Image filtering

Chiara Ravazzi

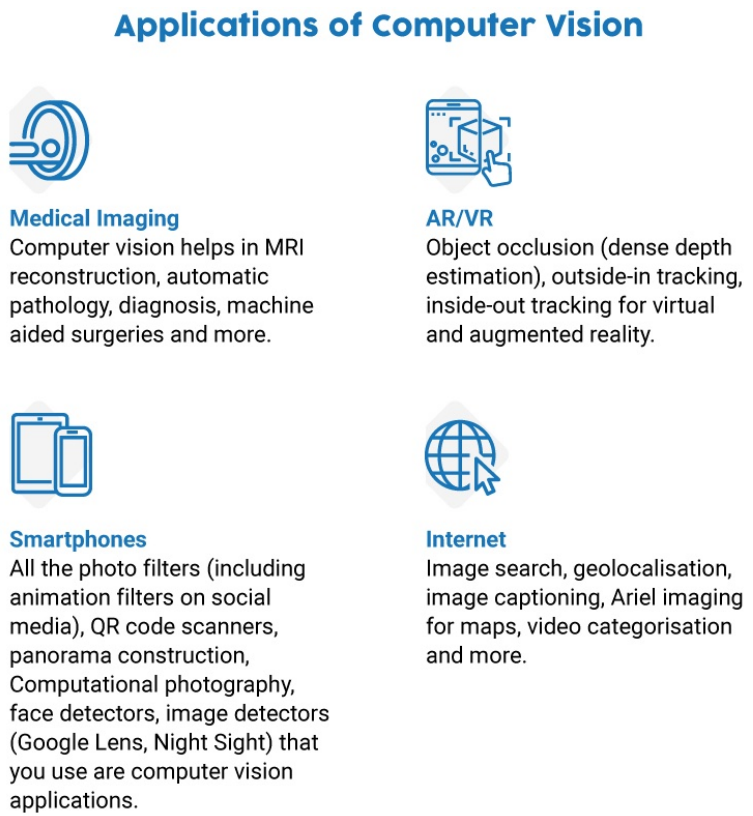
National Research Council of Italy
Institute of Electronics, Computer and Telecommunication Engineering, c/o Politecnico di Torino, Italy
Systems Modeling & Control Group

Analisi tempo-frequenza e multiscala – 01RMQNG

Computer vision

Computer vision ...

- is everywhere
  - social media feeds
  - news articles/magazines
  - books ...
- is a field of study which enables computers to replicate the human visual system.
- Collects information from digital images or videos and processes them to define attributes.



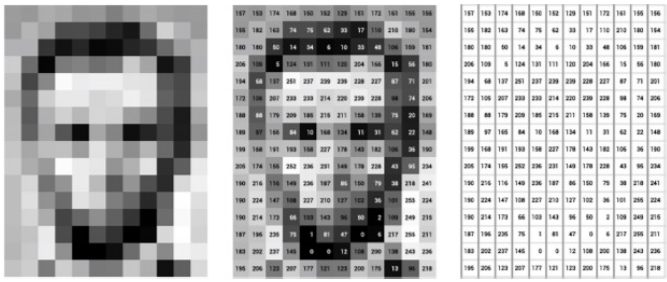
Digital images

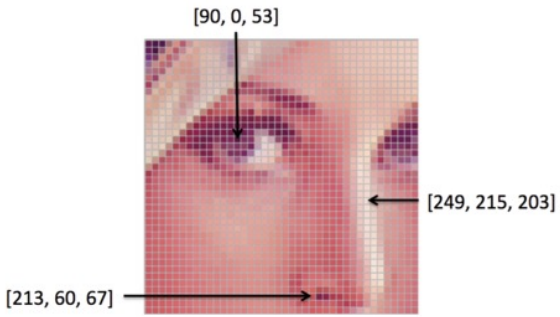
Consideriamo l'immagine come una matrice di pixel –> ad ogni entrata corresponde
- un pixel che ha come valore un numero tra 0 e 255 sulla scala di grici
- un array contenente tre entrate relative alle scale di blu, giallo e rosso con valori sempre tra 0 e 255

Think of images as functions mapping locations in images to pixel values $f(x, y)$

Gray Image

Grayscale Pixel Values
(intensity between 0 and 255)





- $f(x, y)$ vector of three values
- RGB : colors from a combination of Red, Green, and Blue (RGB).
- Three channels + integer values from 0 to 255 $\Longrightarrow$ 256*256*256 = 16,777,216 combinations or color choices.

Filtering images

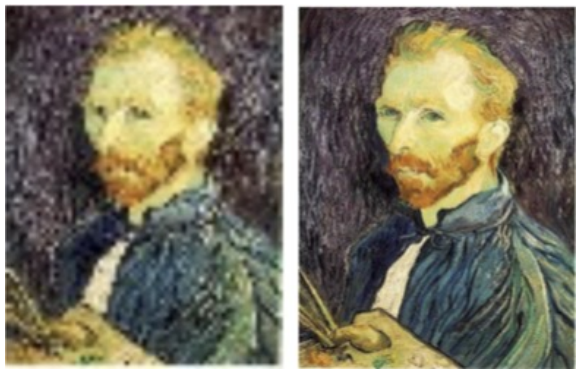
- Image filtering changes the range (i.e. the pixel values) of an image, so the colors of the image are altered without changing the pixel positions
- Filters = systems that form a new image from original image's pixel values
  - elimination of noise (denoising)
  - extraction of valuable information (edges, corners, and blobs)...

Filetering images: andare a cambiare il valore di un pixel –> l'immagine è inalterata nelle forme, ma la luminosità e/o colore si applicano modifiche

Denoising

Super-resolution

Migliora la qualità visiva delle immagini



In-painting

Permette di ricostruire l'immagine originale prima delle modifiche fatte

2-Dimensional DFT

Cosa faremo oggi --> Acquisiremo un'immagine digitale bidimensionale in scala di grigi e vedremo l'effetto di alcuni filtri su di essa
Sfrutteremo gli effetti dei filtri nel dominio delle frequenze --> quindi useremo la DFT bidimensionale

Property	Expression(s)
Fourier transform	$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$
Inverse Fourier transform	$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$
Polar representation	$F(u,v) = \lvert F(u,v)\rvert e^{-j\phi(u,v)}$
Spectrum	$\lvert F(u,v)\rvert = [R^2(u,v) + I^2(u,v)]^{1/2}$, $R = \text{Real}(F)$ and $I = \text{Imag}(F)$
Phase angle	$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
Power spectrum	$P(u,v) = \lvert F(u,v)\rvert^2$
Average value	$\bar{f}(x,y) = F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$
Translation	$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M+vy_0/N)}$ When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then $f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$ $f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$

2-Dimensional DFT

Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $\lvert F(u, v)\rvert = \lvert F(-u, -v)\rvert$
Differentiation	$\dfrac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$ $(-jx)^n f(x, y) \Leftrightarrow \dfrac{\partial^n F(u, v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2)F(u, v)$
Distributivity	$\Im[f_1(x, y) + f_2(x, y)] = \Im[f_1(x, y)] + \Im[f_2(x, y)]$ $\Im[f_1(x, y) \cdot f_2(x, y)] \neq \Im[f_1(x, y)] \cdot \Im[f_2(x, y)]$
Scaling	$af(x, y) \Leftrightarrow aF(u, v), f(ax, by) \Leftrightarrow \dfrac{1}{\lvert ab\rvert} F(u/a, v/b)$
Rotation	$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$ $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.

2-Dimensional DFT

Property	Expression(s)
Computation of the inverse Fourier transform using a forward transform algorithm	$\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}$ This equation indicates that inputting the function $F^*(u, v)$ into an algorithm designed to compute the forward transform (right side of the preceding equation) yields $f^*(x, y)/MN$. Taking the complex conjugate and multiplying this result by MN gives the desired inverse.
Convolution†	$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$
Correlation†	$f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
Convolution theorem†	$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v)$; $f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v)$
Correlation theorem†	$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v) H(u, v)$; $f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$

2-Dimensional DFT

Some useful FT pairs:

Impulse	$\delta(x, y) \Leftrightarrow 1$
Gaussian	$A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(x^2+y^2)} \Leftrightarrow Ae^{-(u^2+v^2)/2\sigma^2}$
Rectangle	$\text{rect}[a, b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow \dfrac{1}{2}[\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)]$
Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow j\dfrac{1}{2}[\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)]$

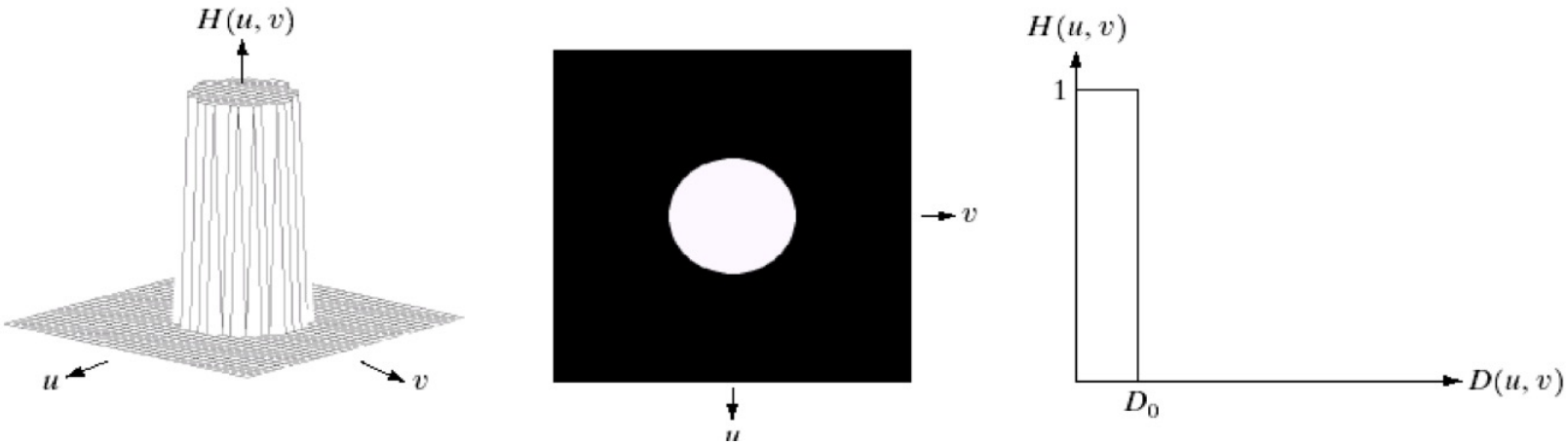
2-Dimensional DFT

Function	Description
`fft, ifft`	General FFT and inverse FFT of a real- or complex-valued signal. The resulting frequency spectrum is complex valued.
`rfft, irfft`	FFT and inverse FFT of a real-valued signal.
`dct, idct`	The discrete cosine transform (DCT) and its inverse.
`dst, idst`	The discrete sine transform (DST) and its inverse.
`fft2, ifft2, fftn, ifftn`	The two-dimensional and the N-dimensional FFT for complex-valued signals and their inverses.
`fftshift, ifftshift, rfftshift, irfftshift`	Shift the frequency bins in the result vector produced by `fft` and `rfft`, respectively, so that the spectrum is arranged such that the zero-frequency component is in the middle of the array.
`fftfreq`	Calculate the frequencies corresponding to the FFT bins in the result returned by `fft`.

Utile per centrare le frequenze nell'origine

Low-pass filter

- Low pass filter : only pass the low frequencies and drop the high ones
- Example : given D_0 cut off distance

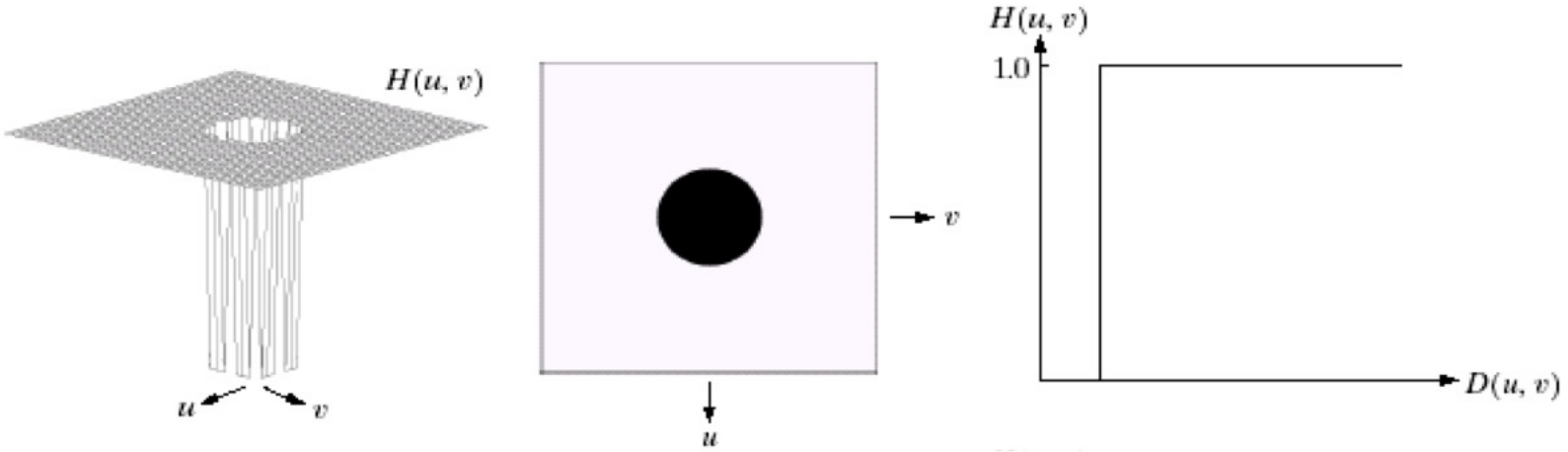


$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{otherwise} \end{cases} \qquad D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

High-pass filter

- High pass filter : only pass the high frequencies, drop the low ones

Example : Given D_0 cut off distance



$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{otherwise} \end{cases} \qquad D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$