Image processing

Filtering, Downsampling & Upsampling

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

L'esercitazione consiste nella manipolazione di immagini tramite il teorema di filtraggio e campionamento su segnali bidimensionali

Image denoising

Image noise :

- variation of brightness/color information
- intrinsic causes : i.e. sensors, scanners, digital cameras
- extrinsic conditions : i.e., environment

Image denoising :

- estimate the original image
- applications : image restoration, visual tracking, image registration, image segmentation, and image classification

Reference Image

Noisy Image

Denoised Image

Denoising by filtering

Denoising by low-pass filtering :

- Suppose Gaussian noise (independent of pixel values)
- Recover an original image from the noisy image using some filter.
- Noise = high frequency component (random variation in the pixel values)

Example : averaging method

Image

100	100	200	200
120	120	220	100
140	140	220	80
160	200	252	40

Kernel

0	1/4	0
1/4	0	1/4
0	1/4	0

Convolution product

100	100	200	200
120	145	160	100
140	170	173	80
160	200	252	40

$$100 \cdot 0 + 100 \cdot 1/4 + 200 \cdot 0 + 120 \cdot 1/4 + 120 \cdot 0 + 220 \cdot 1/4 + 140 \cdot 0 + 140 \cdot 1/4 + 220 \cdot 0 = 145$$

Image filtering

Kernel

0	¼	0
¼	0	¼
0	¼	0

Original image

Filtered image

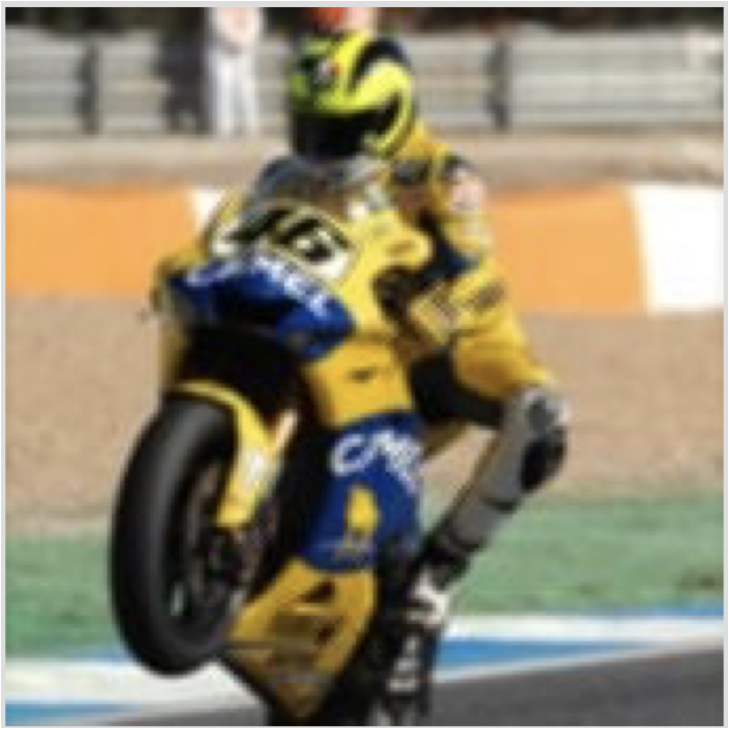


Image filtering

Kernel

0	1	0
1	-4	1
0	1	0

Original image

Filtered image

Image filtering

Different kernels produce different graphical effects

- Graphics programs often make them available as "basic" operations
- It is possible to achieve "special effects"
- Implement and compare the following filters :

$$K_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad K_2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix} \quad K_3 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Image scaling

A volte le immagini sono troppo grosse e i processi di elaborazione diventano troppo costose --> quinid risulta necessario applicare una tecnica di **image scaling**

- Too big image :
  - Requires too much memory
  - Time consuming to process
  - Too big for the screen
  - ...
- Create a smaller image by image sub-sampling

Image downsampling

Selezioniamo solo alcuni pixel --> ad esempio "buttando via" in maniera alternata righe e colonne

1/2

1/4

1/8

Throw away rows and columns ! Image reduced to $1/2$ size along each dimension.

Image downsampling

Si perde di qualità applicando questo sottocampionamento, quinid si vede l'immagine, ma più sgranata perchè si perdono i dettagli –> problema dovuto all'**aliasing**

1/2

1/4 (2x zoom)

1/8 (4x zoom)

Aliasing

- When the (spatial) sampling rate is not high enough to capture the details
- High frequencies are transformed to lower frequencies (i.e. aliases)
- To avoid aliasing the sampling rate must be at least two times the maximum frequency in the image (at least two samples per cycle)
  --> ci ricordiamo del teorema del campionamento per campionare nella maniera corretta
- This minimum sampling rate is called the Nyquist rate.
- Aliasing can be avoided by low-pass filtering the image before downsampling
  --> possiamo adottare un prefiltro

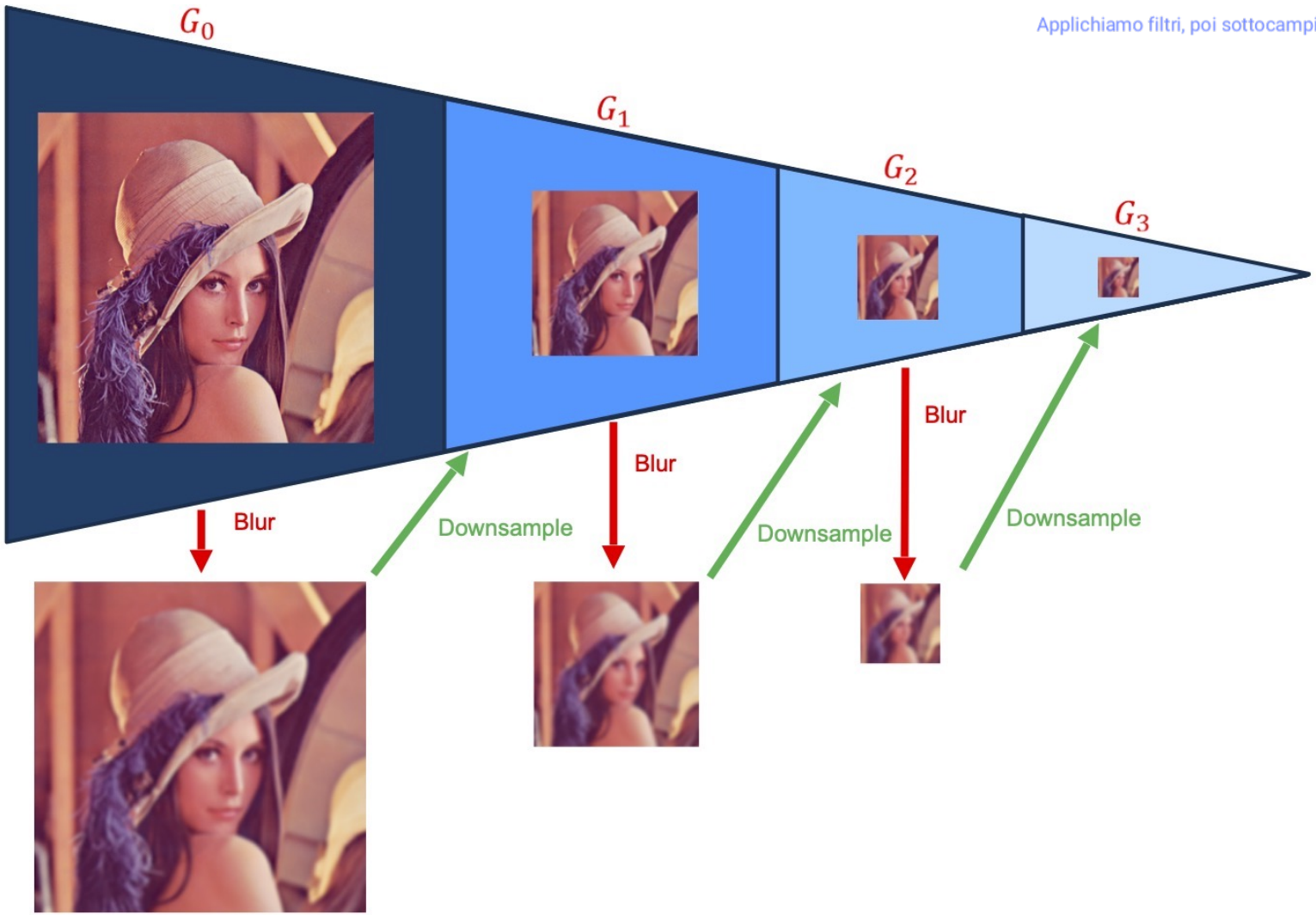
Low-pass prefiltering

Gaussian prefiltering --> è un filtro passabasso che rende più sfocata l'immagine

Subsmapling after low pass filtering

1/2 1/4 1/8

Gaussian pyramid

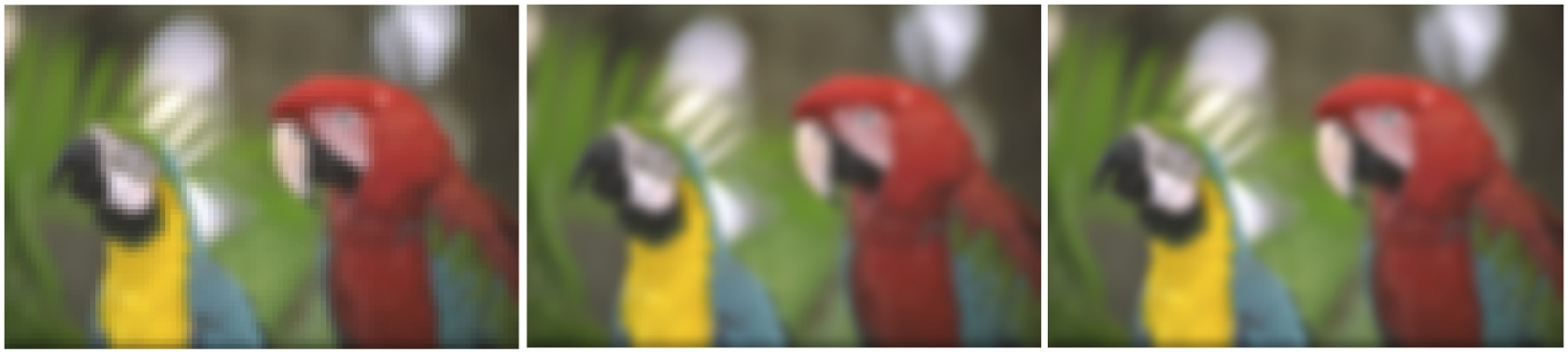


Upsampling & Interpolation

Durante l'upsampling, invece, vogliamo inserire righe e colonne andando ad aumentare la dimensione dell'immagine

Nearest neighbor interpolation :

- Repeat each row and column 10 times --> metodo semplice, ma la qualità non migliora
- Fast and simple approach.
- More sophisticated interpolation (bilinear, bicubic, ...)



Nearest neighbor | Bilinear | Bicubic