Analisi di Immagini e Video (Computer Vision)

Giuseppe Manco

Outline

- Sift
- Harries Corner Detection

Crediti

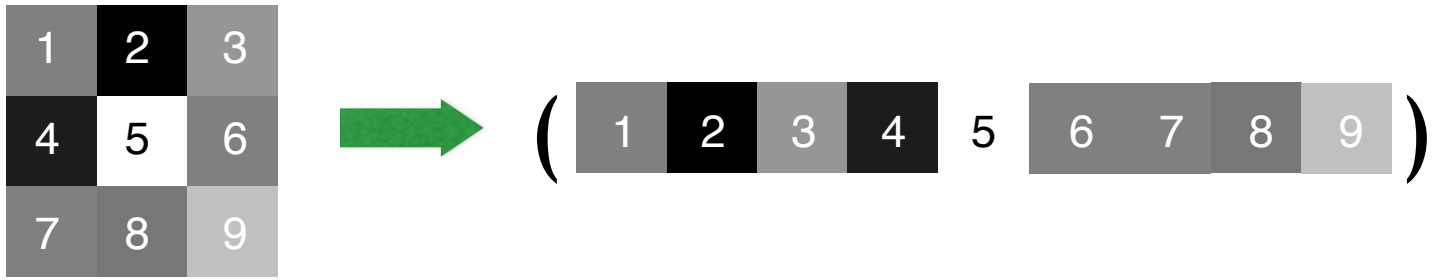
- Slides adattate da vari corsi e libri
  - Analisi di Immagini (F. Angiulli) – Unical
  - Intro to Computer Vision (J. Tompkin) – CS Brown Edu
  - Computer Vision (I. Gkioulekas) - CS CMU Edu
  - Cmputational Visual Recognition (V. Ordonez), CS Virgina Edu
  - Pattern Recognition and Machine Learning (C. Bishop, 2005)
  - Deep Learning (Bengio, Courville, Goodfellow, 2017)

Quali sono i descrittori di un keypoint?

- Vettori
- Istogrammi
- Risposte su patches

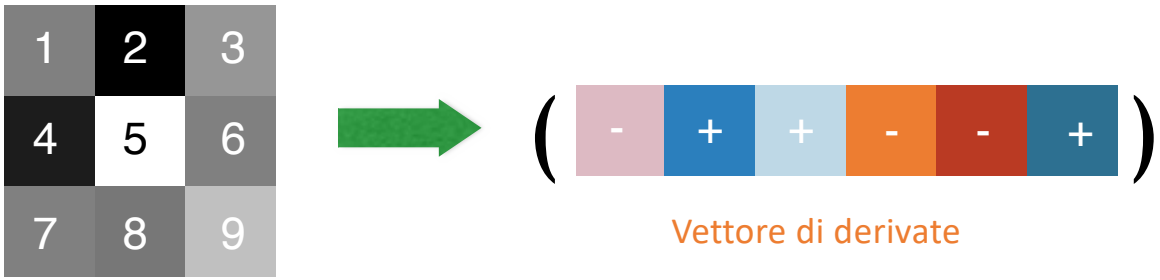
Image patch

Vettore di intensità dei colori, combinato con downsampling

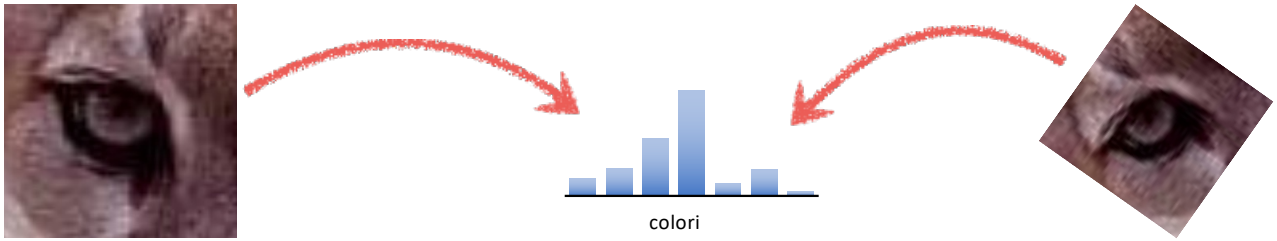


Gradienti

Differenze sui pixel



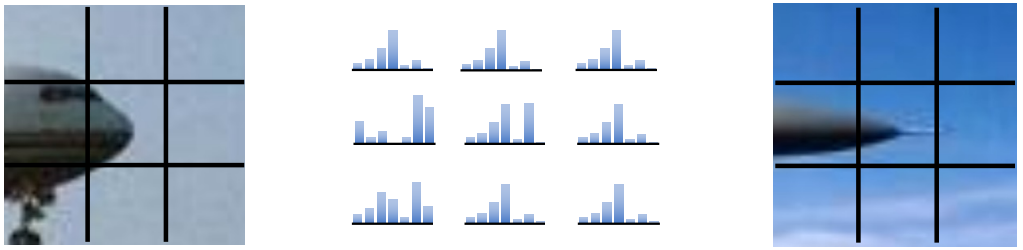
Istogrammi di colori



Invarianti ai cambiamenti di scala e rotazione

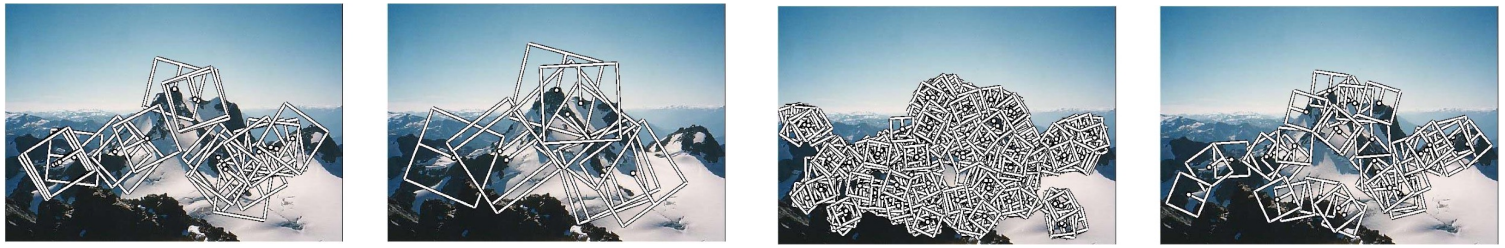
Istogrammi spaziali

Calcoliamo gli istogrammi su celle



Invariante alle deformazioni

MOPS: Multi-Scale oriented patches



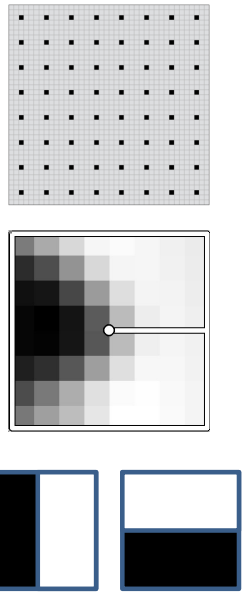
MOPS

Multi-Image Matching using Multi-Scale Oriented Patches. M. Brown, R. Szeliski and S. Winder. International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

$$(x, y, s, \theta)$$

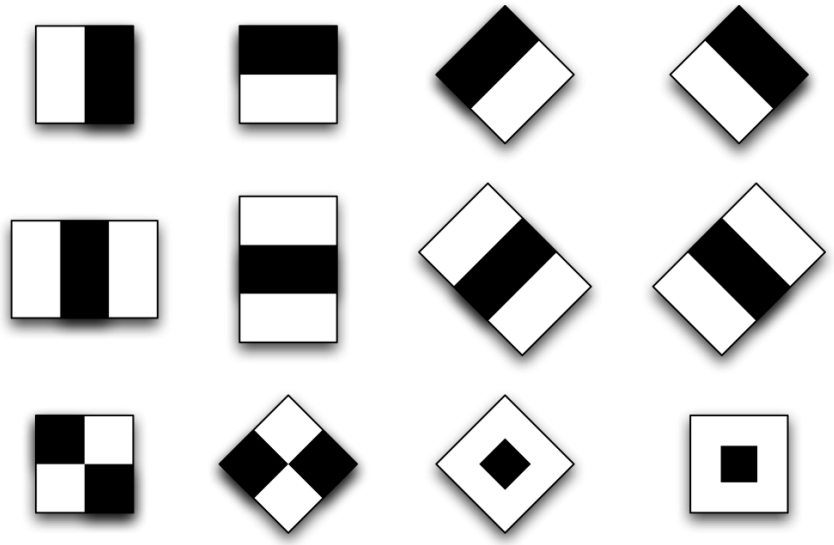
Data una feature

- Considera una patch 40 x 40, campionata
- Standardizza i valori
- Applica i filtri di Haar

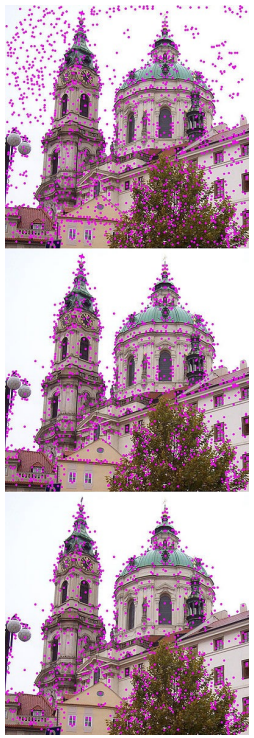


Haar-like features

Si calcola il responso della patch su ogni filtro come descrittore



SIFT
(Scale Invariant Feature Transform)

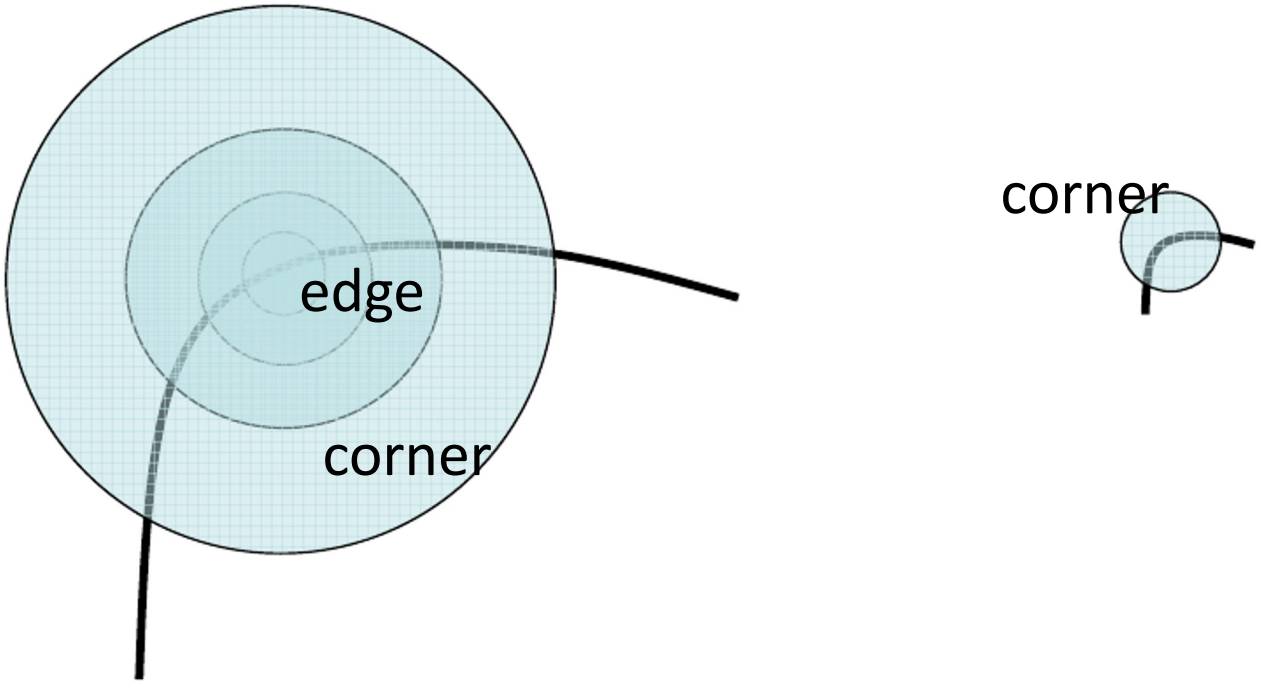


Descrive sia un detector che un descrittore

1. Multi-scale extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

https://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html

Scale Invariant Local Features



Scale Space

- Lo **scale space** è una rappresentazione multiscala dell'immagine basata su un parametro di scala continuo σ
- La rappresentazione a scala σ si ottiene effettuando la convoluzione di *f* con il kernel Gaussiano avente deviazione standard σ
- Le informazioni a scala più fine vengono progressivamente soppresse

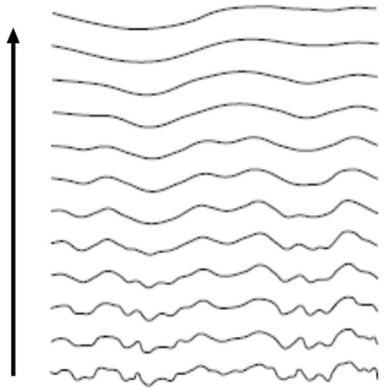


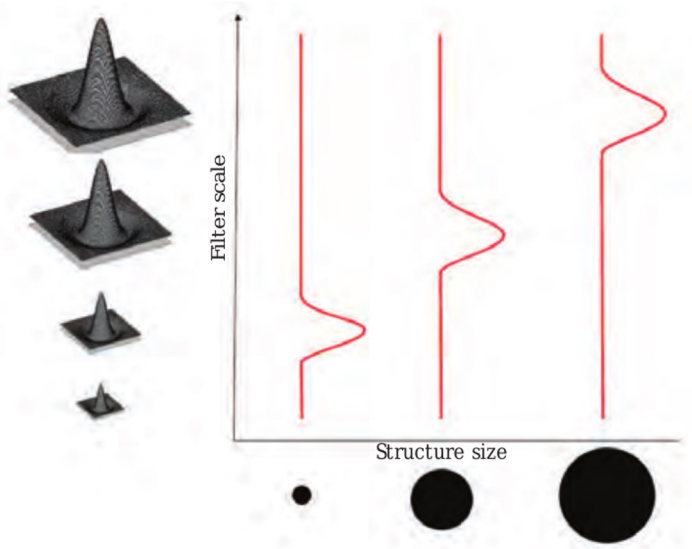
Immagine originale | σ=1 | σ=2 | σ=4

Laplacian of Gaussian Detector

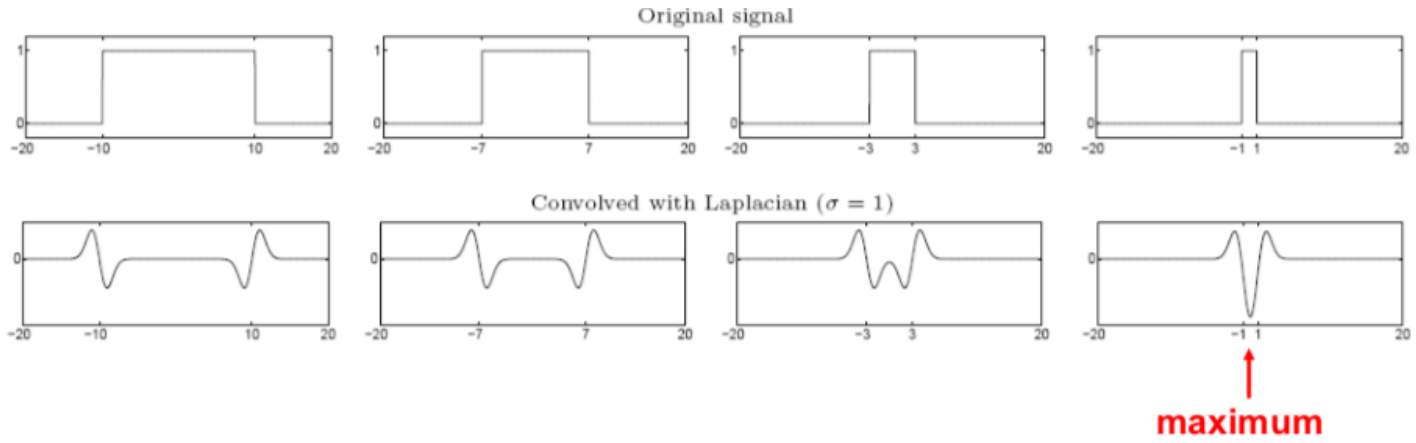
- LoG scale invariant keypoint detector:

$$L(x, y; \sigma) = \sigma^2 \left[\nabla^2 G(x, y; \sigma) \star f(x, y)\right]$$

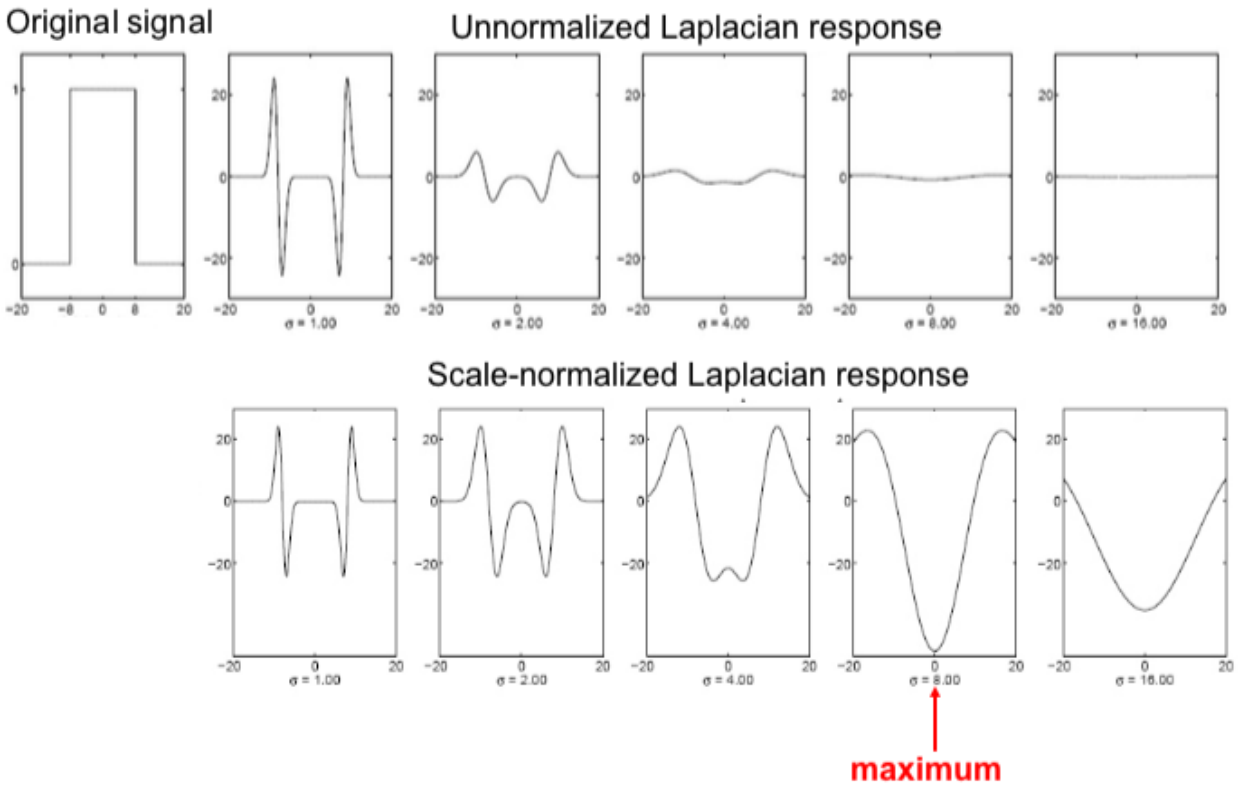
- dove il fattore σ^2 è richiesto per l'invarianza di scala
- Il LoG filter massimizza la risposta quando viene applicato ad una regione che contiene una struttura circolare approssimativamente della dimensione del filtro stesso (*blob detector*)



Laplacian response (in 1D)

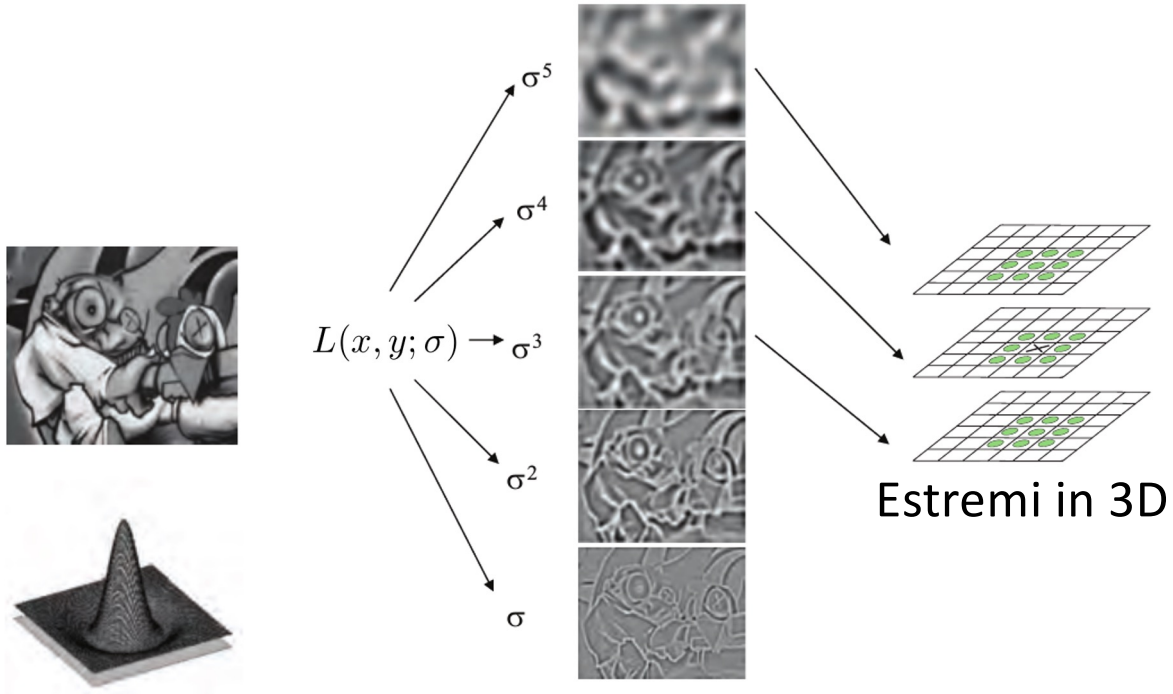


Laplacian response (in 1D)



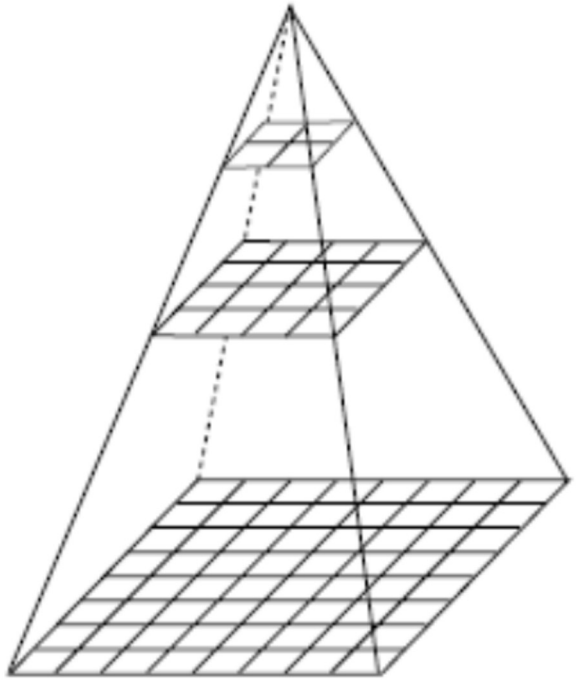
Laplacian of Gaussian Detector

- I LoG keypoint sono gli estremi in questa rappresentazione scale space



Piramide Gaussiana

- La piramide è una collezione di rappresentazioni di un'immagine strutturate su più livelli, in cui ogni livello ha in genere dimensione pari ad un quarto del livello sottostante
- In una piramide gaussiana, ogni livello è ottenuto da quello sottostante effettuando (1) un filtraggio con un kernel Gaussiano di deviazione standard σ e quindi (2) una scalatura di un fattore 0.5



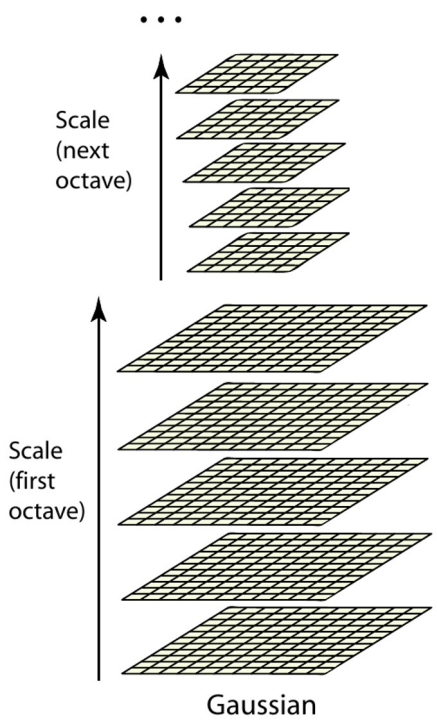
Piramide Gaussiana

- Dato un fattore di scala σ ed un operatore di scalatura *S*:
  - $f_0(x,y) = f(x,y)$
  - $f_{n+1}(x,y) = S(G(x,y,\sigma)*f_n(x,y))$
- dove $f_{n+1}(x,y)$ ha sempre dimensione pari ad un quarto di $f_n(x,y)$ per effetto del dimezzamento del numero di pixel in verticale ed in orizzontale (l'immagine $f_n(x,y)$ al livello *n* della piramide è l'equivalente dell'immagine $G(x,y,2^{n-1}\sigma)*f_0(x,y)$)

Piramide Gaussiana

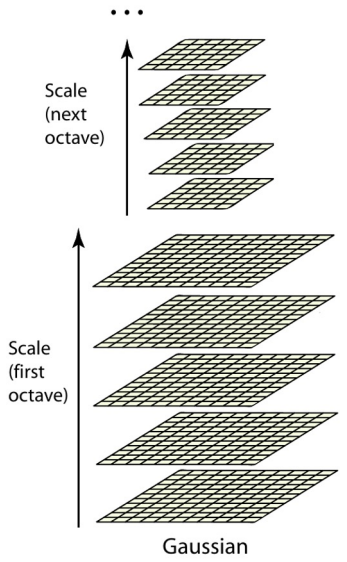
Piramide Gaussiana: Ottave

- Per rendere più efficiente l'analisi, lo scale space è diviso in **ottave**, ognuna delle quali copre un livello della piramide Gaussiana (scale da σ_L a $2\sigma_L$)
- Sia σ_L la scala di un certo livello della piramide, l'ottava (da σ_L a $2\sigma_L$) viene a sua volta suddivisa in s intervalli (n=0,1,...,s) aventi scala
  - $\sigma_n = k^n \sigma_L$ con $k = 2^{1/s}$
- Si noti che $\sigma_0=\sigma_L$ e $\sigma_n=2\sigma_L$ e quindi l'ultima scala di ogni ottava corrisponde alla prima scala dell'ottava successiva (in quanto il kernel Gaussiano utilizzato raddoppia ad ogni livello della piramide)
- Nel passaggio all'ottava successiva, si dimezza la dimensione l'immagine ed è possibile riapplicare gli stessi kernel gaussiani utilizzati nell'ottava precedente (anziché raddoppiare la dimensione dei kernel)



Piramide Gaussiana: Ottave

- Ad esempio, per *s*=3 avremo:
  - $k = 2^{1/s} = 1.2599$
  - $\sigma_0 = k^0 = 1 \rightarrow$ kernel 7x7
  - $\sigma_1 = k^1 = 1.2599 \rightarrow$ kernel 9x9
  - $\sigma_2 = k^2 = 1.5874 \rightarrow$ kernel 11x11
  - $\sigma_3 = k^3 = 2 \rightarrow$ kernel 13x13
- Per ottenere le diverse ottave è sufficiente applicare sempre gli stessi kernel ad ogni livello della piramide



Difference of Gaussian Detector

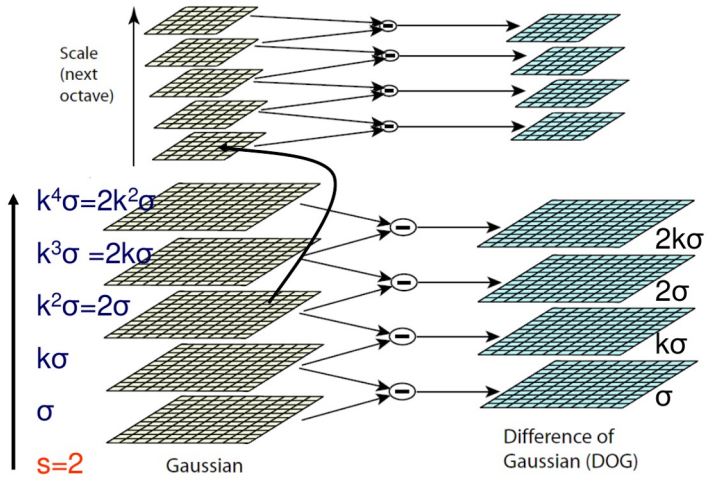
- Il filtro LoG può essere approssimato dalla differenza di due Gaussiane (DoG) aventi differente deviazione standard:

$$\begin{aligned} D(x,y;\sigma) &= [G(x,y;k\sigma) - G(x,y;\sigma)] \star f(x,y) \approx \\ &\approx \left[(k-1)\sigma^2 \nabla^2 G(x,y;\sigma)\right] \star f(x,y) = \\ &= (k-1)L(x,y;\sigma). \end{aligned}$$

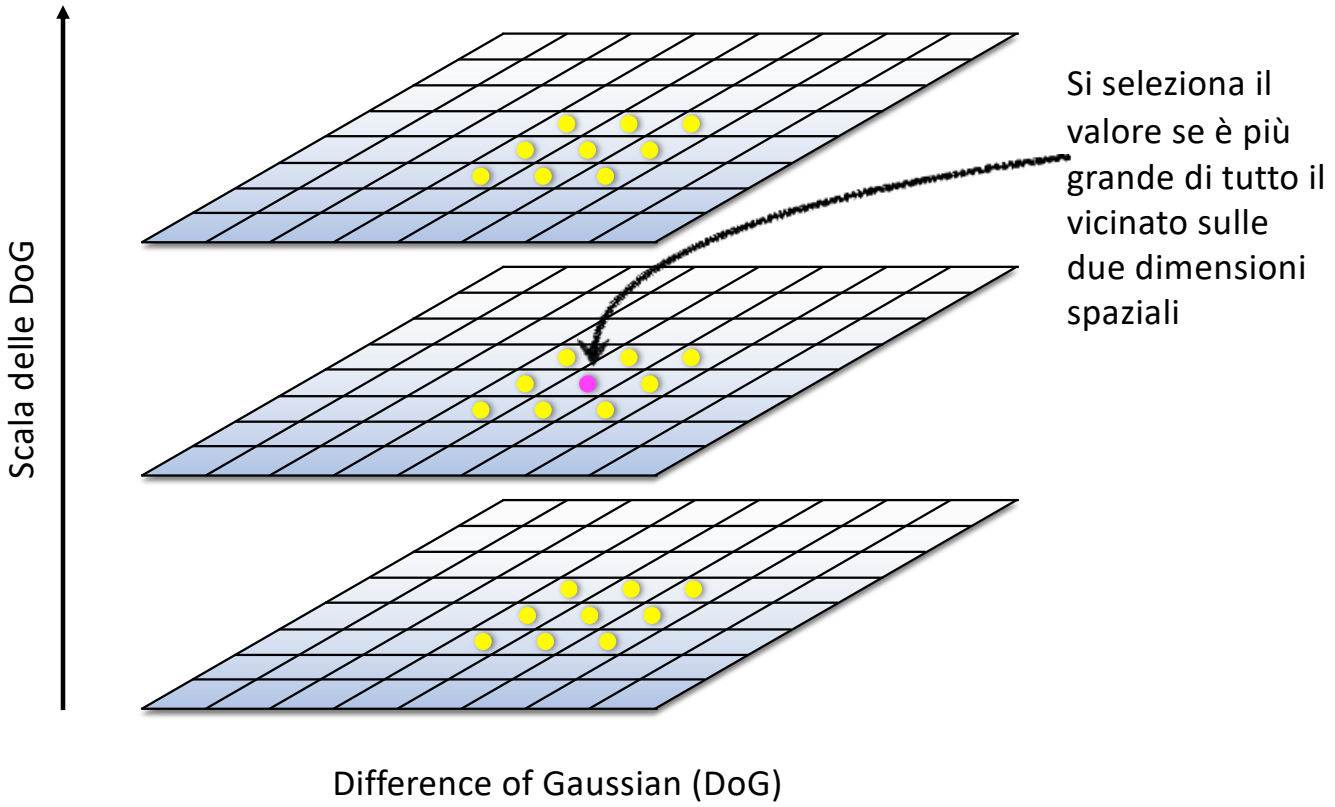
- Il vantaggio di questa espressione è che può essere efficientemente calcolata come differenza di due scale adiacenti

Difference of Gaussian Detector

- Al fine di determinare anche i veri massimi nella prima immagine di ogni ottava, per ogni livello della piramide si determinano s+3 intervalli
- Per s=2 si calcolano s+3=5 immagini con σ crescente di un fattore $k=\sqrt{2}$
- Si calcolano le differenze tra livelli adiacenti (DoG) e si determinano gli estremi nella corrispondente rappresentazione 3D



Scale-space extrema & Keypoint location



Interest Region Extraction

- Per catturare la struttura nell'intorno del keypoint, viene considerata la regione di raggio *r* centrata nel punto individuato (raggio $r=3\sigma$ viene utilizzato da diversi detector, dove σ denota la scala caratteristica del keypoint) detta anche *interest region*
- Dopo che l'interest region è stata selezionata, dev'essere *normalizzata* per invarianza alle rotazioni:
  - Si determina l'*orientazione dominante* e quindi si ruota il contenuto della regione in accordo all'associato angolo in modo da portarla in un'*orientazione canonica*

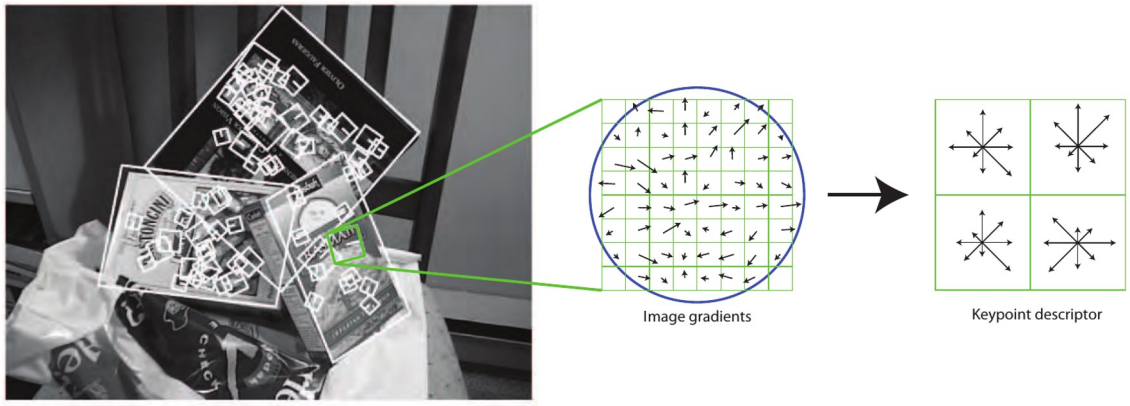
Interest Region Extraction

- Determinazione dell'**orientazione dominante** (Lowe 2004):
  - si costruisce un istogramma delle orientazioni con 36 intervalli (bins) che coprono 360 gradi (con un passo di 10 gradi)
  - Per ogni pixel nella regione considerata, si calcola il gradiente e la corripondente orientazione viene inserita nell'istogramma dopo averla pesata in base al rispettivo modulo e ad una funzione Gaussiana di deviazione standard 1.5σ centrata nel keypoint
  - Il picco dell'istogramma viene scelto come orientazione dominante (per determinare l'orientazione precisa si effettua un'interpolazione tra usando i 3 bin centrati nel picco)

SIFT Feature Descriptor

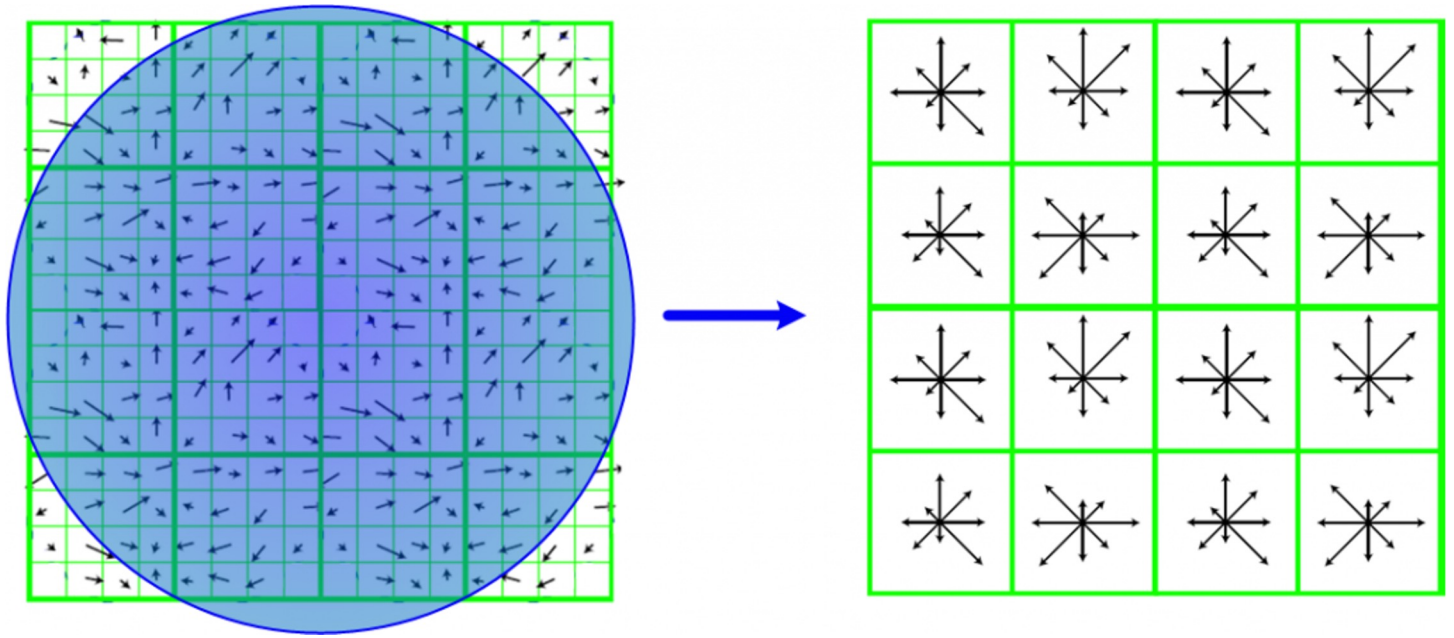
- La **Scale Invariant Feature Transform** (SIFT) è stata introdotta da Lowe (2004) come una combinazione
  - del DoG interest point/region detector e
  - del SIFT feature descriptor
- Queste due componenti sono state utilizzate in seguito anche come tecniche a sé stanti

SIFT Feature Descriptor

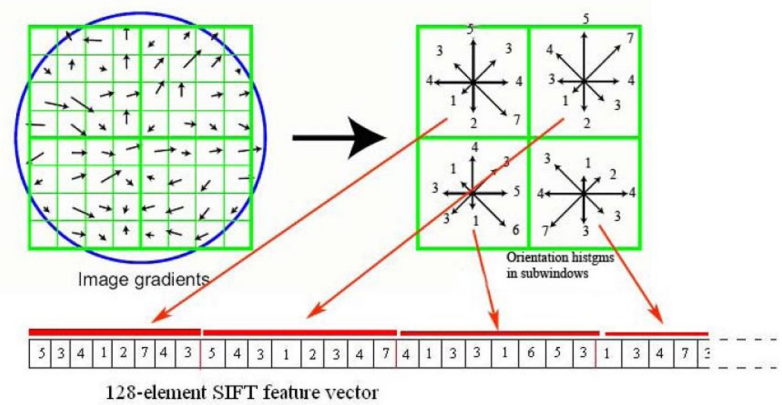


- Dopo che l'interest region è stata portata nell'orientazione canonica, viene costruita una griglia di 16x16 celle che copre l'intera interest region (per ottenere invarianza alla scala)
- Si determinano i gradienti in corrispondenza delle suddette celle e le rispettive orientazioni, dopo essere state pesate in base al rispettivo modulo e ad un kernel Gaussiano con $\sigma=r/2$, vengono inserite in una griglia 4x4 (avente grana più grossa della precedente) di istogrammi di orientazioni, ognuno dei quali è composto da 8 bin (associati alle 8 direzioni principali)

SIFT Feature Descriptor



SIFT Feature Descriptor



- Il SIFT descriptor (*localized set of gradient orientation histograms*) si ottiene concatenando i 16 istogrammi (4 x 4) da 8 bins, ottenendo un vettore a 4 x 4 x 8 = 128 dimensioni:
  - L'alta dimensionalità rende il descrittore distintivo
  - La suddivisione spaziale permette ai gradienti di subire piccoli spostamenti
- Per rendere il descrittore meno sensibile ai cambiamenti d'illuminazione il vettore viene normalizzato all'unità di lunghezza