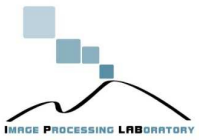
Low Level Vision

Keypoint detectors/descriptors: oltre le SIFT

Tony Meccio

27/05/2009

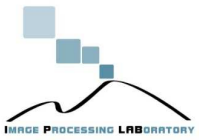


Keypoint

- Abbiamo visto precedentemente come i **keypoint** (feature point) permettono di identificare e caratterizzare i punti salienti di un'immagine.
- Vedremo ora alcune tecniche per individuarli (**keypoint detector**) e caratterizzarli (**keypoint descriptor**).
  - I due problemi sono **completamente separabili**.

Keypoint

- Un keypoint è individuato in una determinata **posizione**, a una certa **scala** e secondo un **orientamento** principale.
  - Il caso di più orientamenti viene gestito replicando il keypoint.
- Indipendentemente dal detector usato per individuarlo, il descriptor viene calcolato relativamente a posizione, scala e orientamento trovati.



Multi-Scale Keypoint Detection

- Per cercare keypoint a più scale, è necessario costruire una rappresentazione scale-space dell'immagine.
- Lowe, nella tecnica SIFT, propone di costruire una piramide gaussiana.
- Se il detector è basato su derivate (come nella maggior parte dei casi) è possibile usare un'alternativa basata sulle **derivate di gaussiana**.

Gaussian Derivatives

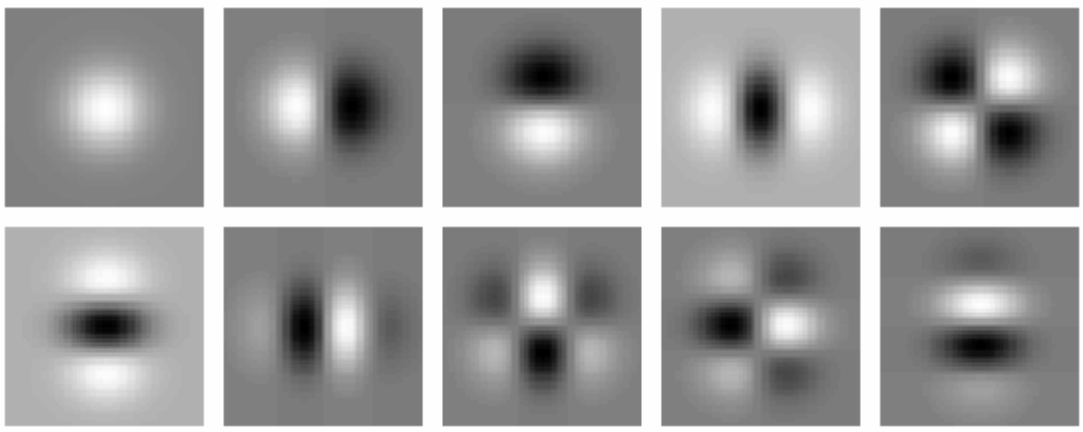
- Si dimostra infatti che derivare l'immagine smussata è equivalente a **derivare il kernel gaussiano**:

$$L_x = \frac{\partial}{\partial x}(I * g) = I * \left(\frac{\partial g}{\partial x}\right) \qquad L_y = \frac{\partial}{\partial y}(I * g) = I * \left(\frac{\partial g}{\partial y}\right)$$

  - E così via per gli altri ordini di derivazione (pura e mista).

Gaussian Derivatives

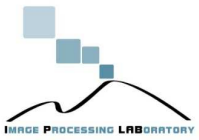
- È quindi possibile costruire una famiglia di kernel di **derivazione gaussiana**:



Low Level Vision
Keypoint detectors/descriptors

Gaussian Derivatives

- La deviazione standard σ_D del kernel di derivazione è chiamata **scala di derivazione**.
- Attraverso tale parametro è possibile variare la finezza della rappresentazione e quindi cercare feature a scale diverse.
- Il **parametro di scala** risulta essere $t = \sigma_D^2$.

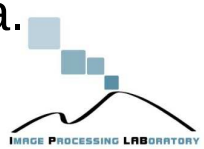


Gaussian Derivatives

- La convoluzione con kernel di derivazione gaussiana, analogamente a quanto già visto col kernel gaussiano, è **separabile (a qualunque ordine)**.
  - È quindi sufficiente effettuare due convoluzioni monodimensionali anziché una bidimensionale.
  - L'ordine di derivazione di ciascuna gaussiana monodimensionale è uguale all'ordine di derivazione del kernel desiderato lungo la direzione corrispondente.

Gaussian Derivatives

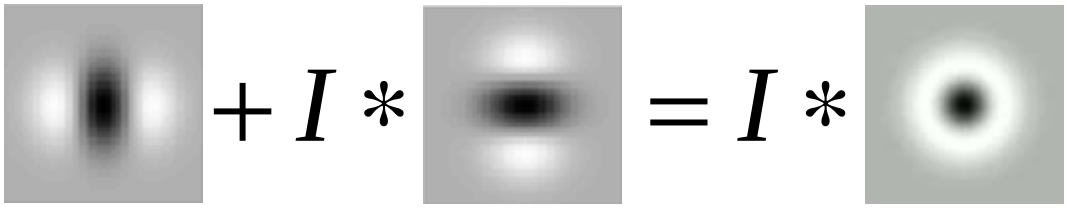
- Lo scale-space gaussiano derivato può essere calcolato in anticipo, possibilmente dividendolo in ottave come già visto precedentemente.
  - Generalmente occuperà di più del semplice scale-space gaussiano (più direzioni e/o ordini di derivazione), ma avrà più “calcoli già fatti”.
- In alternativa è possibile effettuare i calcoli “al volo” ogni volta che servono.
  - Meno efficienza ma più risparmio di memoria.
- Si sceglie in base alle esigenze del sistema.

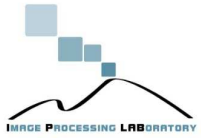


Gaussian Derivatives

- Un’altra proprietà conveniente dei kernel gaussiani derivati è data dalla linearità dell’operazione di convoluzione.
- Essa permette di definire operazioni più complesse **precalcolandone direttamente i kernel**:

$$I * \quad + I * \quad = I * \quad$$



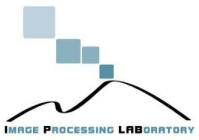


Laplacian of Gaussian

- Un operatore utile per cercare punti interessanti è l'operatore **laplaciano**:

$$\nabla^2 I = I_{xx} + I_{yy}$$

- Esso ha valori alti o bassi in punti con intensità diversa dai punti vicini.



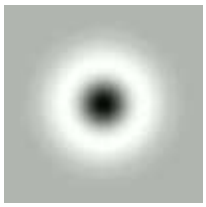
Laplacian of Gaussian

- Ulitizzando i kernel di derivazione gaussiana per calcolare le derivate seconde otteniamo il **laplaciano di gaussiana** (LoG):

$$\nabla^2 L = L_{xx} + L_{yy} = I * \frac{\partial^2 g}{\partial x^2} + I * \frac{\partial^2 g}{\partial y^2} = I * \nabla^2 g$$

Laplacian of Gaussian

- Il Laplaciano di Gaussiana ha risposte estreme (massimi e minimi locali) in presenza di **blob** (zone di intensità maggiore o minore dell'intorno) di dimensione all'incirca uguale a σ_D:



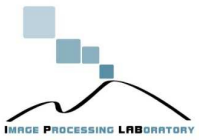
Laplacian of Gaussian

- Variando σ_D, è possibile quindi cercare blob a scale diverse.
- Per ottenere range di valori coerenti tra scale diverse è necessario usare il **laplaciano di gaussiana normalizzato**:

$$\nabla^2_{norm} L = \sigma_D^2 (L_{xx} + L_{yy})$$

Laplacian of Gaussian

- Tramite il laplaciano di gaussiana normalizzato è possibile trovare non solo la posizione, ma anche **la scala caratteristica** di ciascun blob.
- I keypoint selezionati dall'operatore LoG sono dunque **gli estremi locali** (massimi e minimi) della funzione $\nabla^2_{norm} L$, al variare di ***x***, ***y*** e σ_D.
- Tale operatore è del tutto analogo all'operatore usato da Lowe per le SIFT, il Difference of Gaussians, che infatti ne costituisce un'approssimazione.



Laplacian of Gaussian

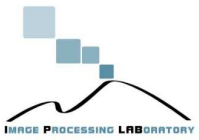
- Come già visto precedentemente, quest'operatore restituisce anche blob a basso contrasto (che possono essere filtrati imponendo una **soglia minima** sul valore della laplaciana).
- Inoltre, restituisce anche blob facenti parte di edge, quindi difficili da localizzare e matchare. Questi vanno filtrati nel modo già visto (tramite il confronto tra gli **autovalori della matrice Hessiana**).

Multi-Scale Harris Detector

- L'Harris Corner Detector si basa sugli autovalori della matrice di autocorrelazione locale:

$$A = \sum_W w \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- Lo svantaggio principale è la **non invarianza alla scala**.



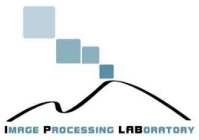
Multi-Scale Harris Detector

- È possibile migliorare l'approccio utilizzando la **derivazione gaussiana** in luogo della derivazione puntuale.

- Variando la scala di derivazione σ_D è possibile **cercare corner su più scale**.

Multi-Scale Harris Detector

- È necessario variare anche la dimensione della finestra W. Si usa una finestra gaussiana con deviazione standard σ_I, chiamata **scala di integrazione**.
  - Solitamente si pone $\sigma_I = \gamma\sigma_D$, con γ fissato tra 2 e 4.
- Per ottenere valori coerenti tra scale diverse si usa anche qui il **fattore di normalizzazione** σ_D^2.



Multi-Scale Harris Detector

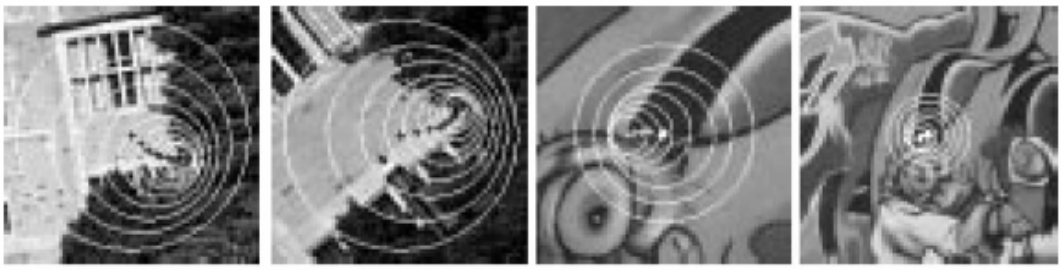
- La matrice di autocorrelazione locale diventa quindi:

$$A = \sigma_D^2 \sum_{W_I} w_I \begin{bmatrix} L_x^2 & L_x L_y \\ L_x L_y & L_y^2 \end{bmatrix}$$

  - Dove W_I rappresenta la finestra gaussiana con deviazione standard σ_I.

Multi-Scale Harris Detector

- Il resto del procedimento avviene come nel caso di Harris semplice.
- Cercando indipendentemente i corner su più scale diverse, ciascun corner viene visto più volte:



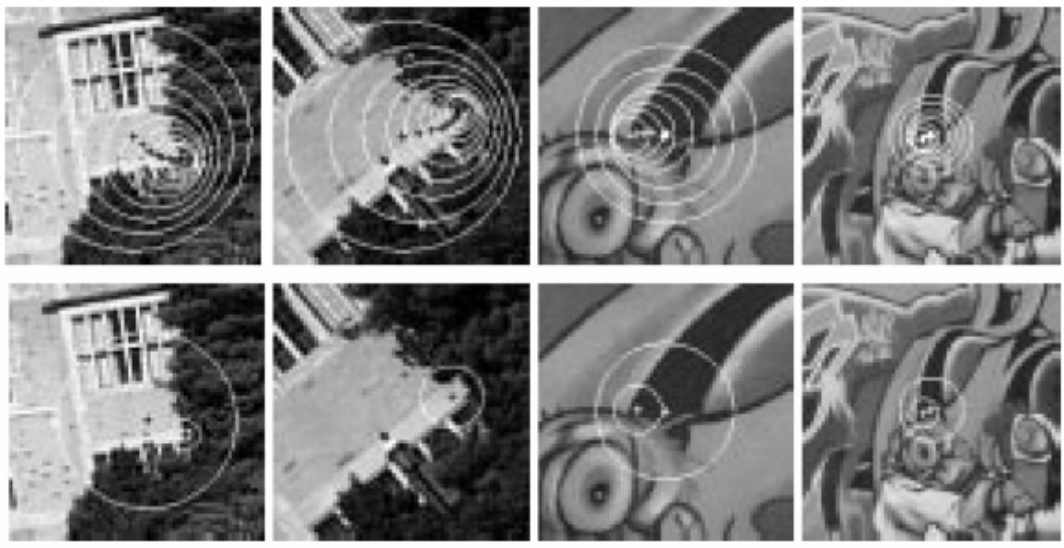
Multi-Scale Harris Detector

- Per evitare ciò, è necessario selezionare la **scala caratteristica** di ciascun corner.
- L'approccio più semplice consiste nel cercare i massimi locali della funzione di cornerness al variare di x, y e σ_D.
- Tuttavia, però, le prestazioni di quest'approccio **non sono soddisfacenti**.
  - Spesso la cornerness non raggiunge lungo l'asse della scala un massimo locale ben definito.
  - Molti corner quindi non vengono rilevati.

Harris-Laplace Detector

- Un approccio migliore per selezionare la scala dei corner è chiamato **Harris-Laplace** detector.
- Esso consiste nel cercare i corner **indipendentemente per ogni scala** e poi scartare i corner per i quali il **laplaciano di gaussiana normalizzato** non raggiunge un estremo (massimo o minimo) lungo l'asse della scala.
  - Massimo locale lungo x, y della cornerness.
  - Estremo locale lungo σ_D del LoG.

Harris-Laplace Detector



- Vengono effettivamente selezionate solo le scale più significative.

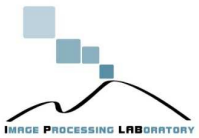
Multi-Scale Hessian Detector

- Un altro tipo di detector è basato sulla **matrice hessiana**:

$$HI = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

- E più precisamente sul suo **determinante**:

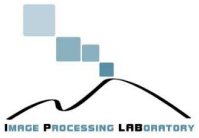
$$\det HI = I_{xx} I_{yy} - I_{xy}^2$$



Multi-Scale Hessian Detector

- Tale determinante ha valori elevati quando l'intensità dei punti vicini **varia concordemente** lungo **entrambe le curvature principali** (è infatti il prodotto dei due autovalori).

- Anche questo, quindi, è un **detector basato su blob** (identifica zone che hanno un'intensità maggiore o minore dell'intorno).

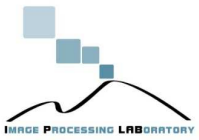


Multi-Scale Hessian Detector

- Passando dalla derivazione puntuale a quella gaussiana e applicando il **fattore di normalizzazione** (che in questo caso è σ_D^4) si ha:

$$\det H_{norm} L = \sigma_D^4 (L_{xx} L_{yy} - L_{xy}^2)$$

- Tale funzione ha massimi locali in corrispondenza di blob di dimensione σ_D.
  - Imponendo una **soglia minima**, è possibile scartare i blob a basso contrasto.



Multi-Scale Hessian Detector

- Il determinante dell'hessiana, rispetto al laplaciano, ha un'importante proprietà: il suo valore è alto solo se **entrambi gli autovalori solo grandi** (in valore assoluto).

- Il detector hessiano, quindi, **filtra automaticamente blob facenti parte di edge**.
  - Non c'è bisogno di applicare un filtraggio successivo.

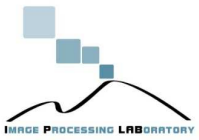
Multi-Scale Hessian Detector

- Questa differenza si spiega facilmente osservando che il laplaciano corrisponde alla **traccia** della matrice hessiana:

$$\nabla^2 I = I_{xx} + I_{yy} = \mathrm{tr}\, HI = \lambda_1 + \lambda_2$$

$$\det HI = \lambda_1 \lambda_2$$

  - Se gli autovalori sono molto diversi la somma ha un valore alto, ma il prodotto ha un valore basso.

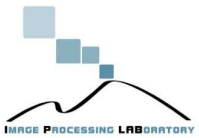


Hessian-Laplace Detector

- Analogamente a quanto visto per Harris, ci sono due modi per scegliere la **scala caratteristica** dei keypoint.
- È possibile cercare i massimi locali del determinante al variare di x, y e σ_D.
- Oppure, analogamente a Harris-Laplace, è possibile usare il laplaciano per selezionare la scala (**Hessian-Laplace** detector).
  - Quest'ultima soluzione ha prestazioni migliori.

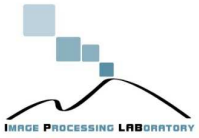
MSER

- Maximally Stable Extremal Regions.
- Si tratta di un detector basato sull'operazione di **thresholding**.
- Consiste fondamentalmente nel cercare (pixel per pixel) **blob stabili rispetto alla variazione di soglia** di thresholding.



MSER

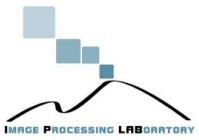
- L'operazione di thresholding trasforma l'immagine in una **mappa binaria** (rappresentata come immagine black/white) che comprende solo i pixel che hanno intensità maggiore di una certa soglia *th*.
- Per *th* che va dal valore massimo (bianco) al valore minimo (nero), l'immagine sogliata comprende sempre più pixel, che formano delle **regioni connesse** che aumentano di area e si uniscono tra loro.



MSER

- Tali regioni sono chiamate **extremal regions**, dato che la loro luminosità è maggiore di quella del loro intorno.

- Se una regione è ben definita rispetto al suo intorno, il numero dei suoi pixel **varia lentamente** al variare della soglia *th*.



MSER

- Sono considerate **maximally stable extremal regions** le extremal regions per le quali la variazione d'area, normalizzata rispetto all'area, raggiunge un **minimo locale**.
  - Se Q_{th} è un'extremal region al variare della soglia *th*, e la funzione $q(th) = \left|Q_{th-\Delta} \setminus Q_{th+\Delta}\right| / \left|Q_{th}\right|$ assume un minimo locale in *th**, allora Q_{th^*} è una MSER.

MSER

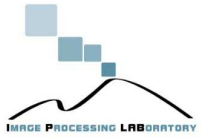
- Vengono selezionate come regioni di interesse tutte le MSER trovate al variare di *th*.

- Successivamente viene effettuata la stessa operazione sul **negativo dell'immagine** (per rilevare anche le extremal regions di minimo locale).

MSER

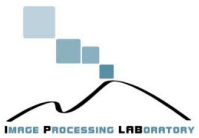
- L'operazione di thresholding seleziona automaticamente le extremal regions indipendentemente dalla loro scala:

- Se si vogliono ottenere keypoint, si può fittare un circonferenza su ciascuna MSER.



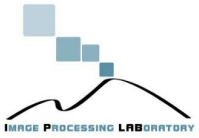
Orientation Assignment

- Finora non abbiamo parlato di **come assegnare un'orientamento ai keypoint trovati**.
- Si può usare il metodo usato da Lowe per le SIFT (**istogramma di orientamenti dei gradienti locali**).
- Oppure si può semplicemente calcolare l'**orientamento della media dei gradienti locali** (meno robusto ma più semplice).



Affine Detectors (cenni)

- I detector Harris-Laplace e Hessian-Laplace sono stati ulteriormente perfezionati per rilevare **keypoint di forma ellittica**, invarianti anche per **trasformazioni affine** (**Harris-Affine** e **Hessian-Affine**).
- Anche col detector MSER è possibile rilevare keypoint ellittici: è sufficiente, per ciascuna MSER trovata, **fittare un'ellisse anziché una circonferenza**.

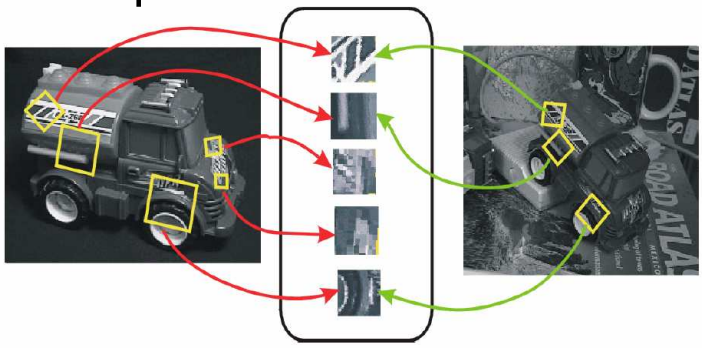


Keypoint Descriptors

- Vediamo ora come associare a ciascun keypoint un **descrittore** che caratterizzi in maniera distintiva la feature trovata al di là delle possibili trasformazioni che essa possa subire tra diverse immagini.
- Come già osservato, il descriptor viene calcolato relativamente a posizione, orientamento e scala del keypoint: ciò garantisce **invarianza a traslazioni, rotazioni e ridimensionamenti**.

Local Patch

- Il descriptor più semplice (ma comunque valido in alcune applicazioni) consiste nell'insieme dei valori della **patch locale**, calcolata rispetto al keypoint e riscalata a dimensione fissa (a volte molto piccola per avere bassa dimensionalità).



Local Patch

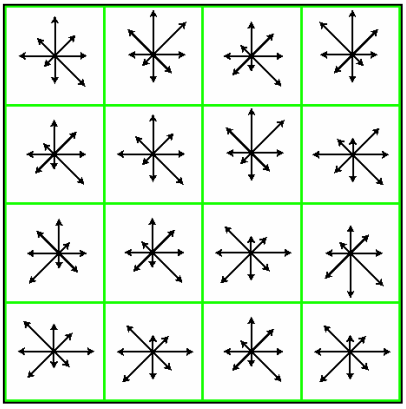
- Come misura di similarità tra patch si può usare la loro **correlazione:**

$$r_{I_1,I_2} = \frac{1}{n} \frac{\sum_{x,y} (I_1 - \overline{I_1})(I_2 - \overline{I_2})}{\sigma_{I_1} \sigma_{I_2}}$$

- Sottraendo la media e dividendo per la deviazione standard si ottiene **invarianza a cambiamenti lineari di luminosità e contrasto**.

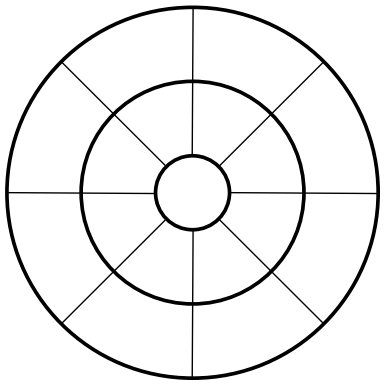
SIFT Descriptor

- Reticolo di **istogrammi di orientamenti dei gradienti locali**.
- Introdotto come descrittore per la tecnica Scale-Invariant Feature Transform, viene usato molto diffusamente anche con altri detector.
- 128-dimensionale.



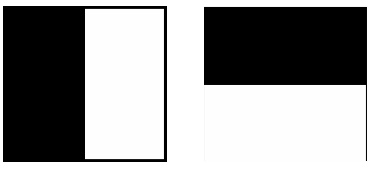
GLOH Descriptor

- Gradient Location-Orientation Histogram.
- Anch'esso è un insieme di istogrammi di orientamenti dei gradienti locali.
- È una variante del descrittore SIFT: **il reticolo ha una forma a "bersaglio"** anziché essere quadrato.
- 272-dimensionale.

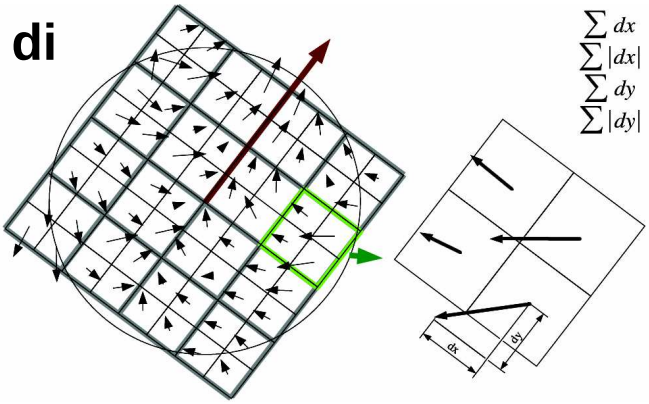


SURF Descriptor

- Speeded Up Robust Features.



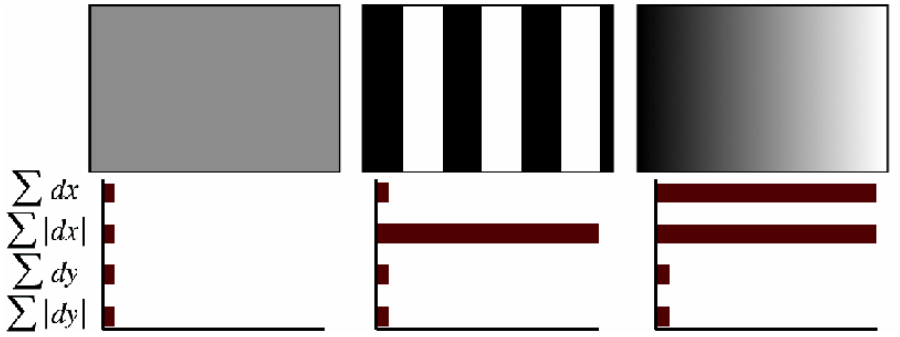
- È basato sulle **wavelet di Haar**.



- Viene considerato un reticolo 20x20 diviso in 4x4 sottoregioni.
- 64-dimensionale.

SURF Descriptor

- Per ciascuna sottoregione vengono integrate le risposte delle **wavelet orizzontali e verticali** (rispetto al keypoint) e i loro **valori assoluti**.



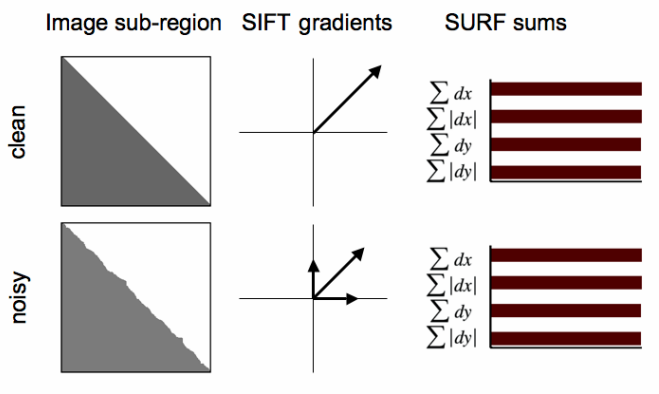
SURF Descriptor

- Come nel caso del descrittore SIFT, i valori accumulati nelle sottoregioni sono **pesati con una gaussiana** centrata nel keypoint.
- Infine il vettore 64-dimensionale risultante viene normalizzato per ottenere **invarianza a cambiamenti di contrasto** (le wavelet sono già invarianti a cambiamenti di luminosità).

Low Level Vision
Keypoint detectors/descriptors

SURF Descriptor

- Per certi versi è simile al descrittore SIFT, ma il fatto di non considerare individualmente la direzione dei gradienti permette di **filtrare meglio il rumore**:

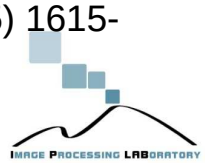


Descriptor: varie (cenni)

- È possibile includere le **informazioni sul colore** calcolando il descriptor per ciascun canale (la dimensionalità triplica: 384 per il SIFT).
- Per ridurre la dimensionalità, si può applicare la **PCA (Principal Component Analysis)** al descrittore, utilizzando un dataset di riferimento per calcolare in anticipo l'autospazio da usare.
- Uno dei filoni di studio è l'impiego di **banchi di filtri**, inizialmente ideati per altre applicazioni, per costruire un vettore di risposte da usare come descriptor.

References

- Witkin, A.P. 1983. Scale-space filtering. In International Joint Conference on Artificial Intelligence,Karlsruhe, Germany, pp. 1019-1022.
- Koenderink, J.J. 1984. The structure of images. Biological Cybernetics, 50:363-396.
- Lindeberg, T. 1994. Scale-space theory: A basic tool for analysing structures at different scales. Journal of Applied Statistics, 21(2):224-270.
- Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision Vol. 60(2), p.9.
- Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision 60 (2004) 63-86.
- Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1615-1630.



References

- Harris, C. and Stephens, M. 1988. A combined corner and edge detector. In Fourth Alvey Vision Conference, Manchester, UK, pp. 147-151.
- Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008
- Matas, J., Chum, O., Urba, M., and Pajdla, T. "Robust wide baseline stereo from maximally stable extremal regions." Proc. of British Machine Vision Conference, pages 384-396, 2002.