

# Traitement de l'image et du signal

## Partie TI

Emanuel Aldea <emanuel.aldea@u-psud.fr>  
<http://hebergement.universite-paris-saclay.fr/emi/453>

Master Electronique, énergie électrique, automatique 1<sup>ère</sup> année

# Nécessité de l'invariance en TI

## Les contours

- ▶ traitement relativement peu coûteux
- ▶ détection robuste de courbes paramétriques (Hough)

# Nécessité de l'invariance en TI

## Les contours

- ▶ traitement relativement peu coûteux
- ▶ détection robuste de courbes paramétriques (Hough)
- ▶ applications variées dans des environnements spécifiques :
  - ▶ détection de la route, des panneaux, du texte
  - ▶ imagerie médicale et satellitaire
  - ▶ inspection par vision industrielle

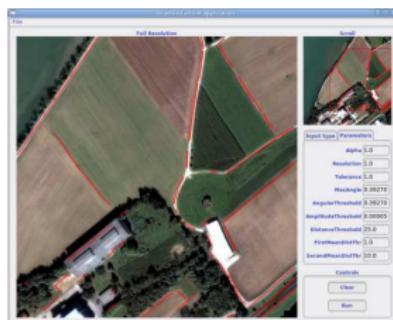
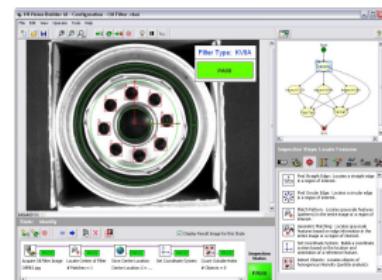


Image aérienne



Détection de voies



Vision industrielle

# Nécessité de l'invariance en TI

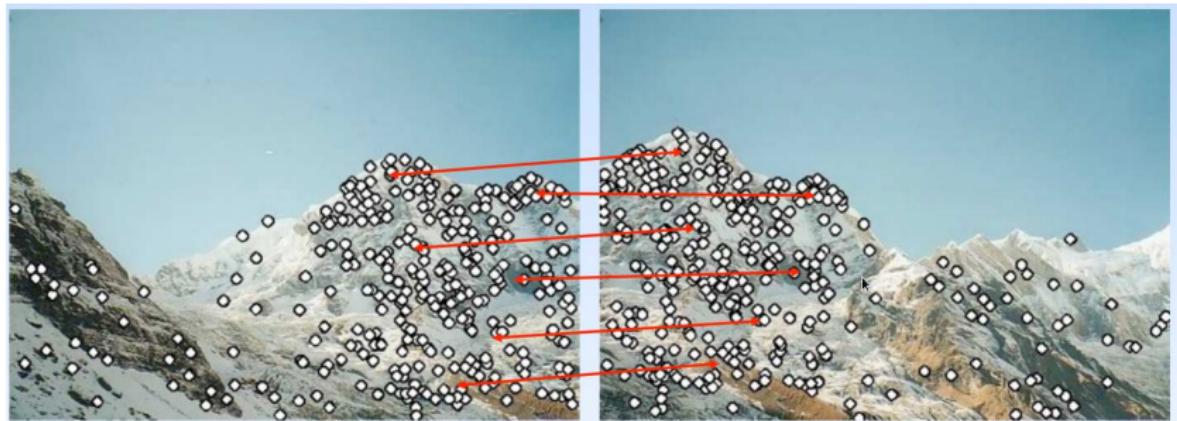
## Les contours

- ▶ traitement relativement peu coûteux
  - ▶ détection robuste de courbes paramétriques (Hough)
  - ▶ applications variées dans des environnements spécifiques :
    - ▶ détection de la route, des panneaux, du texte
    - ▶ imagerie médicale et satellitaire
    - ▶ inspection par vision industrielle
- 
- ✓ tâches rapides et spécialisées
  - ✓ invariants aux variations d'intensité
  - ✗ sensibles aux autres transformations géométriques
  - ✗ problème pour la reconnaissance de formes

## Motivation - images panoramiques



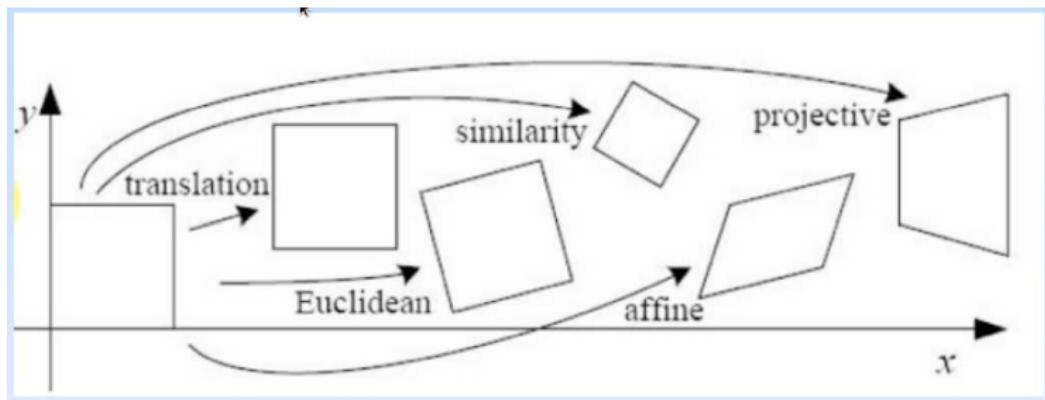
# Motivation - images panoramiques



## Motivation - images panoramiques



# Le problème



- ▶ translation
- ▶ Euclidienne (translation + rotation)
- ▶ similarité (tr. + rot. + échelle)
- ▶ affine (rot. + échelle + shear + translation)
- ▶ projective

# Nécessité de l'invariance en TI

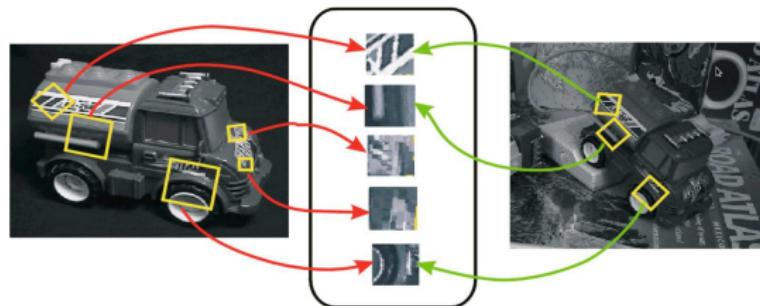
## Objectif

- ▶ identifier des structures qui sont **invariantes** par rapport aux rotations, changements d'échelle, etc.
- ▶ ces structures sont appelées couramment **points d'intérêt** ou **coins**

# Nécessité de l'invariance en TI

## Objectif

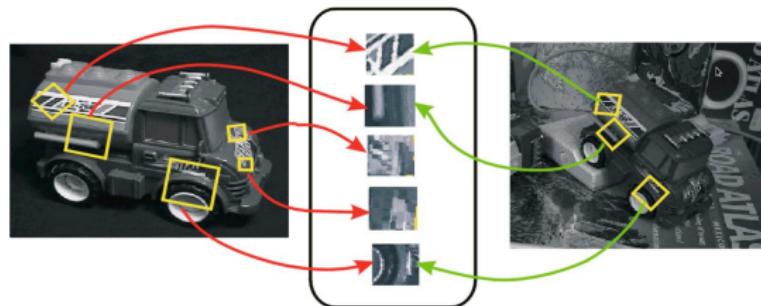
- ▶ identifier des structures qui sont **invariantes** par rapport aux rotations, changements d'échelle, etc.
- ▶ ces structures sont appelées couramment **points d'intérêt** ou **coins**



# Nécessité de l'invariance en TI

## Objectif

- ▶ identifier des structures qui sont **invariantes** par rapport aux rotations, changements d'échelle, etc.
- ▶ ces structures sont appelées couramment **points d'intérêt** ou **coins**



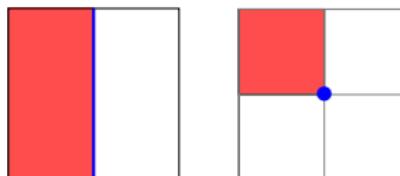
## Comment faire pour :

- ▶ les identifier de manière non supervisée ?
- ▶ les associer de manière robuste ?

# DéTECTEURS de coins : les bases

## Définition

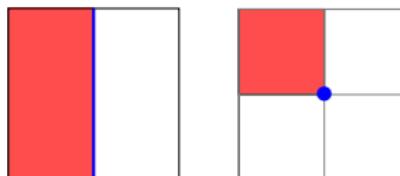
Coin : un endroit de l'image qui présente une forte variation d'intensité en deux directions différentes.



# DéTECTEURS de coins : les bases

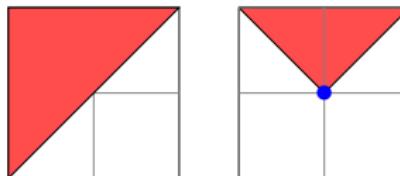
## Définition

Coin : un endroit de l'image qui présente une forte variation d'intensité en deux directions différentes.



Toujours besoin de calculer les gradients locaux, mais

- ▶ pas suffisant de faire comme auparavant (dans le repère image) !



## DéTECTEURS de coins : les bases

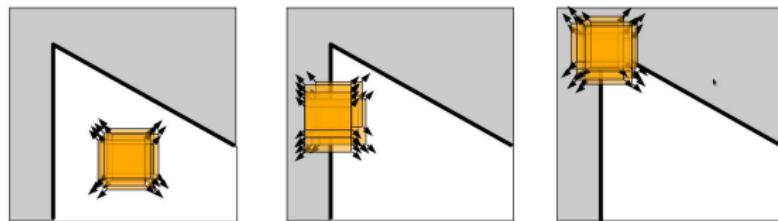
### Définition

Stratégie : le contenu d'un patch centré dans le coin devrait varier dans toutes les directions

# DéTECTEURS de coins : les bases

## Définition

Stratégie : le contenu d'un patch centré dans le coin devrait varier dans toutes les directions



## Comportement :

- ▶ régions homogènes : pas de changement
- ▶ contours : pas de changement le long du contour
- ▶ coins : changement important dans toutes les directions
- ▶ qualité du pixel : le plus petit changement
- ▶ introduit par Moravec en 1980
- ▶ pas isotrope : certains contours mal détectés, fausse réponse coin

## DéTECTEURS de coins : les bases

Changement d'intensité par shift de  $(\Delta x, \Delta y)$

$$E(x, y, \Delta x, \Delta y) = \sum_x \sum_y w(x, y) [I(x, y) - I(x + \Delta x, y + \Delta y)]^2$$

## DéTECTEURS de coins : les bases

Changement d'intensité par shift de  $(\Delta x, \Delta y)$

$$E(x, y, \Delta x, \Delta y) = \sum_x \sum_y w(x, y) \left[ \underbrace{I(x, y)}_{\text{intensité}} - I(x + \Delta x, y + \Delta y) \right]^2$$

## DéTECTEURS de coins : les bases

Changement d'intensité par shift de  $(\Delta x, \Delta y)$

$$E(x, y, \Delta x, \Delta y) = \sum_x \sum_y w(x, y) \left[ \underbrace{I(x, y)}_{\text{intensité}} - \underbrace{I(x + \Delta x, y + \Delta y)}_{\text{intensité shiftée}} \right]^2$$

## DéTECTEURS de coins : les bases

Changement d'intensité par shift de  $(\Delta x, \Delta y)$

$$E(x, y, \Delta x, \Delta y) = \sum_x \sum_y \underbrace{w(x, y)}_{\text{support}} \left[ \underbrace{I(x, y)}_{\text{intensité}} - \underbrace{I(x + \Delta x, y + \Delta y)}_{\text{intensité shiftée}} \right]^2$$

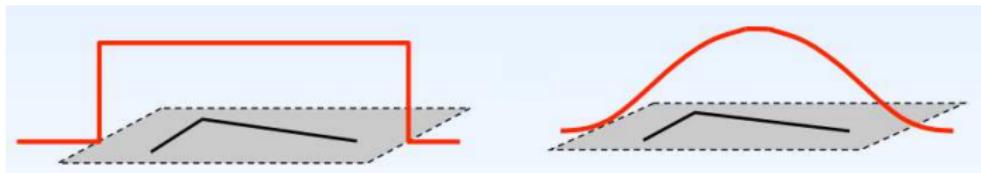


FIGURE – Types de fonction support  $w(x, y)$

$E(x, y)$  large indique potentiellement un coin.

## DéTECTEURS de coins : les bases

Approximation au premier ordre du dév. en série Taylor

$$f(x + \Delta x, y + \Delta y) = f(x, y) + f_x(x, y)\Delta x + f_y(x, y)\Delta y$$

On l'utilise pour réécrire la variation d'intensité par shift :

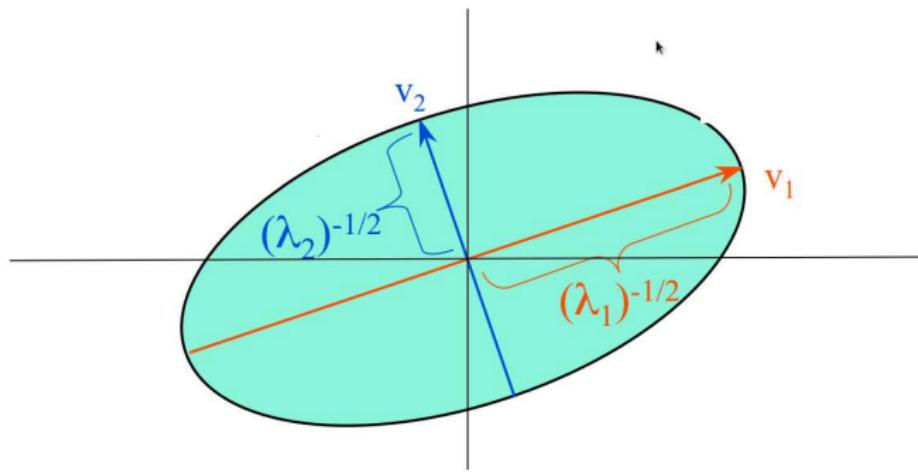
$$\begin{aligned} \sum [I(x + \Delta x, y + \Delta y) - I(x, y)]^2 &\approx \sum [I(x, y) + \Delta x I_x(x, y) + \Delta y I_y(x, y) - I(x, y)]^2 \\ &\approx \sum \Delta x^2 I_x^2 + 2\Delta x \Delta y I_x I_y + \Delta y^2 I_y^2 \\ &\approx [\Delta x \Delta y] \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &\approx [\Delta x \Delta y] \left( \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned}$$

$$\begin{aligned} E(x, y, \Delta x, \Delta y) &\approx [\Delta x \Delta y] \left( \sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &\approx [\Delta x \Delta y] \underbrace{\begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}}_{\text{tenseur de structure}} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned}$$

# DéTECTEURS de coins : le tenseur de structure

## Propriétés

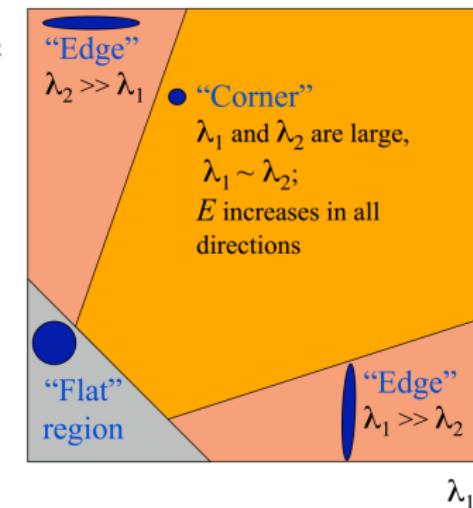
- ▶ les vecteurs propres indiquent les directions principales de variation de gradient dans le voisinage du point (voir l'ellipse du changement constant)
- ▶ ex. : si  $\lambda_2 > \lambda_1$ , variation forte en  $v_2$  et plus faible en  $v_1$
- ▶ si coin,  $\lambda_1, \lambda_2$  sont larges



# DéTECTEURS de coins : le tenseur de structure

## Propriétés

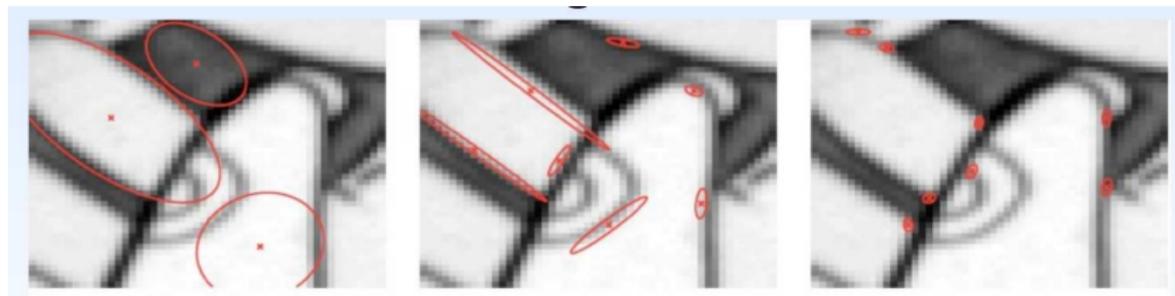
- les vecteurs propres indiquent les direction principales de variation de gradient dans le voisinage du point (voir l'ellipse du changement constant)
- ex. : si  $\lambda_2 > \lambda_1$ , variation forte en  $v_2$  et plus faible en  $v_1$
- si coin,  $\lambda_1, \lambda_2$  sont larges



# DéTECTEURS DE COINS : le tenseur de structure

## Propriétés

- ▶ les vecteurs propres indiquent les direction principales de variation de gradient dans le voisinage du point (voir l'ellipse du changement constant)
- ▶ ex. : si  $\lambda_2 > \lambda_1$ , variation forte en  $v_2$  et plus faible en  $v_1$
- ▶ si coin,  $\lambda_1, \lambda_2$  sont larges



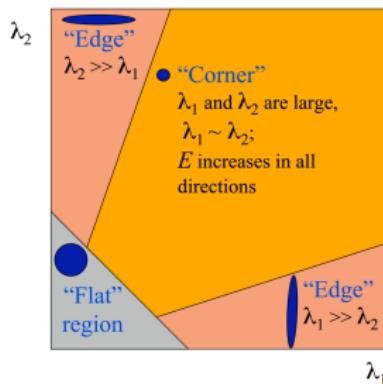
## DéTECTEURS de coins : décision

Décision en fonction des valeurs propres du tenseur

- ▶ on peut calculer  $\lambda_1, \lambda_2$  explicitement mais trop lourd
- ▶ méthode préférée :

$$R = \det(M) - \alpha \text{trace}^2(M) = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

- ▶ valeur du paramètre  $\alpha$  en général 0.04 - 0.06
- ▶ valeurs propres intéressantes = maximum local de  $R$



## DéTECTEURS de coins : rappel

### Étapes de l'algorithme

1. calcul des gradients  $I_x = \frac{\partial}{\partial x} g(\sigma_D) * I$ ,  $I_y = \frac{\partial}{\partial y} g(\sigma_D) * I$
2. calcul du tenseur de structure :

$$M = g(\sigma_I) * \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

3. calcul de la fonction de réponse  $R$  :

$$R = \det(M) - \alpha \text{trace}^2(M)$$

4. seuillage de  $R$
5. suppression de valeurs non maximales de  $R$

## DéTECTEURS de coins : exemple



FIGURE – Paire initiale

## DéTECTEURS de coins : exemple

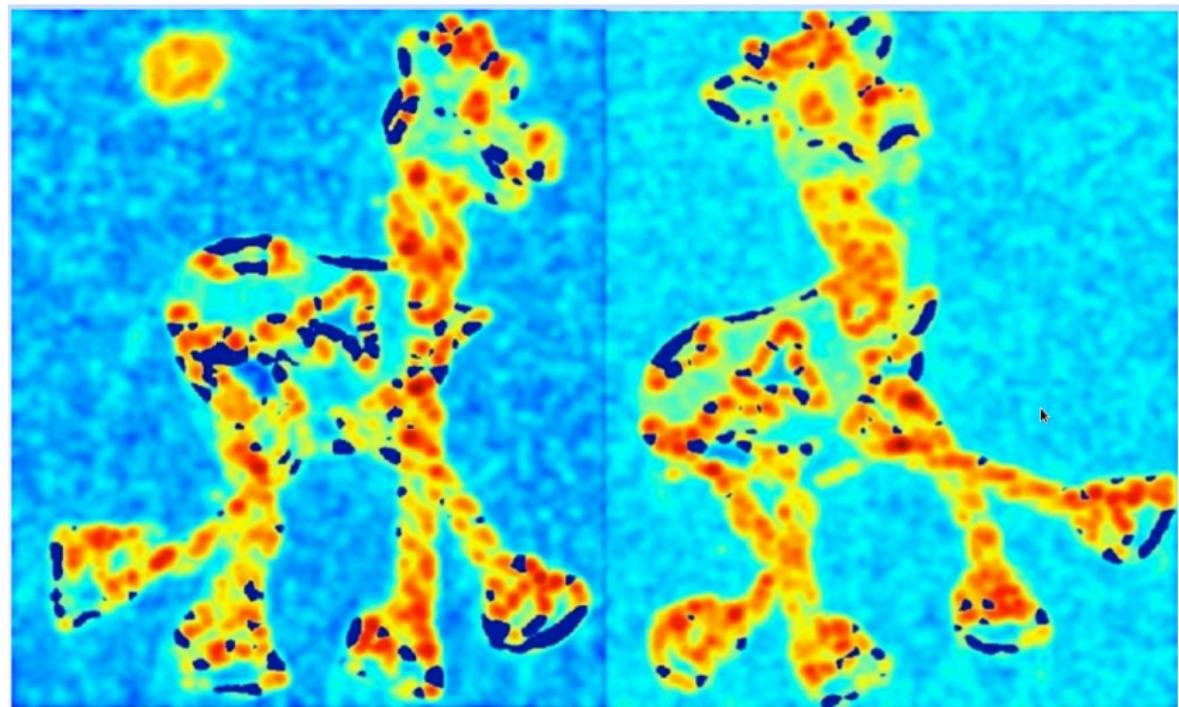


FIGURE – Fonction de réponse  $R$

## DéTECTEURS de coins : exemple



FIGURE – Seuillage de  $R$

## DéTECTEURS de coins : exemple

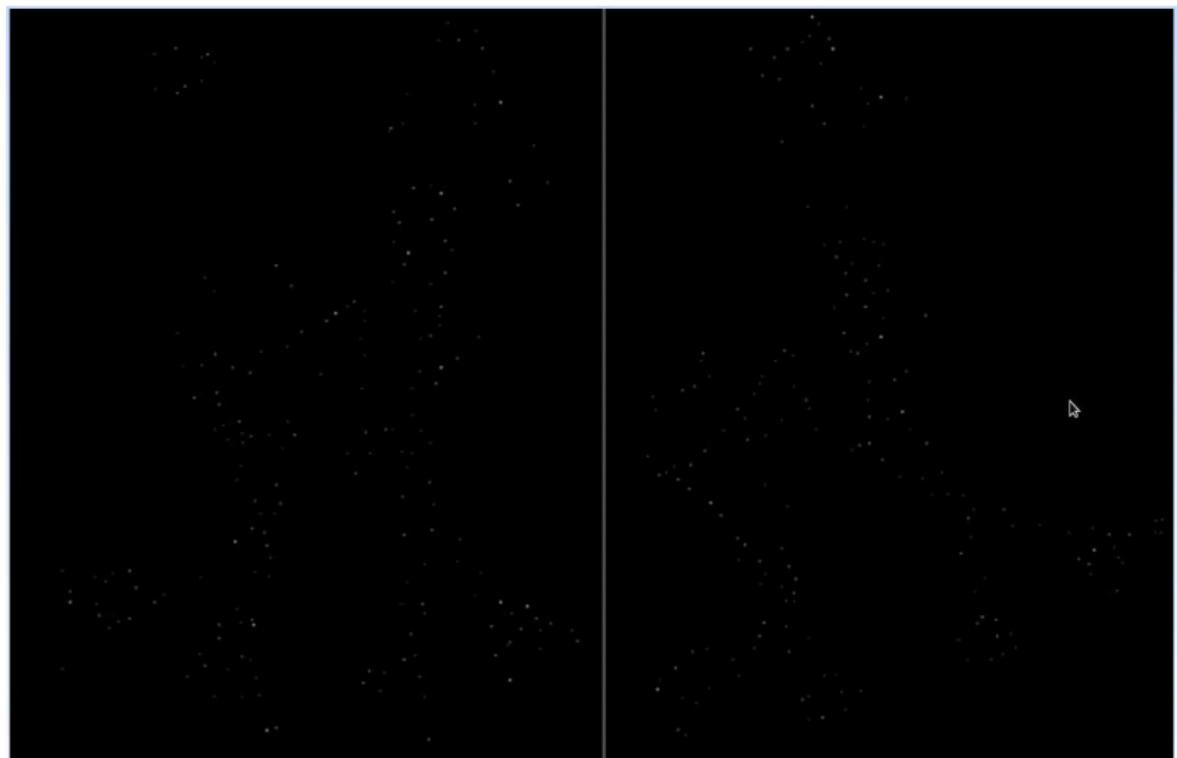


FIGURE – Suppression non maximale de  $R$

## DéTECTEURS de coins : exemple

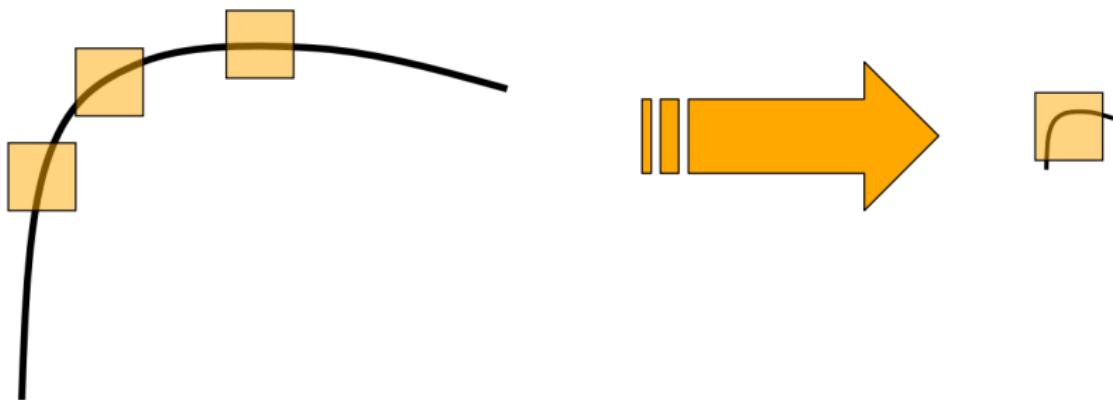


FIGURE – Résultat détection

# Détecteur de Harris

## Conclusions

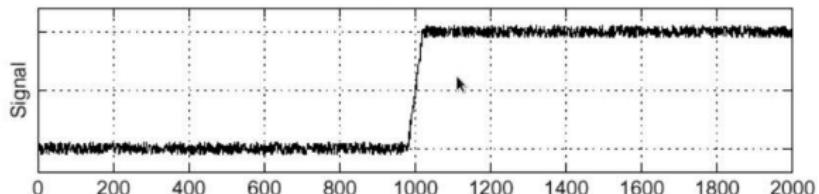
- ✓ détection invariante à la rotation
- ✓ détection invariante aux changements d'intensité
- ✗ pas robuste au changement d'échelle



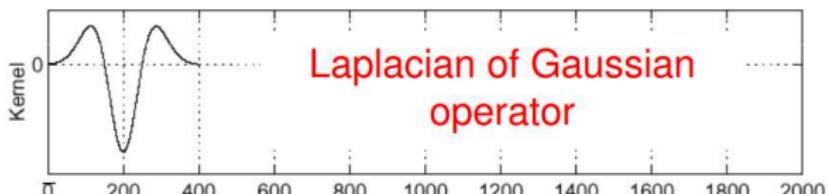
# L'échelle caractéristique

Rappel du cours précédent :

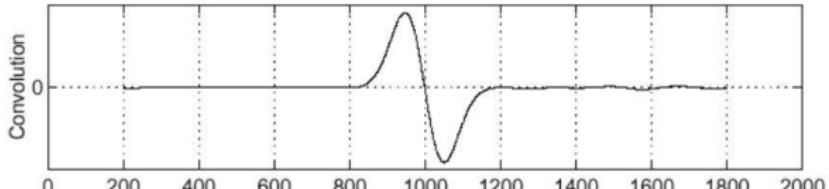
$f$



$$\frac{\partial^2}{\partial x^2} h$$

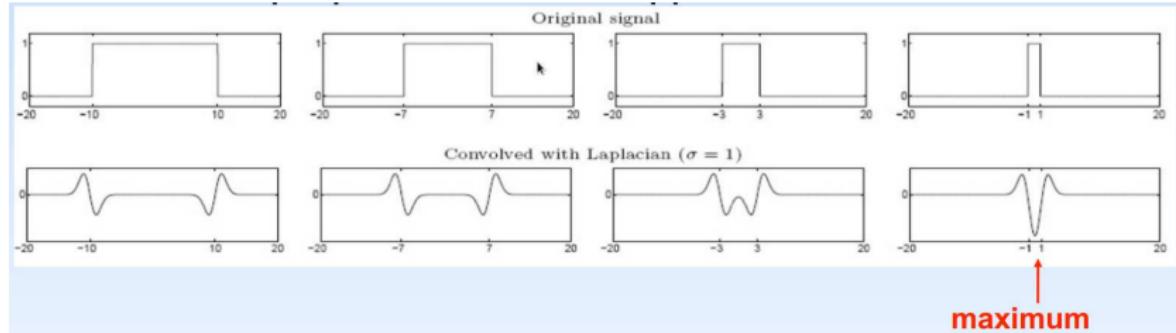


$$\left( \frac{\partial^2}{\partial x^2} h \right) * f$$



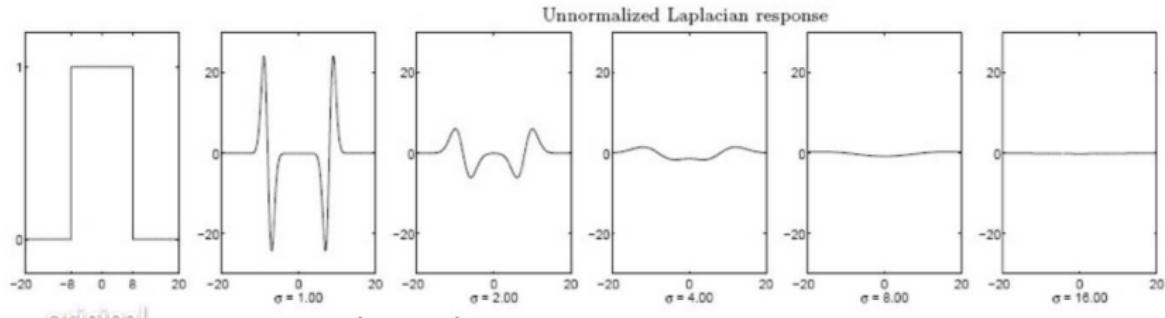
Filtrage Gaussien + Laplace (LoG) - passage en 0

# L'échelle caractéristique



La réponse au Laplacien - maximale si l'échelle du Laplacien correspond à l'échelle de la variation dans l'espace image

# L'échelle caractéristique



Si on varie  $\sigma$ , la réponse du Laplacien diminue ; il faut normaliser le résultat par une multiplication avec  $\sigma^2$

# L'échelle caractéristique

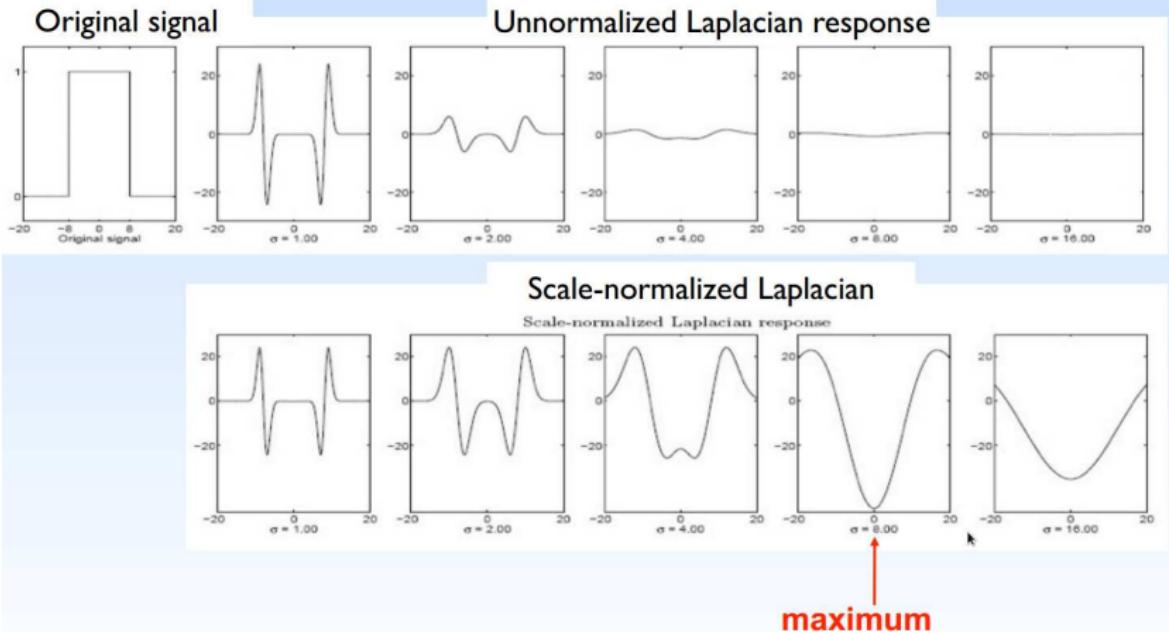
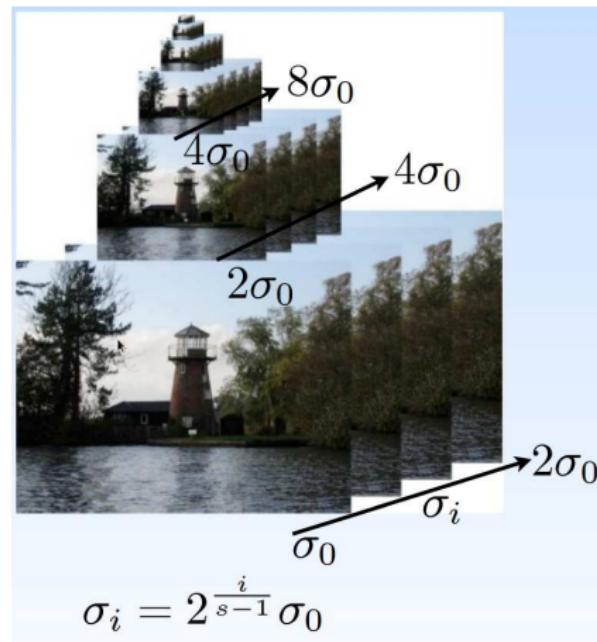
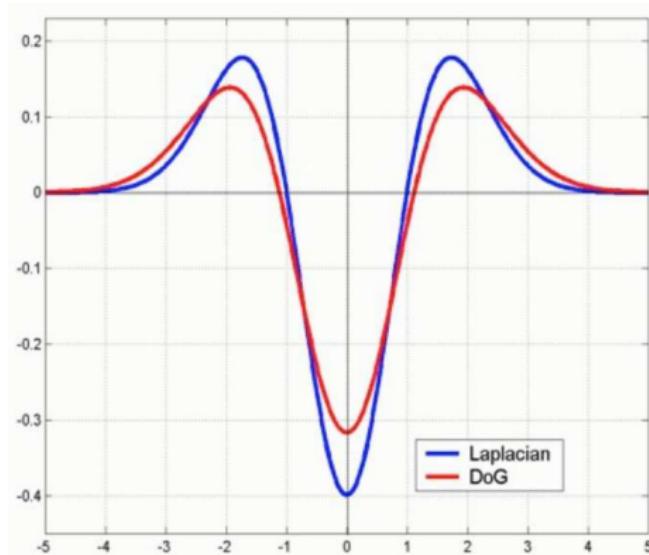


FIGURE – Réponse multi échelle normalisée du Laplacien

# La représentation pyramidale



# Approximation du Laplacien



Laplacien :

$$L = \sigma^2(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$

Différence de Gaussiennes :

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

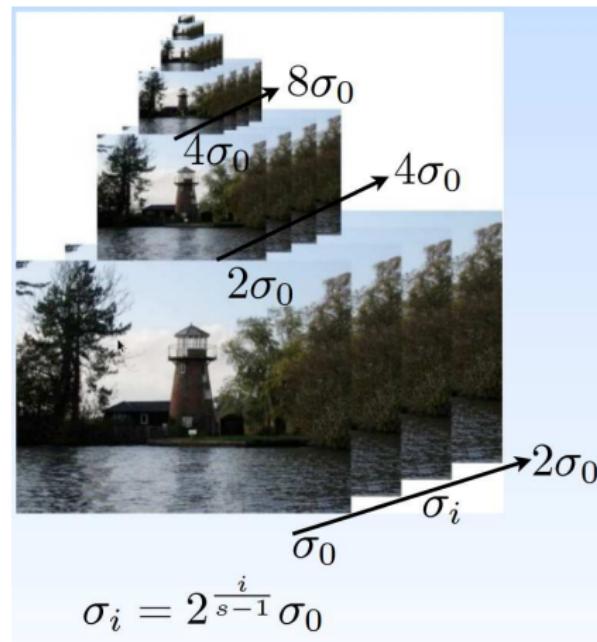
# Le détecteur SIFT

## Scale Invariant Feature Transform

- ▶ très performant
- ▶ très coûteux
- ▶ descripteur intégré

1. Construction de l'espace d'échelle
2. Calcul des DoGs
3. Calcul de l'échelle caractéristique
4. Localisation sous-pixel
5. Élimination des réponses des contours
6. Calcul de l'orientation
7. Calcul du descripteur

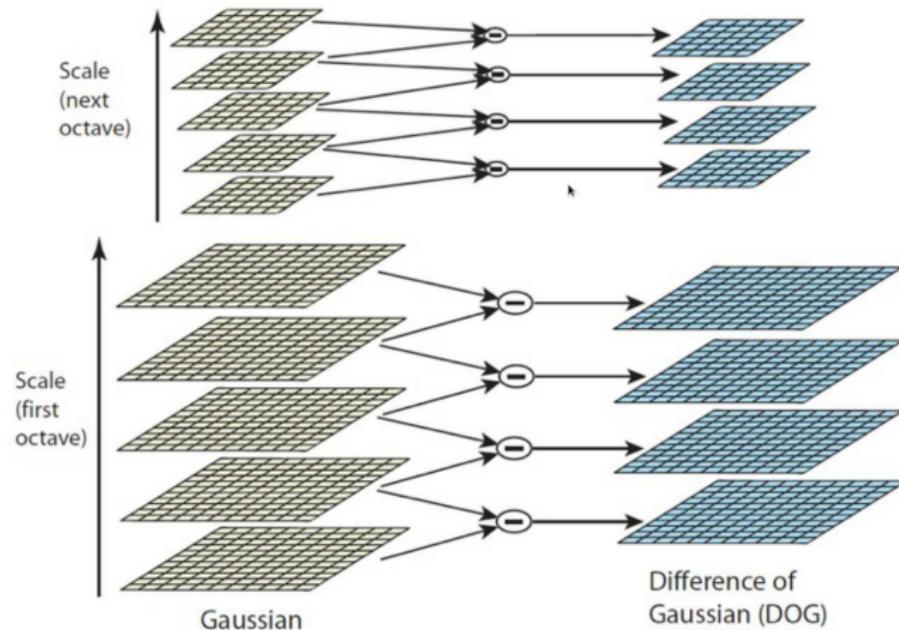
# La représentation pyramidale



# Le détecteur SIFT

1. Construction de l'espace d'échelle
2. Calcul des DoGs
3. Calcul de l'échelle caractéristique
4. Localisation sous-pixel
5. Élimination des réponses des contours
6. Calcul de l'orientation
7. Calcul du descripteur

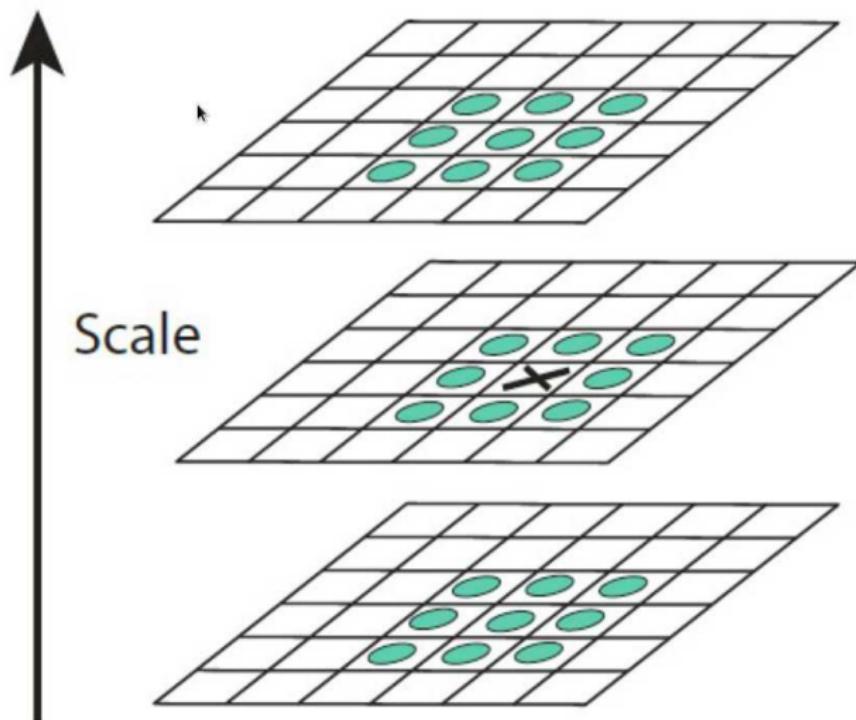
# Calcul des DoGs



# Le détecteur SIFT

1. Construction de l'espace d'échelle
2. Calcul des DoGs
3. Calcul de l'échelle caractéristique
4. Localisation sous-pixel
5. Élimination des réponses des contours
6. Calcul de l'orientation
7. Calcul du descripteur

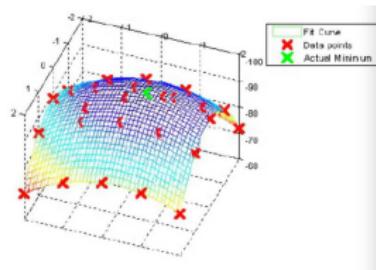
# Détermination des points extrêmes



# Le détecteur SIFT

1. Construction de l'espace d'échelle
2. Calcul des DoGs
3. Calcul de l'échelle caractéristique
4. **Localisation sous-pixel**
5. Élimination des réponses des contours
6. Calcul de l'orientation
7. Calcul du descripteur

# Localisation sous pixel



Interpolation des mesures discrètes de  $D(x, y, \sigma)$ . Utilisation du dév. Taylor de deuxième ordre :

$$D(\hat{x}) = D + \frac{\partial D}{\partial x}^T \hat{x} + \frac{1}{2} \hat{x}^T \frac{\partial^2 D}{\partial x^2} \hat{x}$$

Solution :

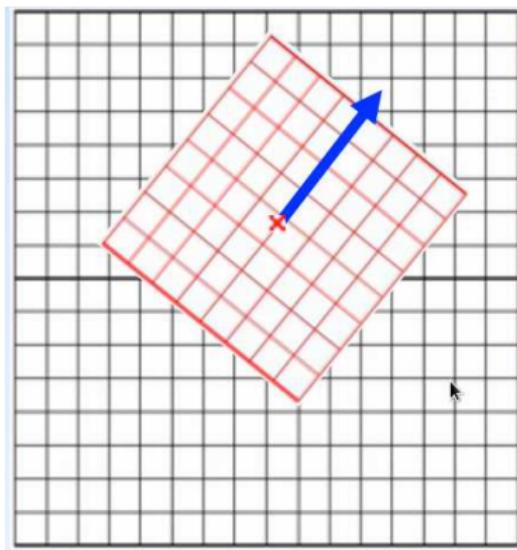
$$\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1} \frac{\partial D}{\partial x}$$

# Le détecteur SIFT

1. Construction de l'espace d'échelle
2. Calcul des DoGs
3. Calcul de l'échelle caractéristique
4. Localisation sous-pixel
5. Élimination des réponses des contours
6. **Calcul de l'orientation**
7. Calcul du descripteur

# Calcul de l'orientation

1. Calcul des gradients locaux à l'échelle caractéristique
2. Calcul de l'histogramme de gradients locaux
3. L'orientation canonique est la direction maximale
4. Chaque coin est caractérisé par : localisation, échelle, orientation
5. système de coordonnées local pour construire un descripteur

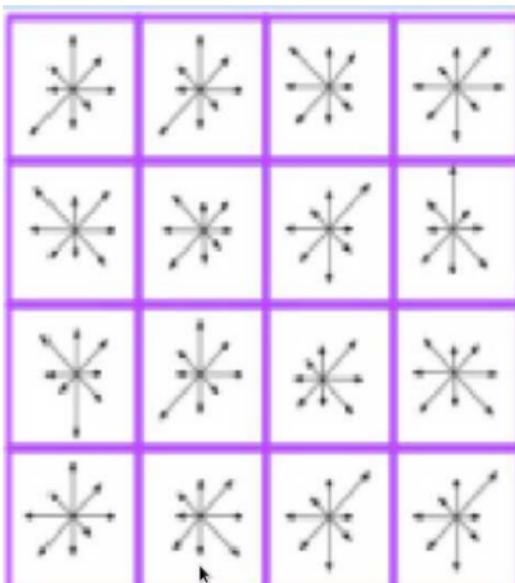


# Le détecteur SIFT

1. Construction de l'espace d'échelle
2. Calcul des DoGs
3. Calcul de l'échelle caractéristique
4. Localisation sous-pixel
5. Élimination des réponses des contours
6. Calcul de l'orientation
7. Calcul du descripteur

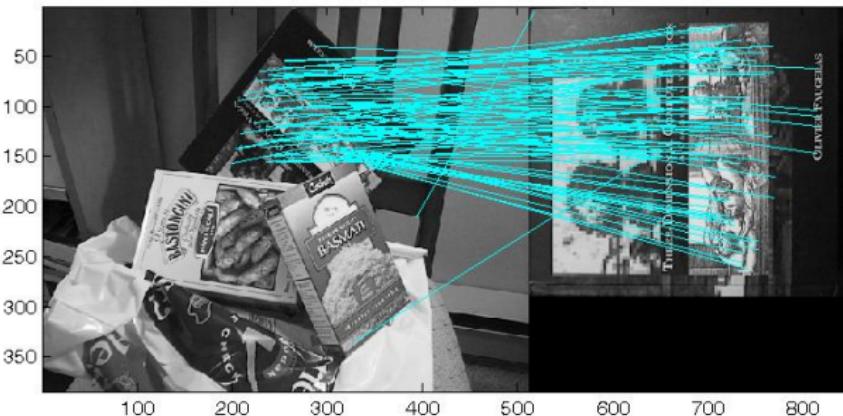
# Calcul du descripteur

1. Orientations locales du gradients en 16 régions voisines
2. Système de coordonnées relatif au coin
3.  $4*4*8$  orientations = 128 (dimension descripteur)



# Conclusion SIFT

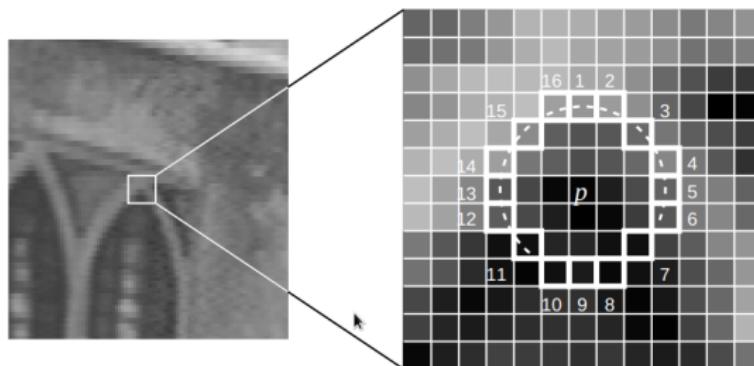
- ▶ Invariant échelle
- ▶ Invariant rotation
- ▶ Invariant illumination
- ▶ Invariant perspective
- ▶ Coûteux



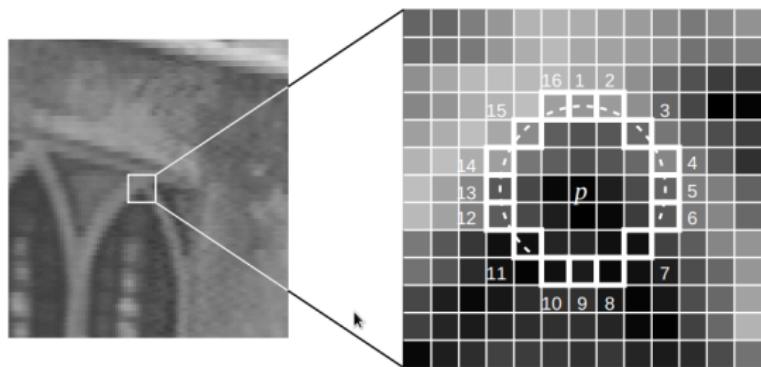
# Le détecteur FAST

## Features from Accelerated Segment Test

- ▶ extrêmement rapide
- ▶ pas d'opérations complexes (convolution, calcul de gradients etc.)
- ▶ peu robuste
- ▶ pas de descripteur



# Le détecteur FAST - stratégie



$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t \\ b, & I_p + t \leq I_{p \rightarrow x} \end{cases}$$