea $C(M, N)$ a set of (m, n) pixels $M := [m_{min}, m_{max}]$ and $N := [n_{min}, n_{max}]$) around (neighbors) to the pixel.

From this type of neighbor operations, for example, the average value can be calculated in a rectangular environment ($M \times N$) of a pixel of interest [5]. It turns out:

$$g(m, n) = \frac{1}{MN} \sum_{(i, j) ÿ C(M, N)} f(i, j)$$

The geometric transformations of an image can be obtained from $f(m, n)$ T geometric transformation of spatial coordinates: the value of the is assigned the value of a pixel (m, n) HE pixel (i, j) .

Due to the discrete nature of image representation, the interpolation process to obtain the values as a result of applying I pixels transformation . T

One of the main categories of transformation operators are the transformations called affine, which include translations, rotations, scaling, reflections and projections, among others.

Some examples of transformation operators are:

- Rotation: $T_{Rot} = \begin{bmatrix} \cos(ÿ) & \sin(ÿ) \\ ÿ \sin(ÿ) & \cos(ÿ) \end{bmatrix}$
- Scaling: $T_{Esc} = \begin{bmatrix} e_i & 0 \\ 0 & e_g \end{bmatrix}$
- Translation: $T_{Tra} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ you & t_j & 0 \end{bmatrix}$

Dicrete transforms: The Fourier transform

From a general point of view, transforms constitute spatial operations on an original image, represented $f(m, n)$ in the spatial domain (which refers to the coordinates (m, n)) and a resulting image that processes $F(m, n)$ the values of the geometric plane. pixels in

There are different ways of representing the image, in terms of the central space. representation:

1. the image is represented $f(m, n)$ by a matrix F(mÿ, nÿ) $M \times N$ pixels (m, n) discreet.
2. The image is represented by a matrix of variables $M \times N$ transformed $(mÿ, nÿ)$.

As introduced qualitatively ([EqXXV] and [EqXXVI]), a linear transformation of an original image means $f(m, n)$

$$F(mÿ, nÿ) = ÿ_{m\ ÿn} f(m, n)\, k(mmÿ, nnÿ)$$

where is k the kernel of the transformation.

Machine Translated by Google

The direct transform (forward of f(m, n)) becomes F(mÿ, nÿ) , and the transformed inverse () of inverse F(mÿ, nÿ) becomes f(m, n) . Therefore, the equivalent of the expression ([EqXLIII]) is:

$$f(m, n) = ÿ_{mÿ} ÿ_{nÿ} F(mÿ, nÿ)\ k(m ÿ m, n ÿ n)$$

where k(m ÿ m, n ÿ n) is the kernel of the inverse transformation.

This enables the possibility of operating in the transform space. That is:

$$f(m, n) \xrightarrow{T} F(mÿ, nÿ) \xrightarrow{EITHER} G(mÿ, nÿ) \xrightarrow{T^{ÿ1}} g(m, n)$$

where and represent the forward and inverse transforms, respectively. is a arbitrary operator. T Tÿ1 EITHER

Of particular importance is the property of saying: kernels of being separable into variables. It is

$$k(mmÿ, nnÿ) = k_{(m,n)}(m, n)\ k_{(mÿ,nÿ)}(mÿ, nÿ)$$

The 2D two-dimensional discrete Fourier transform is defined from the transformations: kernels of

$$k_{TF}(mmÿ, nnÿ) = e^{ÿ2ÿi\left(\frac{mmÿ}{M} + \frac{nnÿ}{N}\right)}$$

$$k_{(TF)^{ÿ1}}(mmÿ, nnÿ) = \frac{1}{MN} e^{2ÿi\left(\frac{mmÿ}{M} + \frac{nnÿ}{N}\right)}$$

Therefore, the direct discrete transform operation (TF) and inverse ( ) (TF)ÿ1 result:

$$F(mÿ, nÿ) = ÿ_{m}^{Mÿ1} ÿ_{n}^{Nÿ1} f(m, n)\ e^{ÿ2ÿi\left(\frac{mmÿ}{M} + \frac{nnÿ}{N}\right)}$$

$$f(m, n) = \frac{1}{MN} ÿ_{mÿ}^{Mÿ1} ÿ_{nÿ}^{Nÿ1} F(mÿ, nÿ)\ e^{2ÿi\left(\frac{mmÿ}{M} + \frac{nnÿ}{N}\right)}$$

Whose analogue in continuous spaces is:

$$F(u, v) = TF[f(x, y)] = ÿ_{ÿÿ}^{ÿ} ÿ_{ÿÿ}^{ÿ} f(x, y)\ e^{ÿ2ÿ\backslash i(u x+vy)}\ dxdy$$

$$f(x, y) = (TF)^{ÿ1}[F(u, v)] = \frac{1}{4ÿ2} ÿ_{ÿÿ}^{ÿ} ÿ_{ÿÿ}^{ÿ} F(u, v)\ e^{2ÿi(u x+vy)}\ dudv$$

The expression ([EqXLIX]) for the Fourier transform can be interpreted, leaving aside momentarily problems of existence and uniqueness, as a sum of exponentials complex with weights for the terms, where the variables mÿ nÿ and represent the frequencies in the transform domain.

The value of the transformation in (mÿ, nÿ) F(mÿ, nÿ) contributes through F(mÿ, nÿ) and $^{2ÿi(u\ x+vy)}$ and it can be seen, since f(m, n) is a real function, that F(mÿ, nÿ) = F (mÿ, nÿ)^{ÿ} , where ÿ indicates the complex conjugate.

As an example, figures [Fig2_3], [Fig2_4] and [Fig2_5] present results of applying the Fourier transform of the original image for different cases f(m, n)

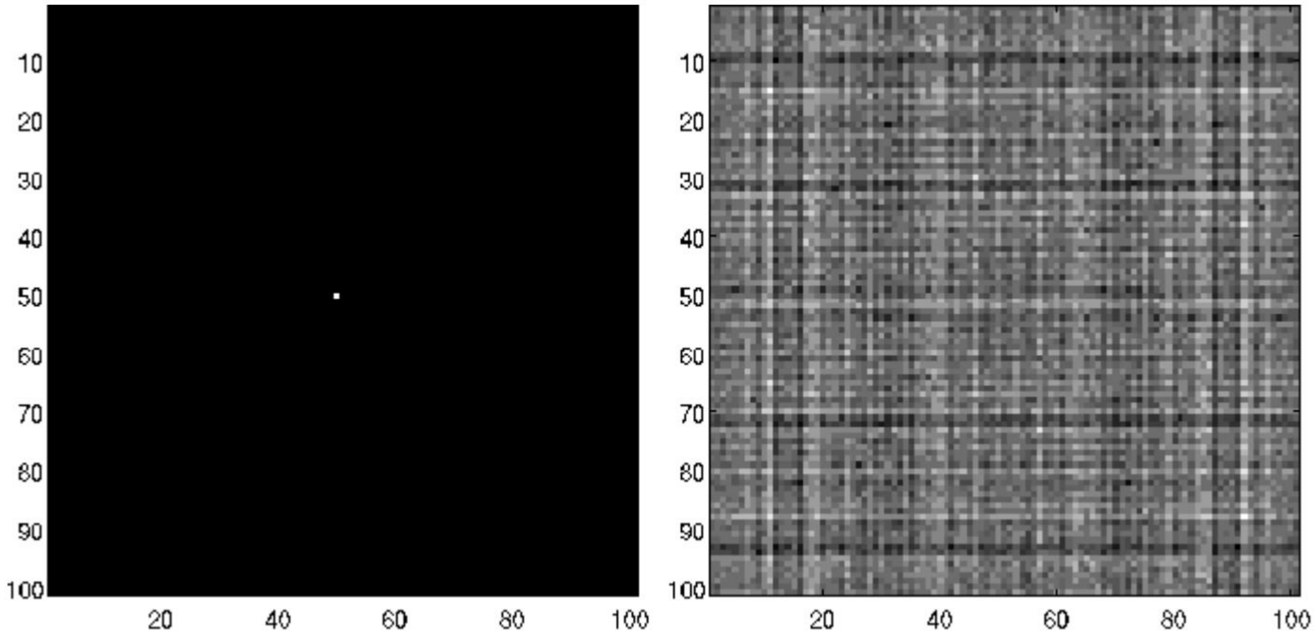

Figure 4: Example of transformation of Fourier: f(m, n) = 0 ÿ(m, n) ÿ (51, 51) ; m, n ÿ [1, 101] obtained with MatLab platform [*].

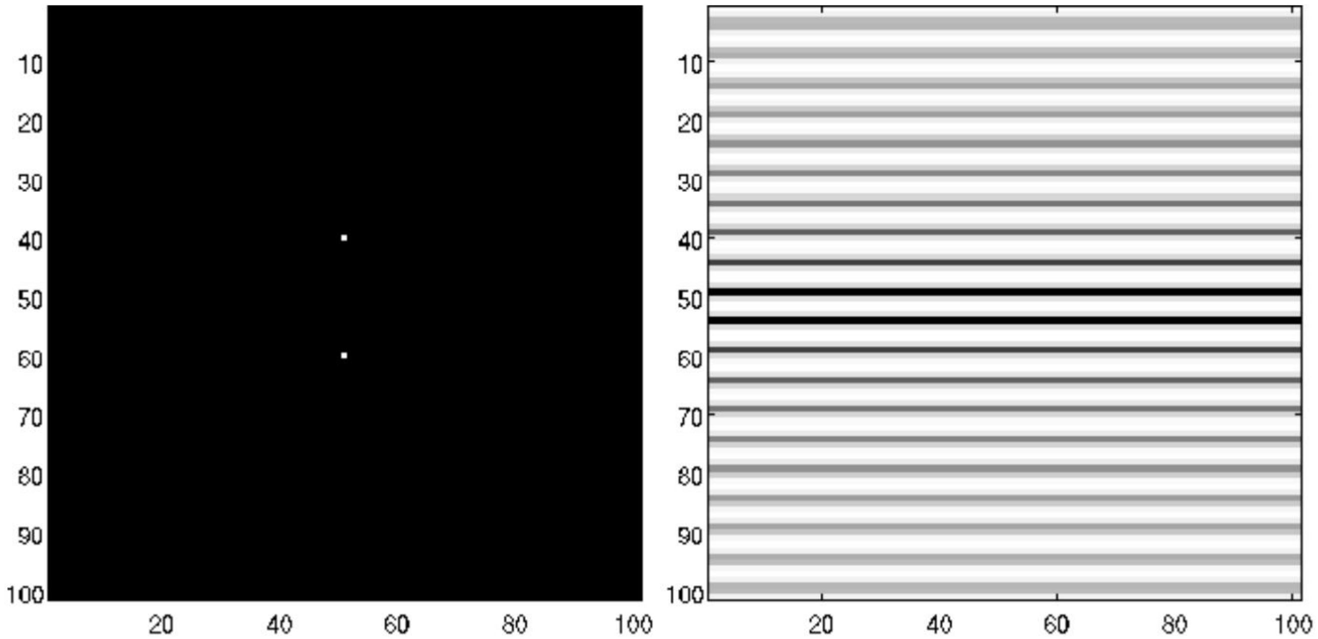

Figure 5: Example of transformation of Fourier: f(m, n) = 0 ÿ(m, n) ÿ (40, 51) , ÿ ÿ (60, 51) m, n ÿ [1, 101] obtained with MatLab platform.

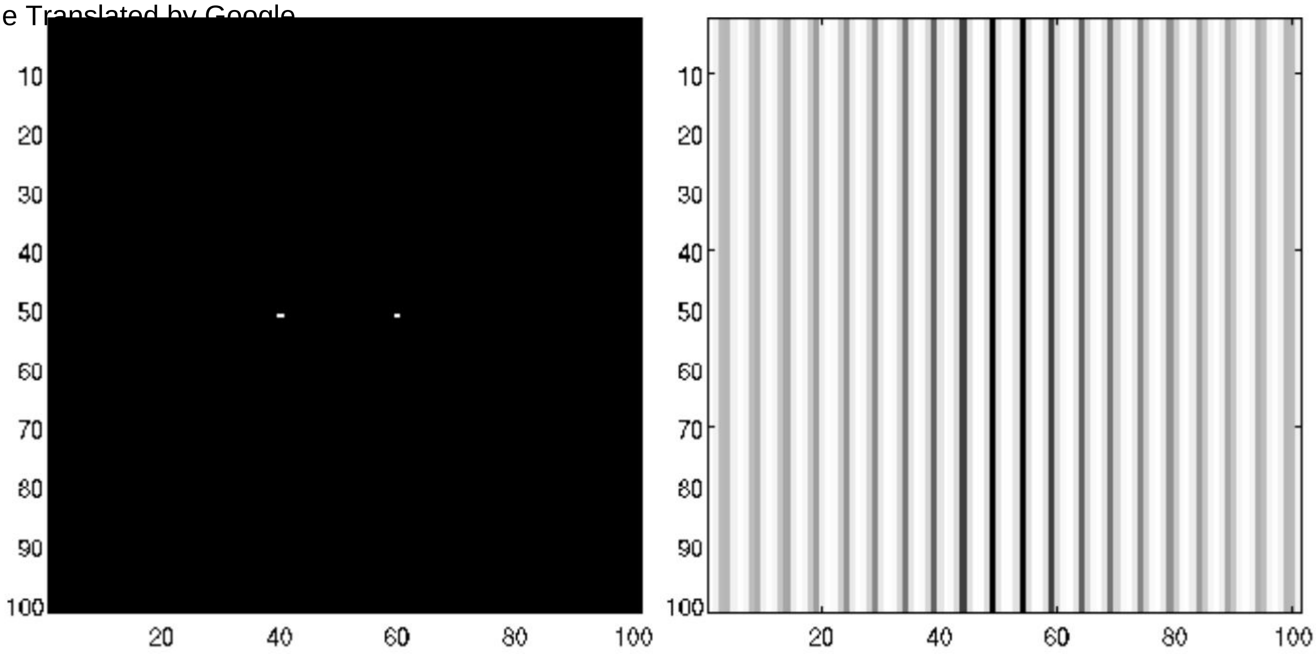

Figure 6: Example of transformation of Fourier: $f(m, n) = 0$ ÿ(m, n) ÿ $(51, 40)$, ÿ ÿ $(51, 60)$; m, n ÿ $[1, 101]$ obtained with MatLab platform.

Filters

From the definitions introduced by the expressions ([EqL]) and ([EqLI]) it is possible perform filtering processes both in the special domain of the original image $f(m, n)$ as in the frequency domain of the transform $F(m$ÿ$, n$ÿ$)$.

A significant feature, which in fact represents one of the main advantages of the transform spaces, is that the filtering operation is performed by means of a multiplication of transforms; while the operation in coordinate space means a convolution denoted by the symbol . ÿ By virtue of the convolution theorem, we have:

$$f(m, n) \text{ ÿ } g(m, n) = \text{ÿ} f(m, \text{ÿ}n) \, g(m \text{ ÿ } k, n \text{ ÿ } l) \, dk \, dl$$

Applying the definition of Fourier transform, we obtain:

$$F_{f,g} (m\text{ÿ}, n\text{ÿ}) \text{ ÿ } TF[f(m, n) \text{ ÿ } g(m, n)] =$$
$$TF[f(m, n)] \, TF[g(m, n)] = F(m\text{ÿ}, n\text{ÿ}) \, G(m\text{ÿ}, n\text{ÿ})$$

For a given original function and its corresponding Fourier transform $f(m, n)$ $F(m$ÿ$, n$ÿ$)$, in reference to the expression ([EqLIII]) the operator $G(m$ÿ$, n$ÿ$)$ is defined as a linear spatial filter either transfer function of filter.

Then, the image resulting from the filtering process is obtained by applying the $h(m, n)$ inverse transform:

$$h(m, n) = TF^{\text{ÿ}1}[F_{f,g}(m\text{ÿ}, n\text{ÿ})]$$

The filter is determined by means of the transfer function or by j(m, n) answer of impulse defined from:

$$j(m, n) = Ff,g\,(m, n) = ÿ(m, n)\ ÿ\ j(m, n) =$$

$$ÿ\ ÿ(m, n)\ j(k\ ÿ\ m, l\ ÿ\ n)\ dk\ dl$$

It turns out that j(m, n) It is an intense filtering in terms of the Dirac function. ÿ

For computational purposes, the discrete Fourier transform can be obtained, in a manner analogous to the expression ([EqXLVIII]) operating:

$$F(m, n) = TF[f(m, n)] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)$$

$$[\cos(2ÿ(\frac{m m}{M} + \frac{n n}{N})) + i \sin(2ÿ(\frac{m m}{M} + \frac{n n}{N}))]$$

$$f(m, n) = TF^{-1}[F(m, n)] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(m, n)$$

$$[\cos(ÿ2ÿ(\frac{m m}{M} + \frac{n n}{N})) ÿ i \sin(2ÿ(\frac{m m}{M} + \frac{n n}{N}))]$$

Due to the discrete nature of the sampling space intrinsic to digital processing of images, the relationship between spatial and frequency domains is determined by means of:

$$ÿm = \frac{1}{M\, ÿm}$$

$$ÿn = \frac{1}{N\, ÿn}$$

It should be noted that, for processing convenience, in the case of images "square" for which $N = M$, the expression for the Fourier transform is redefined, multiplying the expression ([EqLVI]) by , that is: N

$$F(m, n) = TF[f(m, n)] = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n)\ e^{ÿ2ÿ i (\frac{m m + n n}{N})}$$

$$f(m, n) = TF^{-1}[F(m, n)] = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m, n)\ e^{2ÿ i (\frac{m m + n n}{N})}$$

The complex spectral component of $F(m, n)$ determines module and phase, respectively, given by:

$$|F(m, n)| = \sqrt{[R(F(m, n))]^2 + [I(F(m, n))]^2}$$

$$ÿ(m, n) = \arctan[\frac{I(F(m, n))}{R(F(m, n))}]$$

where and GO represent the imaginary and real components, respectively. When filtering a original square image f(m, n) dimensions using a filter N × N j(m, n) of dimensions will give you a resulting image L × L g(m, n) of dimensions N + L ÿ 1 × N + L ÿ 1 .

The properties of the Fourier transform allow us to easily identify the most relevant operators of digital processing, such as:

i(f(m, n)e 2ÿ (m ÿ m0 + n ÿ n0)/N) ÿ F(ÿ mÿ mÿ 0 , n ÿ ÿ) n ÿ 0

f(m ÿ m0, n ÿ n0) ÿ F(m ÿ , n ÿ) and ÿ2ÿ i((mÿ m0+n ÿ n0)/N)

That is, a translation to the point (m0, n0) is identified with the shift of the origin of the plane of the frequency domain at the point (m ÿ 0 n ÿ 0) .

In polar coordinates m = ÿ cos ÿ n = ÿ sin ÿ m = ÿ cos ÿ = ÿ sin ÿ ÿ n ÿ , the original and transformed images f(m, n) F(m ÿ , n ÿ) are expressed as f(ÿ, ÿ) and F(ÿ, ÿ) .

Applying the definition of Fourier transform, we have:

f(ÿ, ÿ + ÿ0) ÿ F(ÿ, ÿ + ÿ0)

That is, the rotation of the original image (or) by an angle f(ÿ, ÿ) is linked to f(m, n) ÿ0 rotation of the same angle in the resulting image (or F(ÿ, ÿ) F(m , a ÿ n ÿ)).

Figures [Fig2_6] and [Fig2_7] show examples of application of rotation operators, in coordinate and frequency domains, respectively.

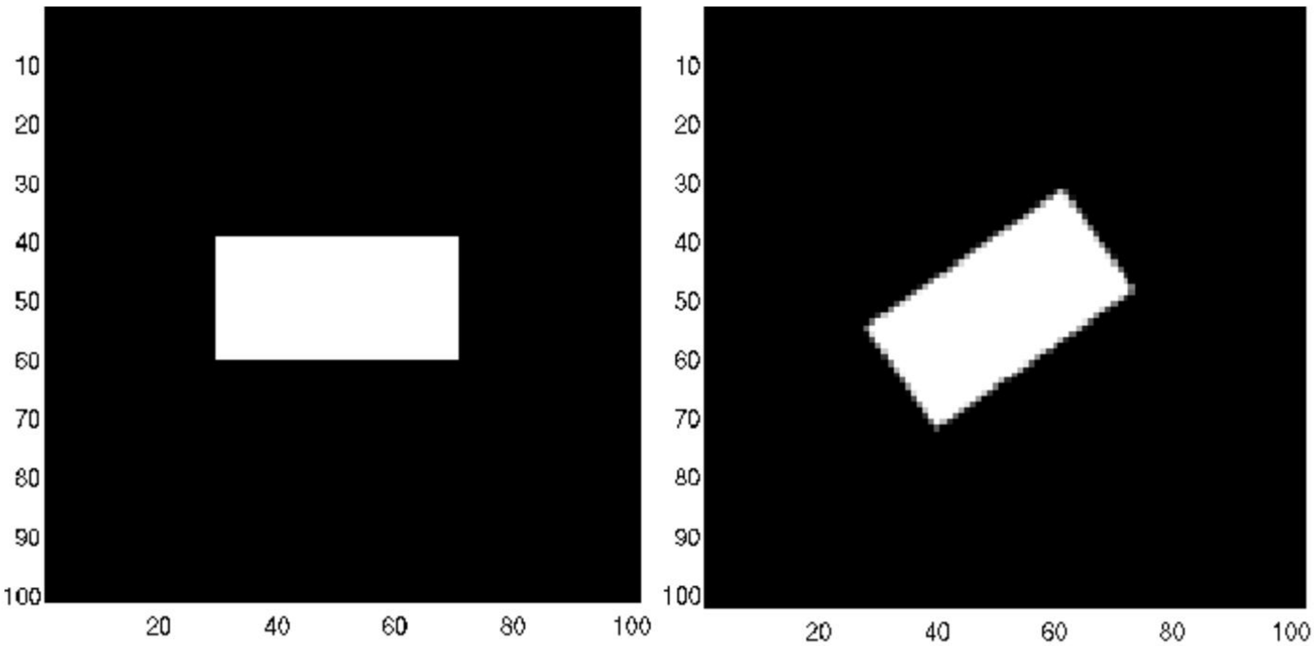

Figure 7: Rotation of 35 deg from an original image f(m, n) f(ÿ, ÿ) (edited)) obtained with MatLab platform.

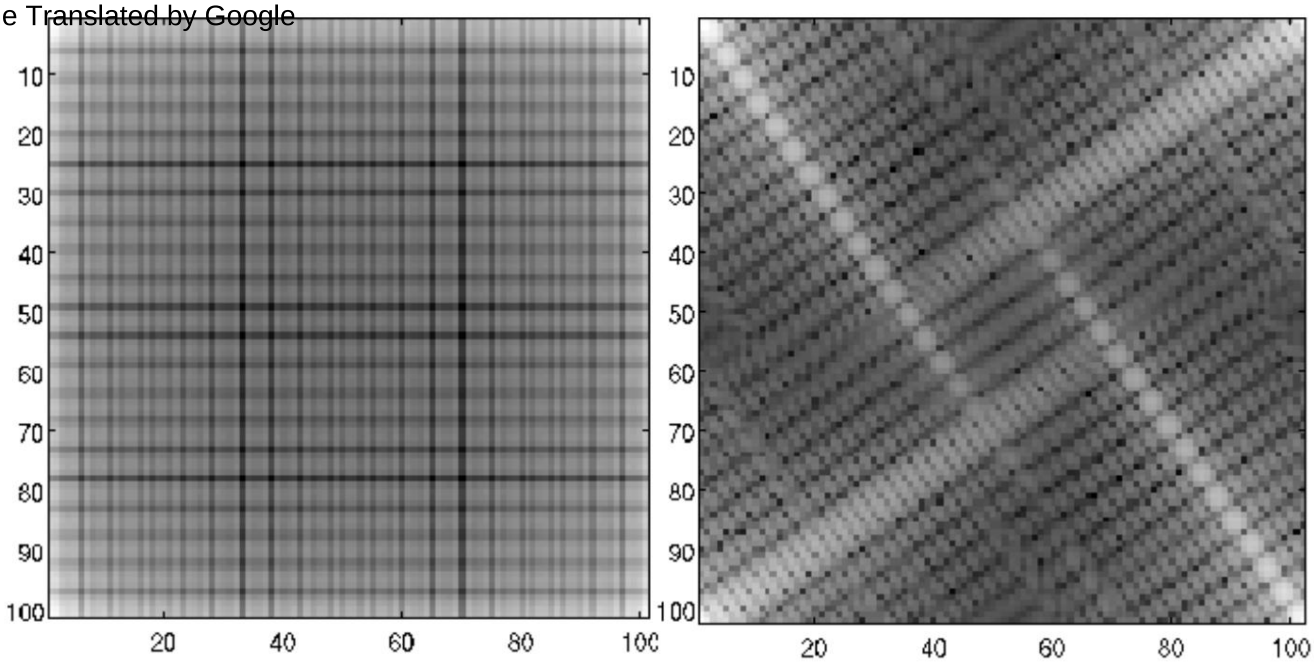

Figure 8: Rotation of 35 deg from an image) transform domain in $F(m ,^{ÿ} n^{ÿ})$ (either $F(ÿ, ÿ)$ obtained with the MatLab platform.

Scaling Operator

The definition of Fourier transform implies that

$TF[fA (m, n) + fB(m, n)] = TF[fA (m, n)] + TF[fB(m, n)]$ but

$TF[fA (m, n) ÿ fB(m, n)] ÿ TF[fA (m, n)] ÿ TF[fB(m, n)]$.

However, for scalars and we have: ÿ ÿ

$$ÿ f(m, n) ÿ ÿ F(m ,^{ÿ} n^{ÿ})$$

$$f(ÿm, ÿn) = F(ÿ , ÿ) \frac{1}{|ÿÿ|} mÿ n^{ÿ}$$

Calculation of averages

The average value ÿFÿ It is obtained from:

$$ÿfÿ = \frac{1}{N 2} \sum_{ÿm=0}^{Nÿ1} \sum_{ÿn=0}^{Nÿ1} f(m, n)$$

In particular, taking $F(m =^{ÿ} 0, = 0)^{n^{ÿ}}$ in the expression ([EqLIX] (59)) we obtain $F(0, 0) = \frac{1}{N} ÿNÿ1 m=0 ÿNÿ1 n=0 f(m, n)$. Therefore, the average value can be calculated directly from:

$$ÿfÿ = F(\frac{1}{N} mÿ = 0, = 0) n^{ÿ}$$

Calculation of derivative operators: Laplace operator

The Laplacian of an original image is given by: $ÿ2 f(m, n)$

$$ÿ 2f(m, n) ÿ + \frac{ÿ^2}{ÿm2} \frac{ÿ^2}{ÿn2}$$

Applying the definition of Fourier transform, a useful expression is obtained for the calculation of the Laplacian:

$$TF[ÿ^2 f(m, n)] = ÿ(2ÿ [(+ (mÿ)^2 \quad ^2 \quad n^ÿ)^2] F(, mÿ n \quad ^ÿ)$$

Band-pass filters

Abrupt transitions, such as edges and contours, in an original image correspond $f(m, n)$ HE to high frequencies in the transform domain.

Smoothing filters

This feature can be exploited to implement filtering methods for smoothing. operating in the frequency domain.

It is possible to suppress frequencies below or above predetermined values of so that smoothing effects are produced according to requirements.

Ideal high-pass filters

It consists of using the expression ([EqLIII]) with the transfer function $GP,A (m , ^ÿ \; n^ÿ)$ defined by:

$$GP,A (m , ^ÿ \; _n{}^ÿ) = \begin{cases} 0 & D(mÿ , n ÿ) ÿ Dmax \\ 1 & D(m , ^ÿ \; _n{}^ÿ) > Dmax \end{cases} .$$

For a maximum distance value) $Dmax$ as a threshold for distance (regardless of the metric), $D(m , ^ÿ \; n^ÿ)$ is the distance to the origin of frequencies $(m^ÿ = 0, = 0)n^ÿ$.

In the case of the Euclidean metric $D(mÿ , n ÿ) = (mÿ + (ÿ) 2 nÿ) ÿÿÿÿÿÿÿÿÿÿ2$, the filter is represented by a circle of radius Dmax as shown in figure [Fig2_8].

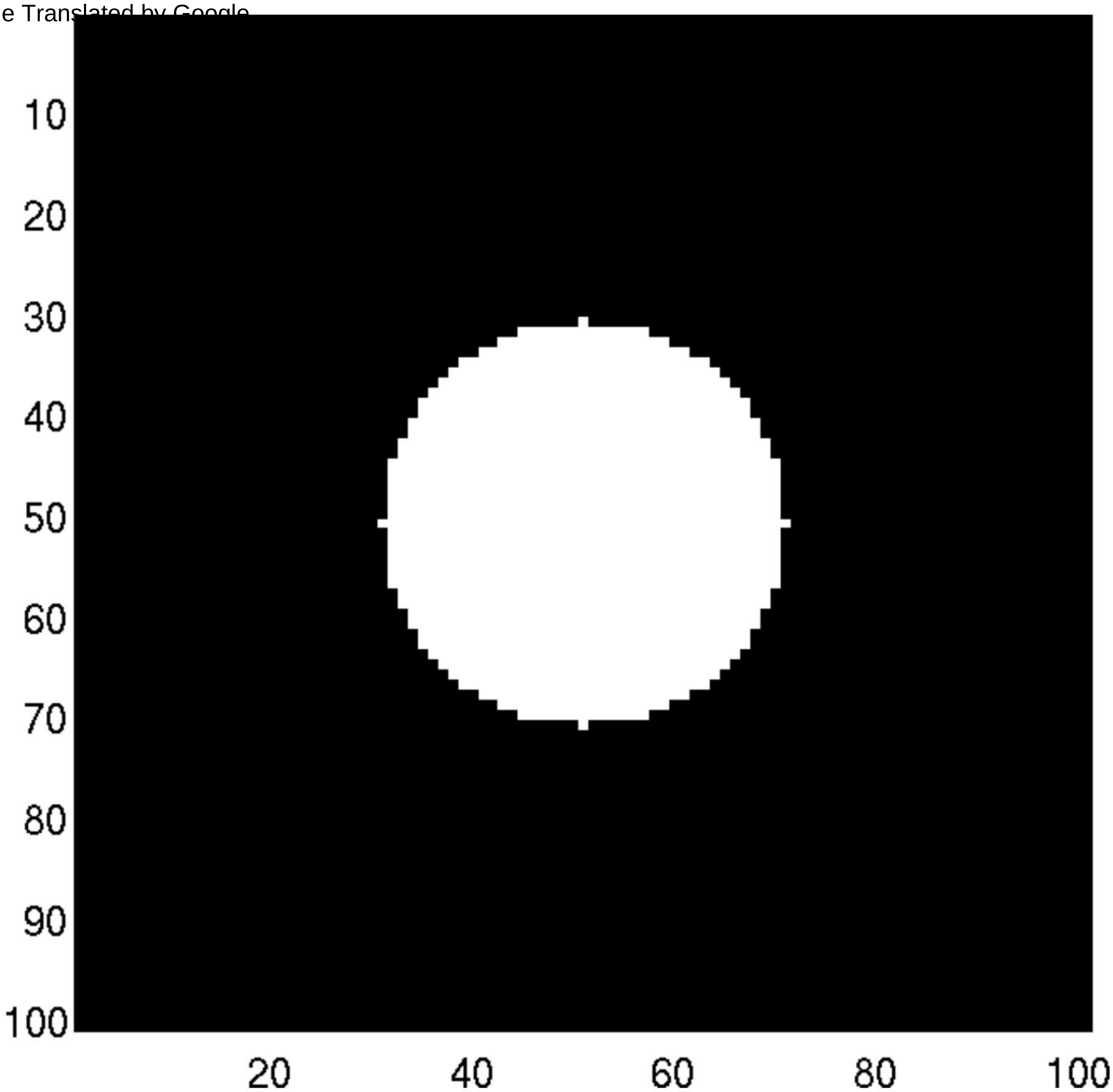

Figure 9: Transfer matrix $G_{P\,A}$ ideal high filter obtained with for a MatLab platform step.

Ideal low-pass filters

Similarly, for the case of the low-pass filter it is defined from the matrix of transfer $G_{P,B}$ given by:

$$G_{P,B}(m', n') = \begin{cases} 1 & D(m', n') \leq D_{max} \\ 0 & D(m', n') > D_{max} \end{cases}.$$

Figure [Fig2_9] shows the low-pass transfer matrix $G_{P\,B}$.

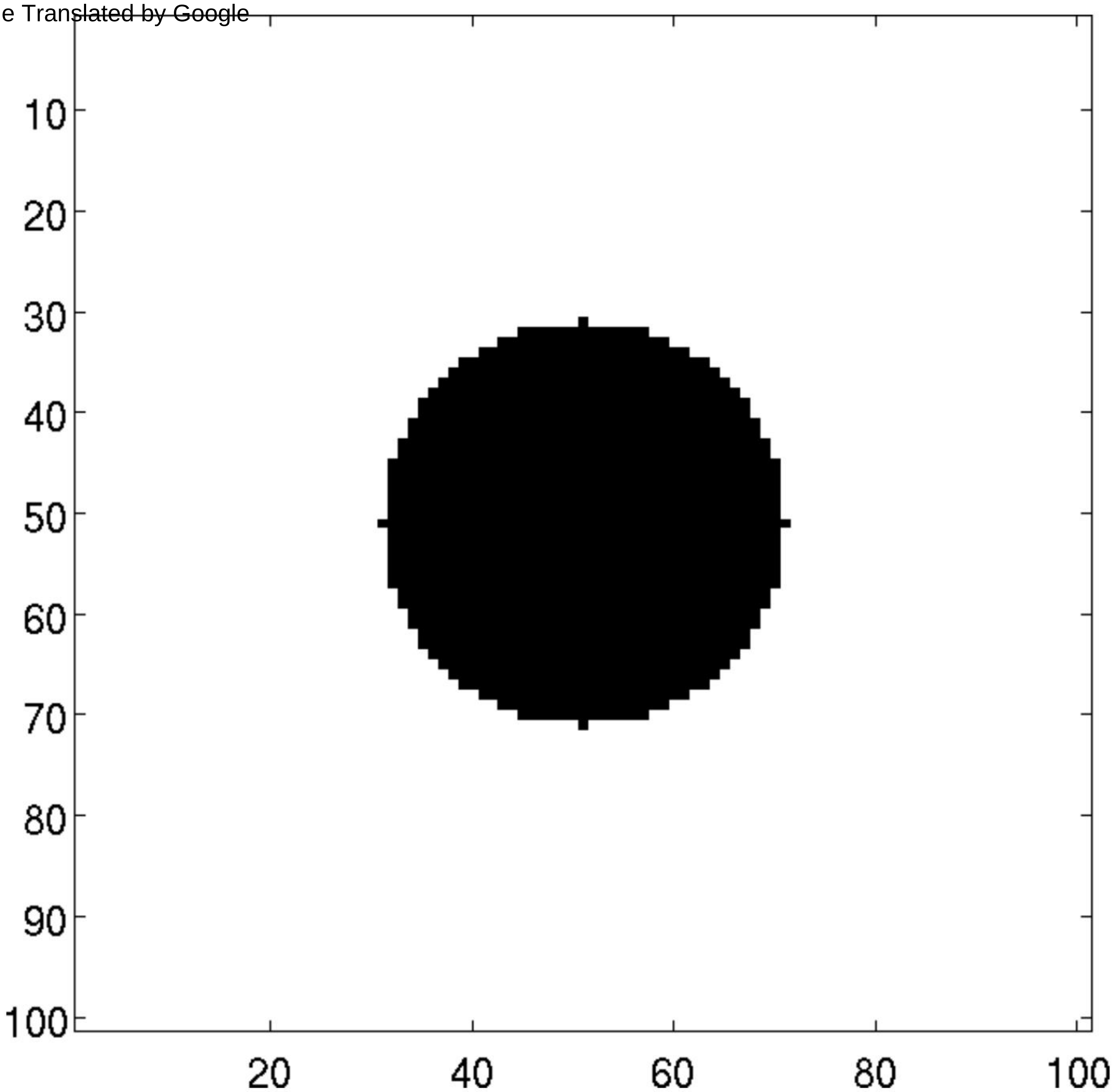

Figure 10: Transfer matrix GP B for a step ideal high filter obtained with MatLab platform.

Filtering masks

A filter mask h(m, n) It is called a spatial convolution mask if it is defined through:

$$F(m', n') = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} h(m, n)\, e^{-2\pi i\left(\frac{mm' + nn'}{N}\right)}$$

If the mask is restricted to a specific region, such that h(m, n) = 0 for m − n − Nmax < N so that the restricted mask is designated by $\hat{h}$, we obtain:

$$\hat{H}(m', n') = \frac{1}{N} \sum_{n=0}^{Nmax-1} \sum_{m=0}^{Nmax-1} \hat{h}(m, n)\, e^{-2\pi i\left(\frac{mm' + nn'}{N}\right)}$$

So that the coefficients of the expansion in ([EqLXX]) can be determined , which are minimize the amount:

$$ÿ ÿ_2 ÿ \sum_{m=0}^{Nÿ1} \sum_{n=0}^{Nÿ1} \left(\hat{H}(mÿ n) ÿ H(mÿ n) \right)^2$$

The above expression ([EqLXXI]) can be solved by means of a representation linear algebraic $\hat{H} = Q\hat{h}$ with $\hat{H}$ represented by a vector of dimension N^2 whose elements are those ordered $\hat{H}$ in some arbitrary way, it is a column vector $\hat{h}$ of $2max$ dimensions N containing the elements of $\hat{h}$ and is an array Q of dimensions $N^2 \times N\,2max$ of exponential terms according to the expression ([EqLXIX]) given by:

$$Q(k, l) = q(k, l) = \frac{1}{N} \text{ and } ÿ2ÿi\left(\frac{m m ÿ + n n ÿ}{N}\right)$$

for $k = mÿN + n\,ÿ$ with $m\,ÿ\,ÿ\,n\,ÿ \in [0, N ÿ 1]$ and $l = mN_{max} + n$ with $m\,ÿ\,n\,ÿ\,[0, N_{max}\,ÿ\,1]$.

[1] X-rays are understood to be those produced in atomic interactions, and rays are those ÿ produced by internuclear interactions. It does not exist to priori energy difference between them and both are made up of photons, the difference being made based on their origin. In From now on, the letters X or X will be used interchangeably to refer to photons.

[2] This parameter is called “threshold” and its value determines the performance of the technique.

[3] Here the function floor It is defined by assigning the largest integer to the argument that is less than the argument.

[4] A typical example is contrast-enhanced medical imaging, such as angiography.

[5] This method is useful for suppressing details or enhancing regions.

[*] Official license MathWorks 3407-8985-4332-9223-7918.