

# IMAGE - ANALYSE ET TRAITEMENTS D'IMAGES

## Détections d'éléments caractéristiques

# PLAN

- Introduction
- Détection
  - de points
  - de contours
  - de régions
  - d'objets

# INTRODUCTION

- Recherche d'éléments invariants
  - Mise en correspondance d'images



[extrait de 5]

# INTRODUCTION

- Recherche d'éléments invariants
  - Mise en correspondance d'images



by [Diva Sian](#)

[extrait de 5]



by [swashford](#)

# INTRODUCTION

- Recherche d'éléments invariants
  - Mise en correspondance d'images



by [Diva Sian](#)



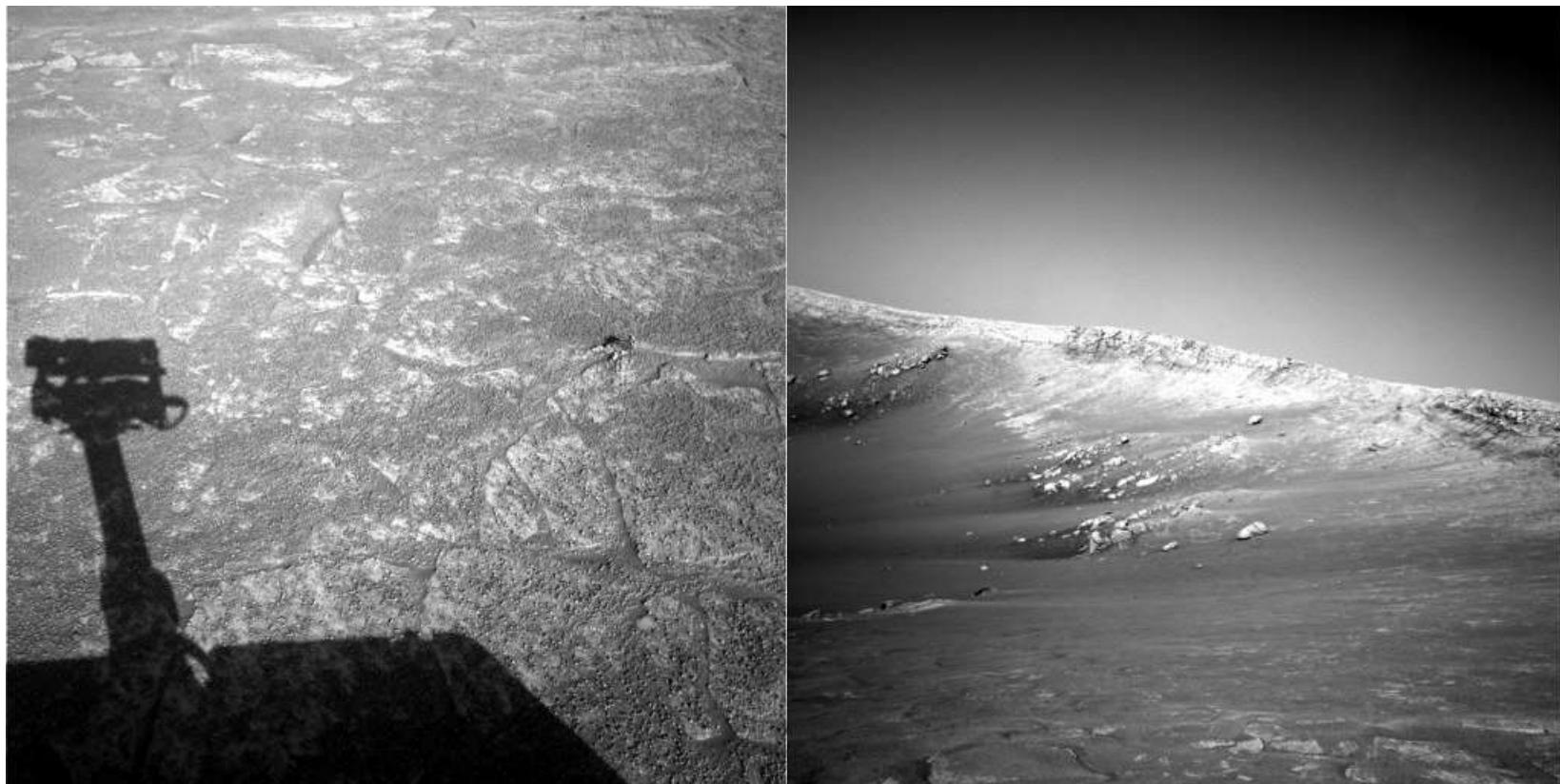
by [scgbt](#)

[extrait de 5]



# INTRODUCTION

- Recherche d'éléments invariants
  - Mise en correspondance d'images

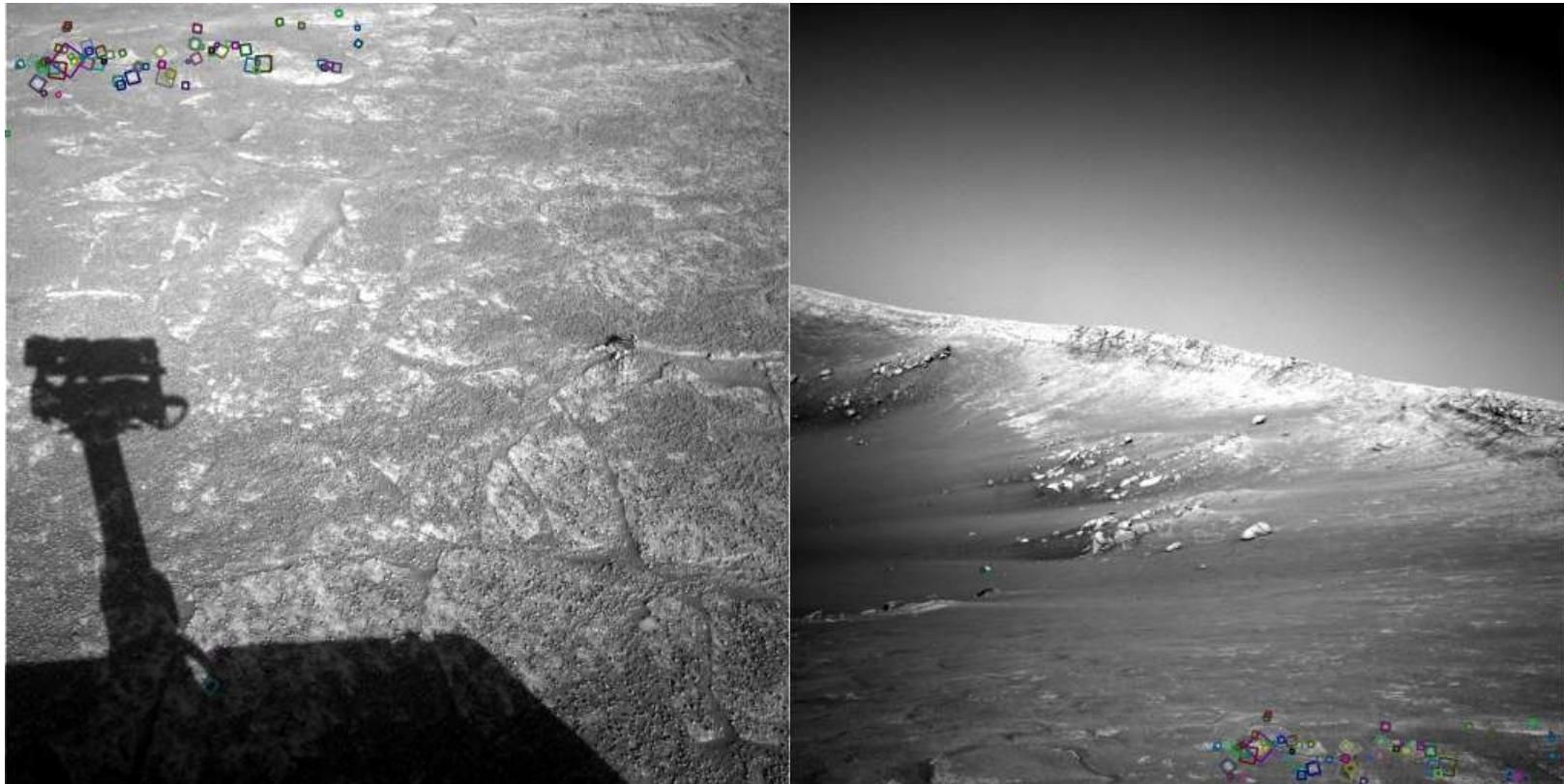


[extrait de 5]

NASA Mars Rover images

# INTRODUCTION

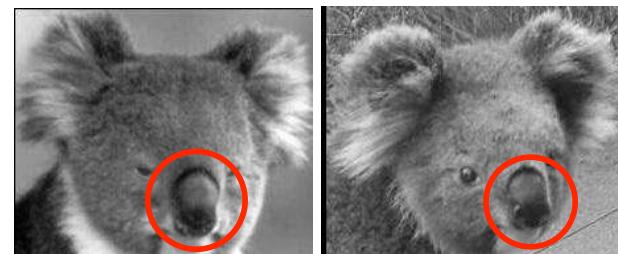
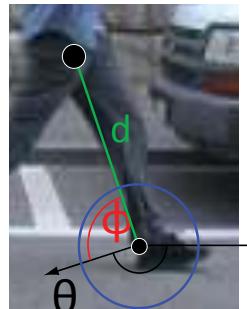
- Recherche d'éléments invariants
  - Mise en correspondance d'images



[extrait de 5] NASA Mars Rover images with SIFT feature matches  
(Figure by Noah Snavely)

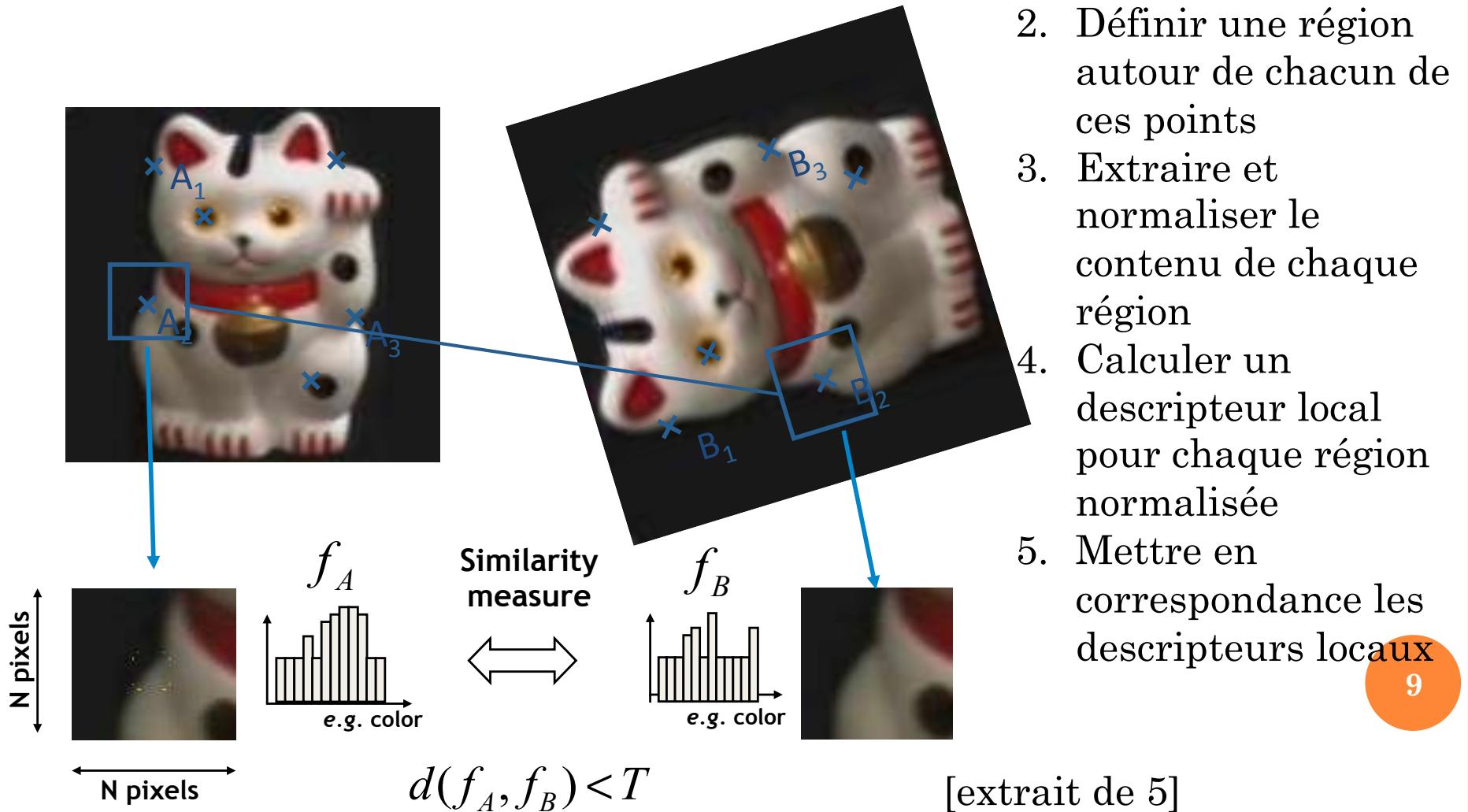
# INTRODUCTION [5]

- Motivations pour utiliser des éléments caractéristiques locaux
  - Les représentations globales ont des limitations majeures
  - A la place, description et mise en correspondance uniquement de régions locales
  - Augmente la robustesse aux
    - Occlusions
    - Déformations rigides
    - Variations intra-catégories



# INTRODUCTION [5]

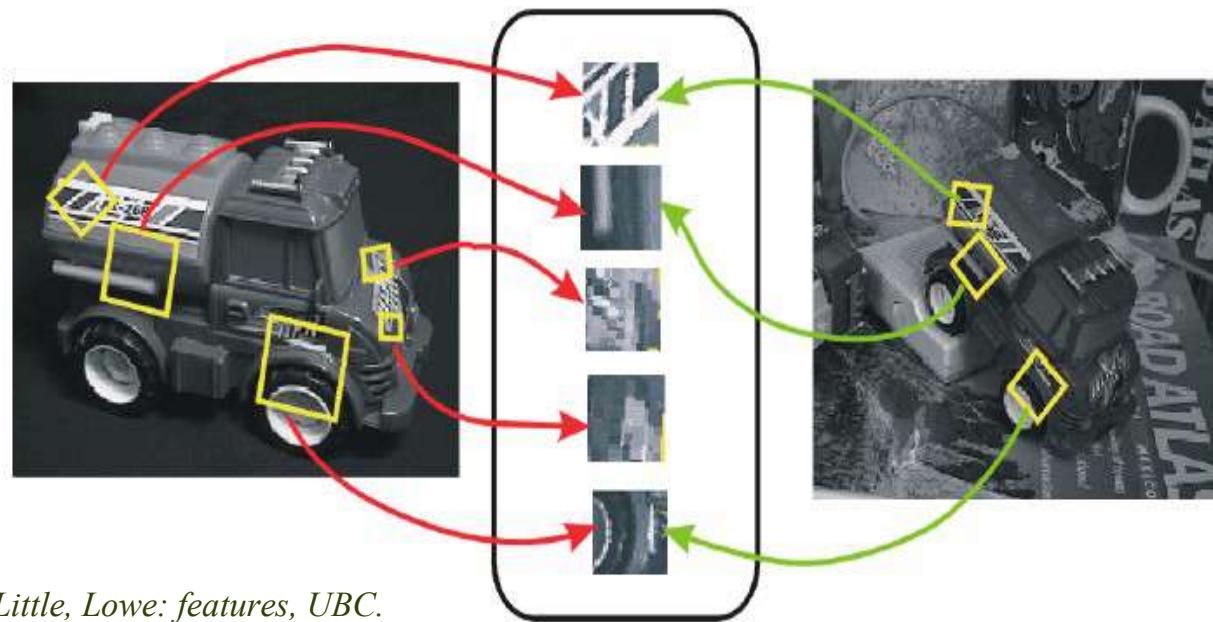
## Approche générale



# INTRODUCTION

## ○ Détection de caractéristiques - Motivations

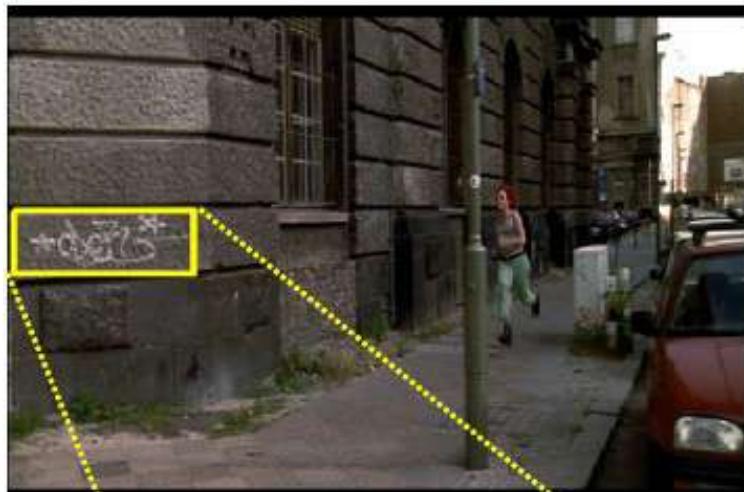
- **But** : identifier des **caractéristiques robustes** dans l'image pour mise en correspondance (3D), reconnaissance, suivi (mouvement), ...
- On cherche souvent à **éviter la segmentation** (source d'erreurs) en identifiant d'autres caractéristiques



Source : Jim Little, Lowe: features, UBC.

# INTRODUCTION

- Détection de caractéristiques - Motivations
  - Recherche d'images similaires

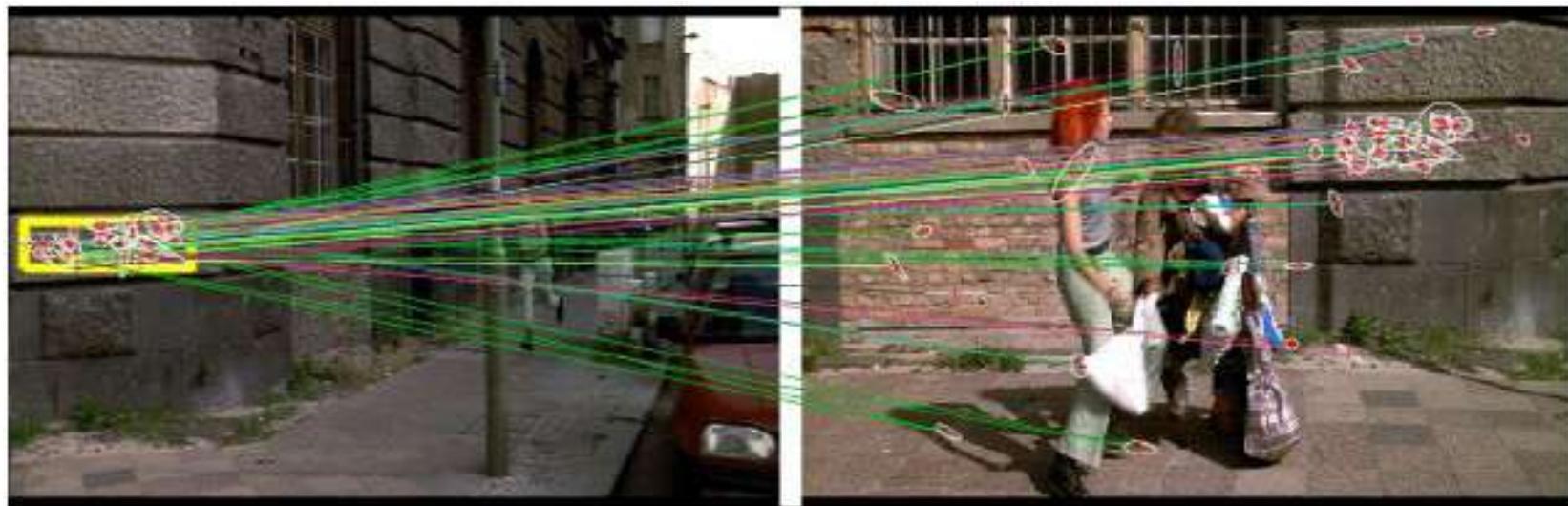


*Retrouver des images  
contenant un graffiti*

11

# INTRODUCTION

- Détection de caractéristiques - Motivations
  - Mise en correspondance d'images

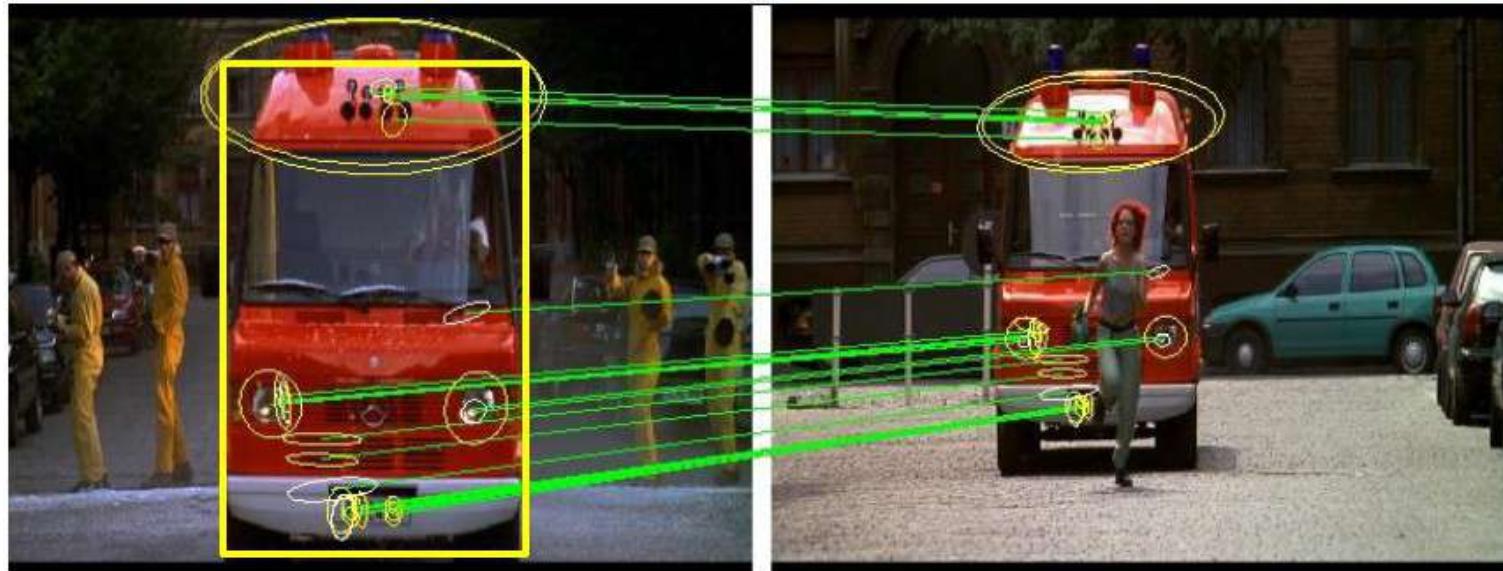


*Des scènes sont différentes (angle de vue, zoom, lumière, ...), mais on veut identifier des éléments semblables...*

12

# INTRODUCTION

- Détection de caractéristiques - Motivations
  - Recherche d'objets dans des images



*Ici, c'est le même autobus malgré la taille différente et le fait qu'il soit en partie cachée sur une image...*

# INTRODUCTION

- Détection de caractéristiques - Motivations
  - Exemple – panorama

(1) Aligner les deux images

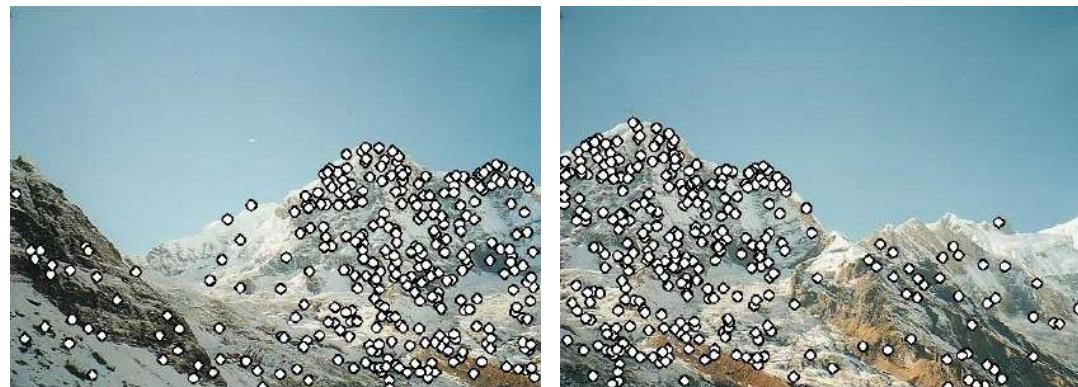


Source : Frolova & Simakov, *Invariant Feature Detectors and Descriptors*, Weizmann Ins. (Israel).

# INTRODUCTION

- Détection de caractéristiques - Motivations
  - Exemple - panorama

(2) détection des points caractéristiques dans les 2 images

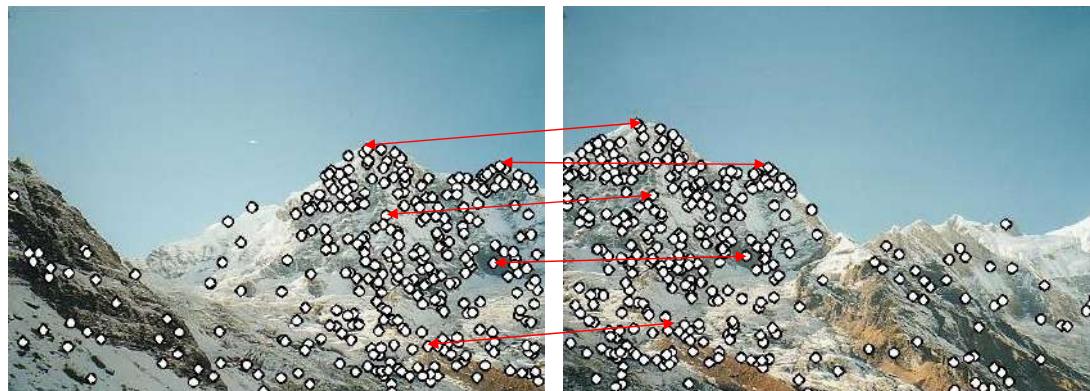


Source : Frolova & Simakov, *Invariant Feature Detectors and Descriptors*, Weizmann Ins. (Israel).

# INTRODUCTION

- Détection de caractéristiques - Motivations
  - Exemple - panorama

(3) trouver des paires de points correspondants

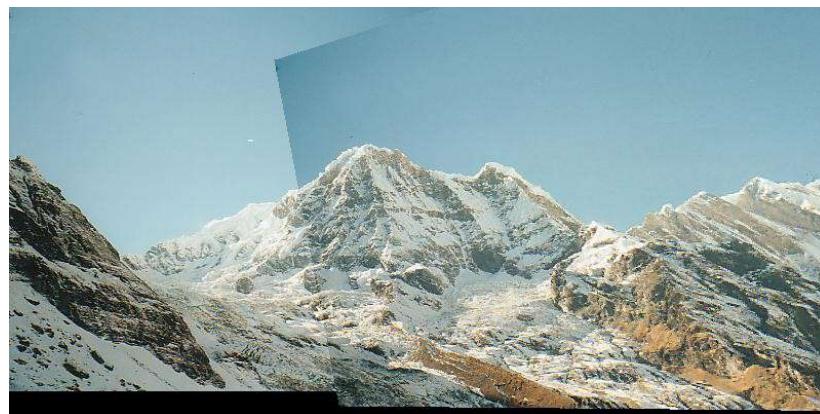


Source : Frolova & Simakov, *Invariant Feature Detectors and Descriptors*, Weizmann Ins. (Israel).

# INTRODUCTION

- Détection de caractéristiques - Motivations
  - Exemple - panorama

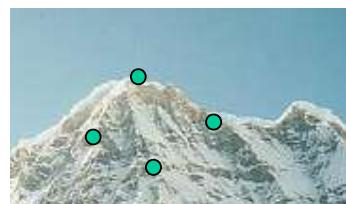
(4) Assembler les deux images pour en former une seule



Source : Frolova & Simakov, *Invariant Feature Detectors and Descriptors*, Weizmann Ins. (Israel).

# INTRODUCTION

- Détection de caractéristiques – Motivations
  - Exemple - panorama
  - Problème à résoudre no 1 :
    - Déetecter le même point dans les deux images

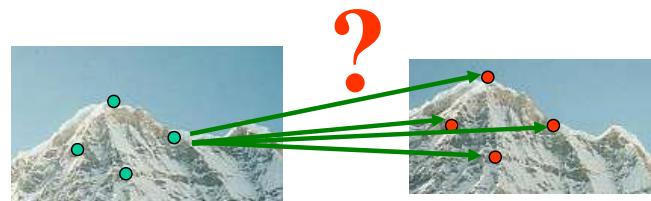


*cas impossible*

Répétabilité des points

# INTRODUCTION

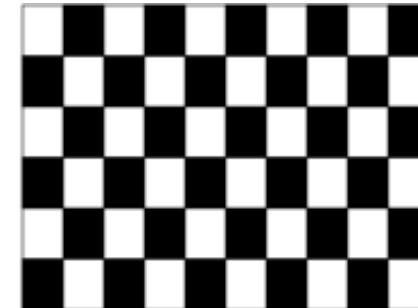
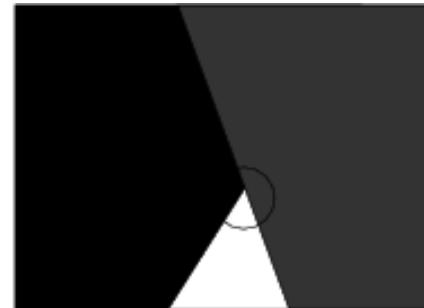
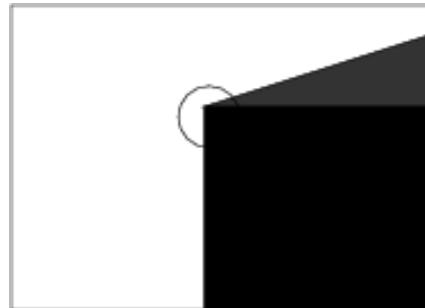
- Détection de caractéristiques - Motivations
  - Exemple - panorama
  - Problème à résoudre no 2 :
    - Pour chaque point, trouver la bonne correspondance



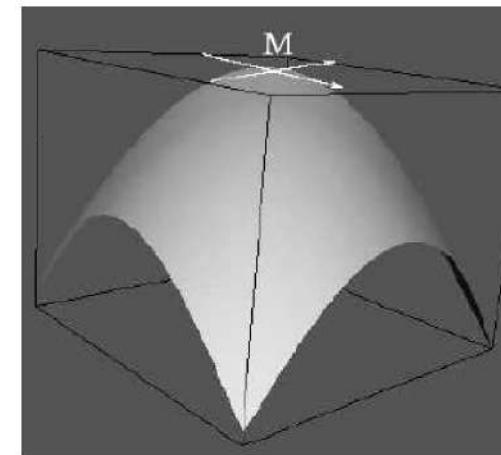
Descripteur robuste et discriminant

Source : Frolova & Simakov, *Invariant Feature Detectors and Descriptors*, Weizmann Ins. (Israel).

# QU'EST-CE QU'UN POINTS D'INTÉRÊTS [6]



- **contour** : discontinuité dans une direction de la fonction d'intensité ou de ses dérivées
- **point d'intérêt** : dans deux directions



# QU'EST-CE QU'UN POINTS D'INTÉRÊTS [6]

Avantages des points d'intérêt :

- ▶ Sources d'informations plus fiable que les contours car plus de contraintes sur la fonction d'intensité.
- ▶ Robuste aux occultations (soit occulté complètement, soit visible).
- ▶ Plus facile à extraire que les contours
- ▶ Présents dans une grande majorité d'images ( $\neq$  contours !).

# POINTS D'INTÉRÊTS – PROPRIÉTÉS RECHERCHÉES

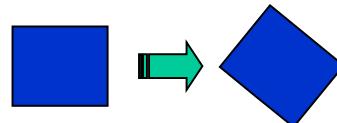
- | **Localité** : une caractéristique locale (*vs globale*) sera plus robuste aux occlusions
- | **Invariance** : à la translation, la rotation, le changement d'échelle, ...
- | **Robuste** : au bruit, aux conditions d'acquisition de l'image, à la compression, ...
- | **Discriminante** : permet d'identifier quelques objets spécifiques parmi beaucoup d'autres
- | **Quantité** : pouvoir générer beaucoup de caractéristiques par objet / portion d'images
- | **Précision** : position précise dans l'image / l'objet
- | **Efficacité** : calcul rapide (temps réel)

source : Tinne Tuytelaars, Local Invariant Features: What? Why? When? How?, ECCV 2006.

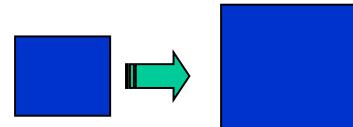
# POINTS D'INTÉRÊTS - PROPRIÉTÉS D'INVARIANCE

- Invariance à la géométrie

- **Rotation**



- **Changement d'échelle**

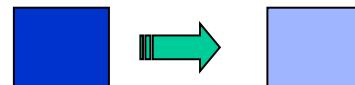


- **Transformation affine**



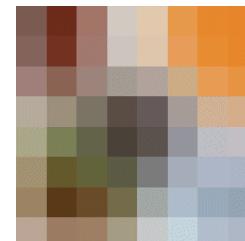
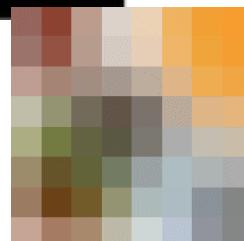
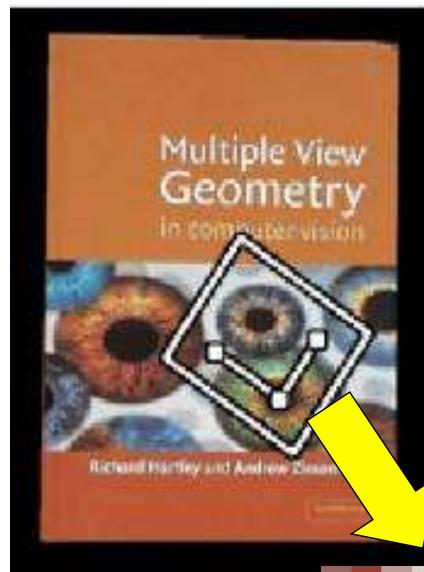
- Invariance à la photometrie

- **Changement affine d'intensité** ( $I \rightarrow aI + b$ )



# POINTS D'INTÉRÊTS - PROPRIÉTÉS D'INVARIANCE

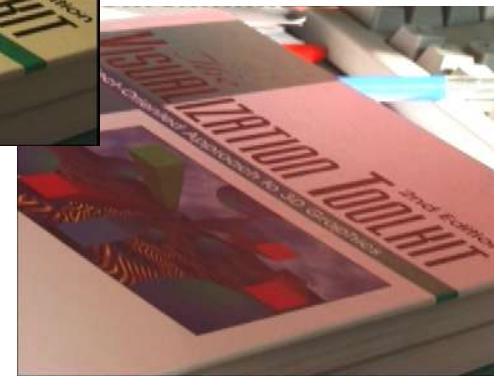
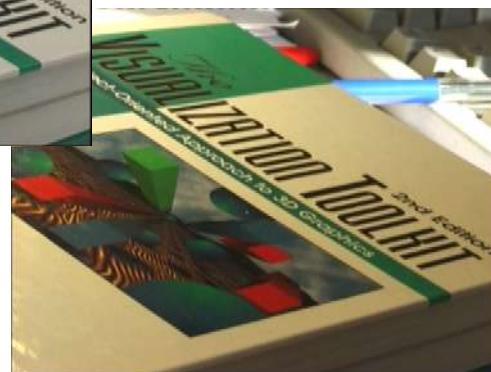
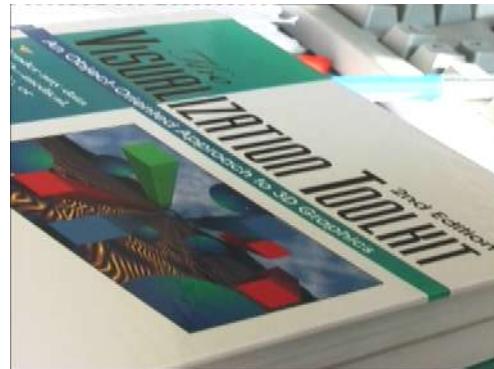
- Invariance aux transformations géométriques



[Extrait de [5]]

# POINTS D'INTÉRÊTS - PROPRIÉTÉS D'INVARIANCE

- Invariance aux transformations photométriques

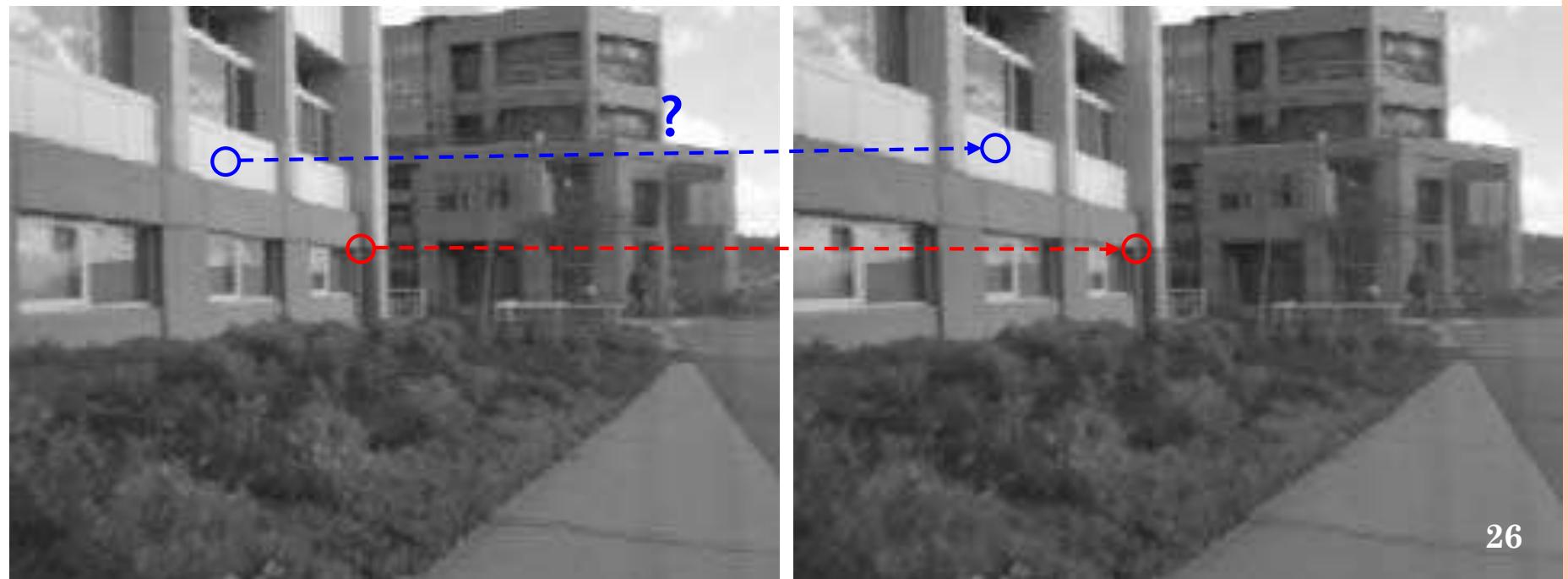


Souvent modélisée comme une transformation linéaire

[Extrait de [5]]

# POINTS D'INTÉRÊT - FEATURES [3] - RÉSUMÉ

- Points faciles à mettre en correspondance en n'utilisant que de l'information locale.



# POINTS D'INTÉRÊTS - FEATURES [3] - RÉSUMÉ

- Propriétés recherchées :
  - Distinctif : un point d'intérêt peut être apparié avec une forte probabilité.
  - Invariant : invariant aux changements d'échelle, à la rotation, aux transformations affines, à l'éclairage ...
- Applications :
  - Reconnaissance d'objets ou de scènes
  - Structure from motion
  - Stéréo
  - Suivi de mouvements (Motion tracking)
- Constitués de :
  - **Feature detection** : localisation de leur positionnement
  - **Feature description** : description de ce qu'ils sont
  - **Feature matching** : mise en correspondance selon si ce sont les mêmes ou non

# DÉTECTION DE POINTS D'INTÉRÊTS [6]

## Différentes approches

1. **Approches contours** : détecter les contours puis extraction des points d'intérêts le long des contours en considérants les points de courbures maximales ainsi que les intersections de contours.
  2. **Approches intensité** : à partir des niveaux de gris de l'image, trouver un opérateur qui est maximal aux points d'intérêt
  3. **Approches à base de modèles** : identification des points d'intérêts par mise en correspondance de la fonction d'intensité avec un modèle théorique de cette fonction des point d'intérêts considérés.
- Les approches de la deuxième catégorie sont celles utilisées généralement car (a) indépendance vis à vis de la détection de contours (b) indépendance vis à vis du type de points d'intérêts

## LISTE (NON-EXHAUSTIVE) DE DÉTECTEURS [5]

- Hessian & Harris [Beaudet '78], [Harris '88]
- Laplacian, DoG [Lindeberg '98], [Lowe '99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid '01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid '04]
- EBR and IBR [Tuytelaars & Van Gool '04]
- MSER [Matas '02]
- Salient Regions [Kadir & Brady '01]
- Others...

La plupart de ces détecteurs sont devenus des briques de base pour des applications récentes en vision par ordinateur

# LISTE (NON-EXHAUSTIVE) DE DÉTECTEURS

- FAST [ST02]
- SIFT [Low04]
- SURF [BETV08]
- ORB [RRKB11]
- KAZE [ABD12]
- CARD [ABD12]
- BRIEF [CLÖ<sup>+</sup>12]
- BRISK [LCS11]
- FREAK [AOV12]
- LDB [XK12]
- ...

La plupart de ces détecteurs sont devenus des briques de base pour des applications récentes en vision par ordinateur

# PLAN

- Introduction
- Détection
  - de points
  - de contours
  - de régions
  - d'objets

## DÉTECTION DE POINTS [5]

- Buts :

- Détection répétable
- Localisation précise
- Contenu pertinent

Travail sur des changements bidimensionnel du signal



# Repeatability



[Extrait de [4]]

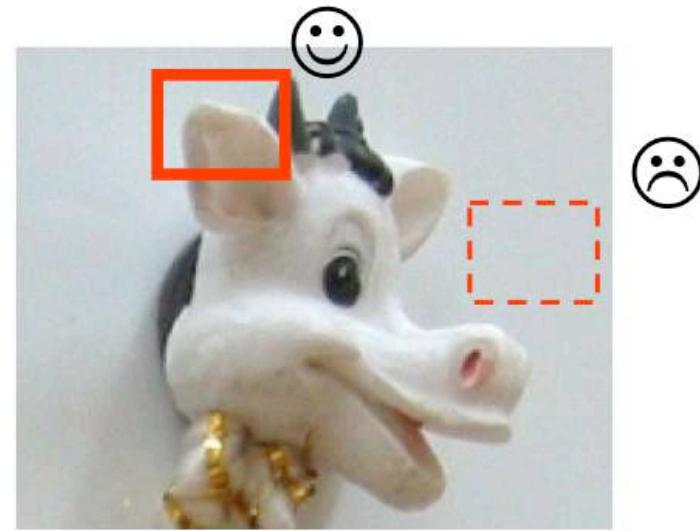


Illumination  
invariance

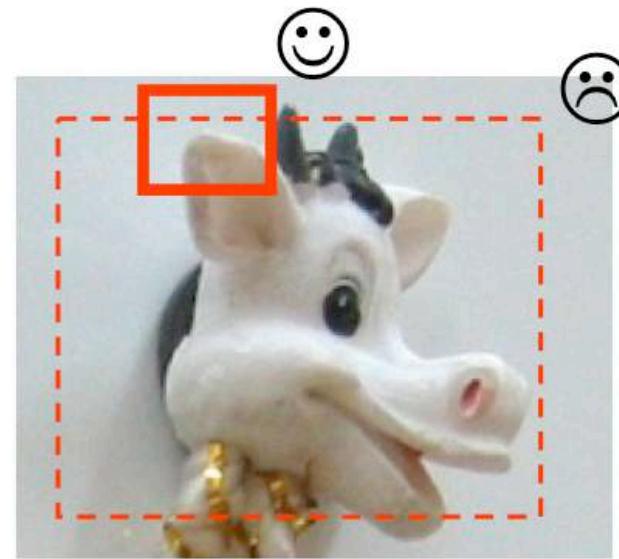
Scale  
invariance

Pose invariance  
•Rotation  
•Affine

- Saliency



- Locality



[Extrait de [4]]

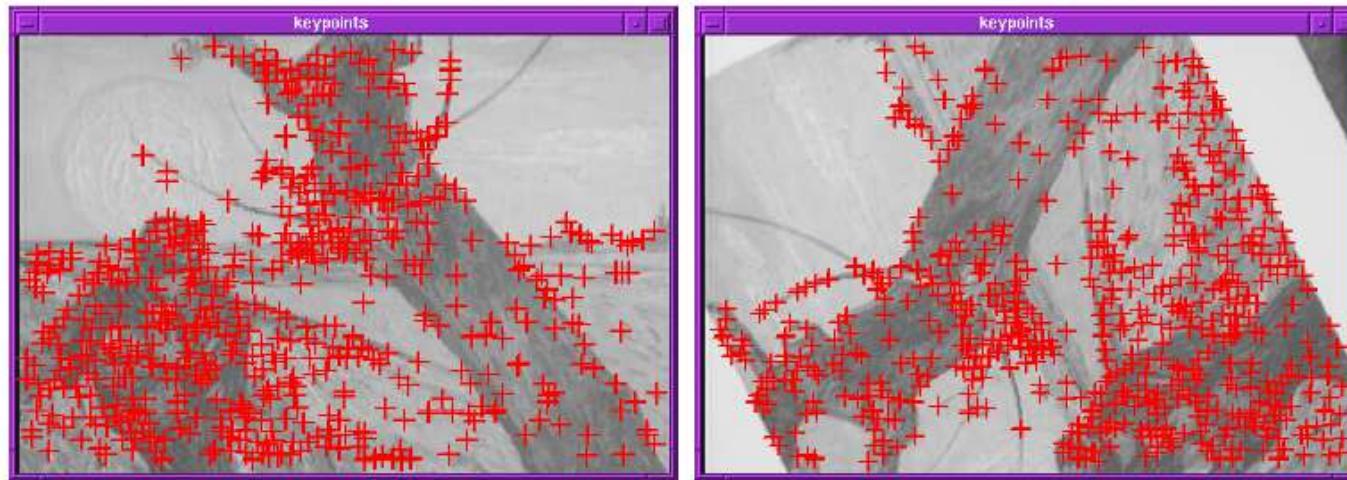
# DÉTECTION DE POINTS

- Détection de coins
  - Intersection de deux contours
  - Algorithmes
    - Moravec (1977)
    - Harris/Plessey (1988)

# DÉTECTION DE POINTS [5]

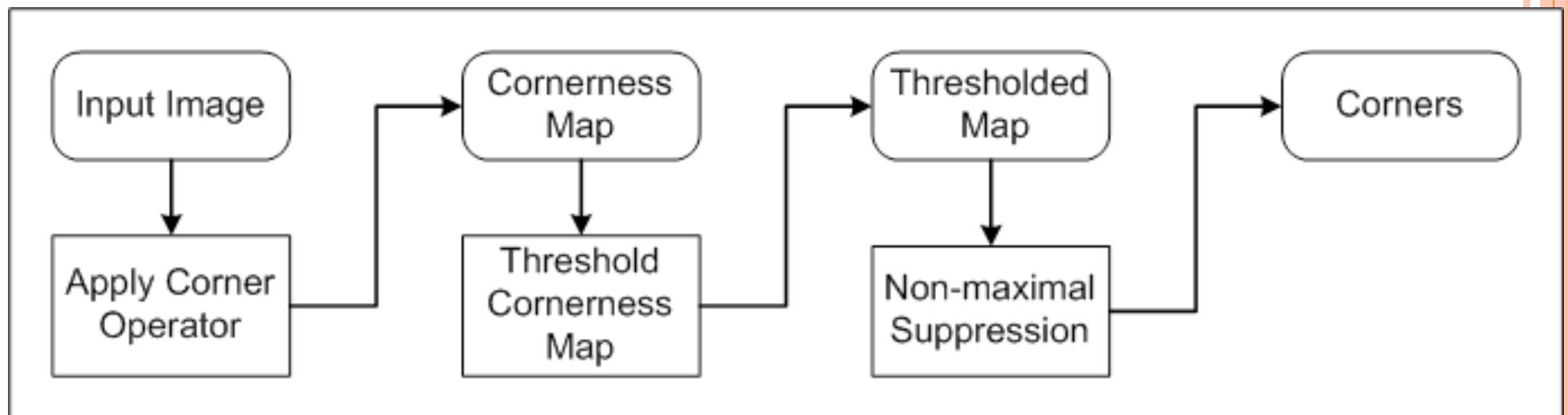
## ○ Détection de coins

- Dans les régions autour des coins, le gradient de l'image a 2 ou plus directions prépondérantes
- Les coins sont des éléments caractéristiques répétables et distinctifs



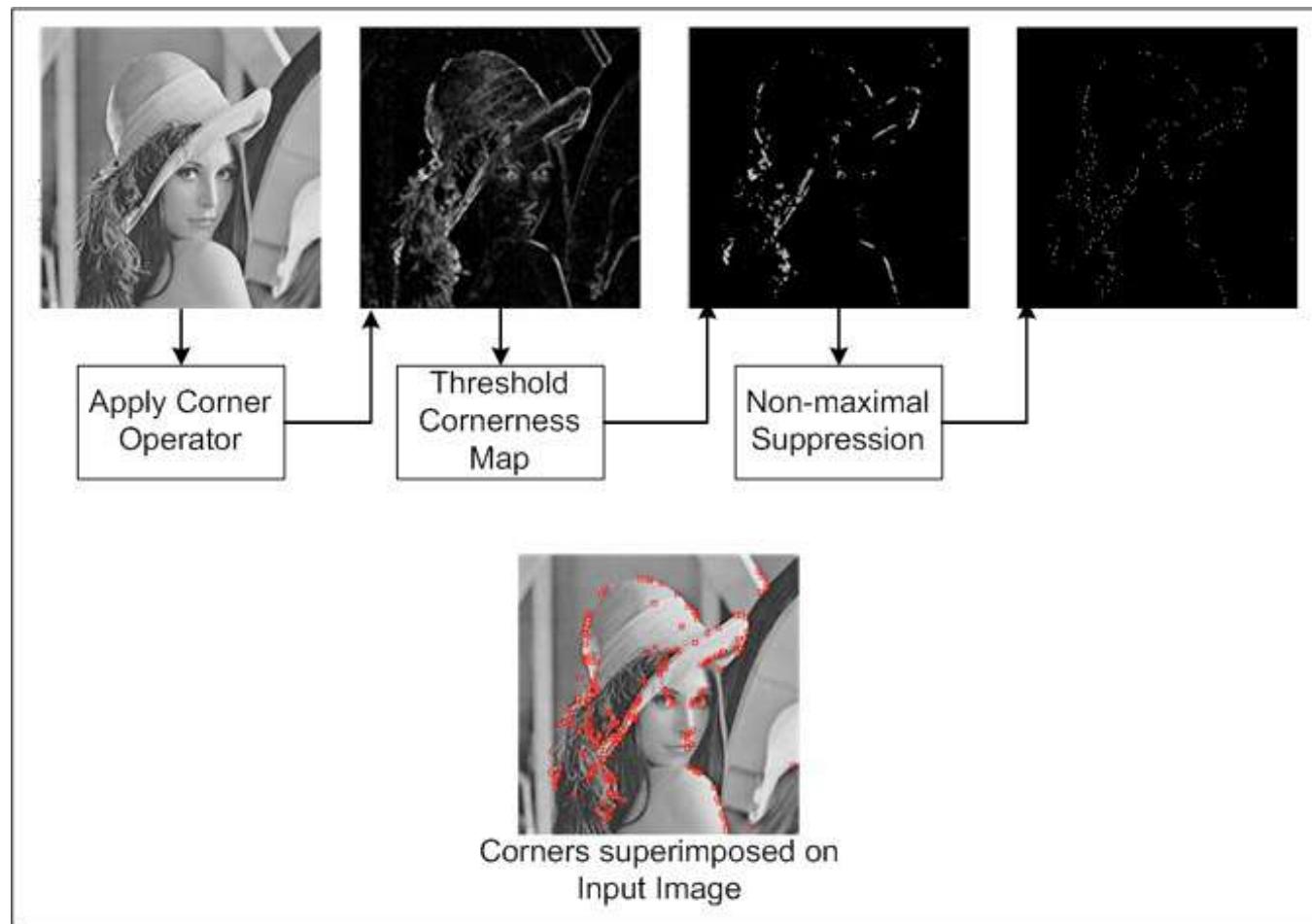
# DÉTECTION DE POINTS

- Détection de coins
  - Principe de ces algorithmes [1]



# DÉTECTION DE POINTS

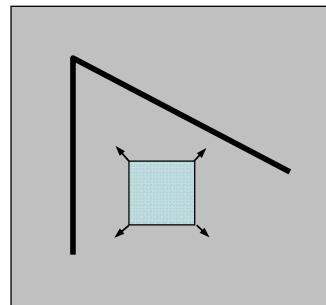
- Détection de coins [1]



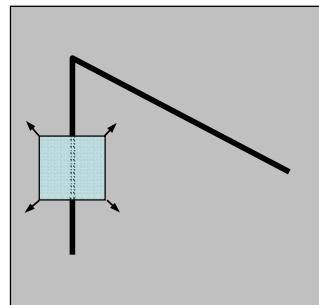
# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Moravec (1980)

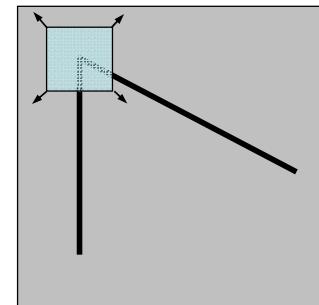
- ▶ Observer les changements d'intensité à l'intérieur d'une fenêtre



“flat” region:  
no change in  
all directions



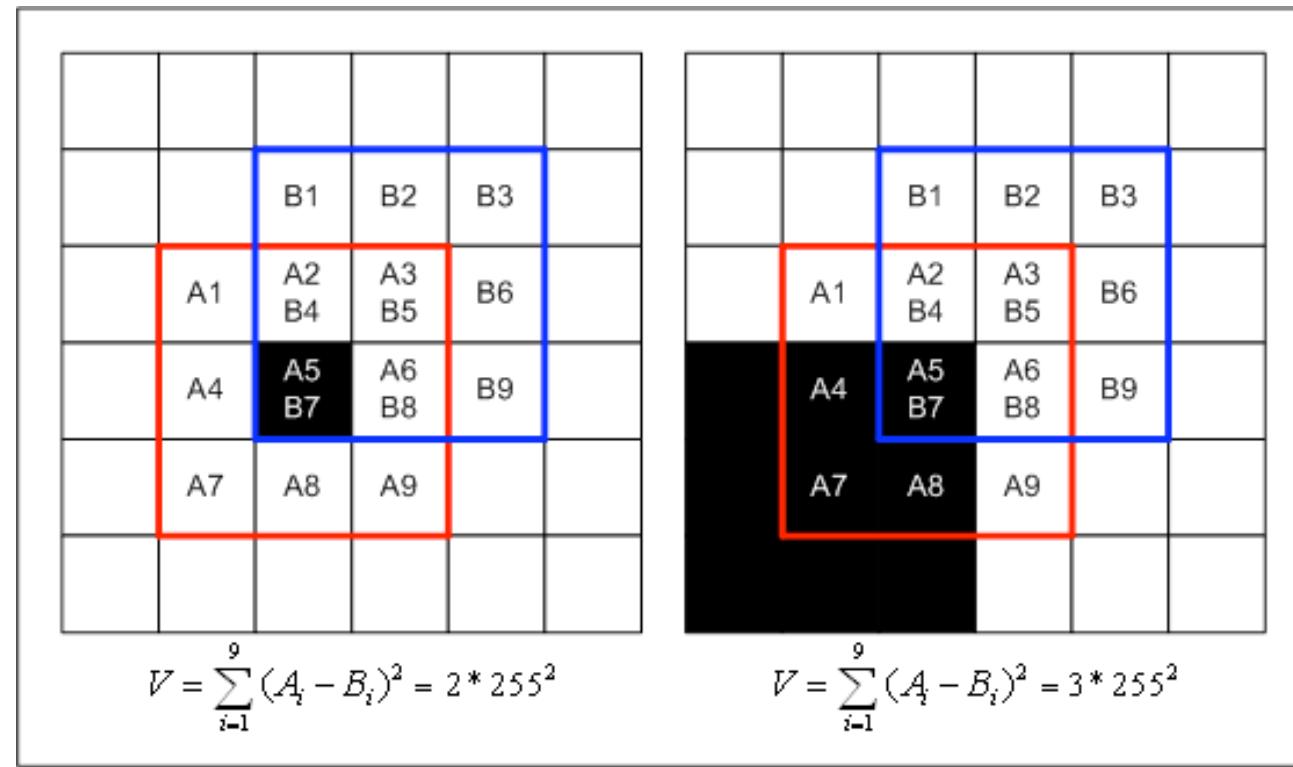
“edge”:  
no change along  
the edge direction



“corner”:  
significant change  
in all directions

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Moravec [1]



40

Calcul de la variation d'intensité avec une fenêtre 3x3 dans la direction diagonale

## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Moravec [1]

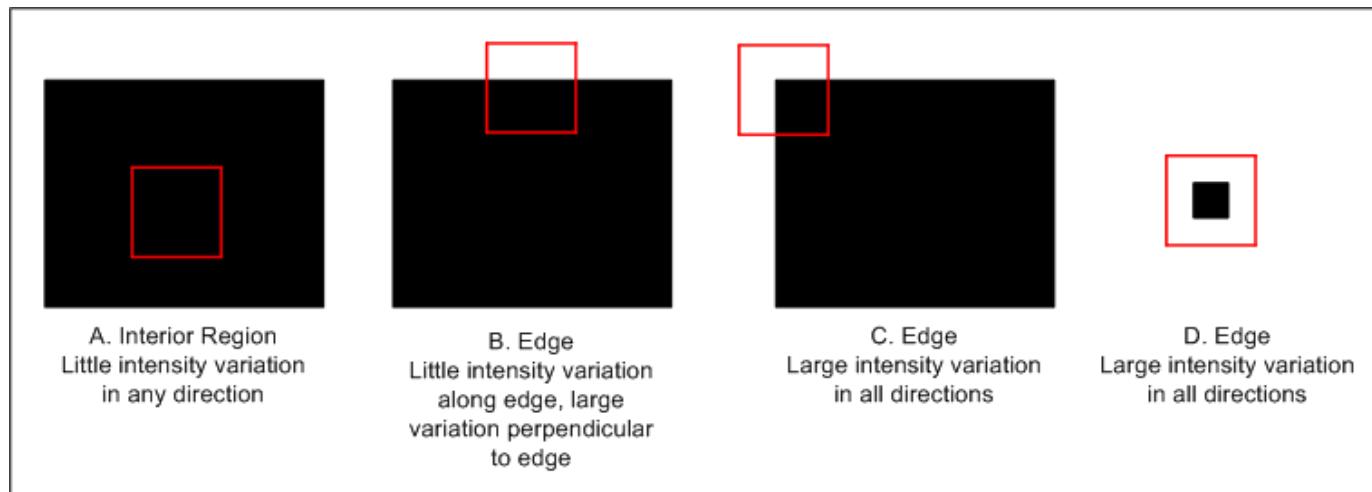
Variation moyenne de l'intensité pour un petit déplacement  $(u, v)$

$$E(u, v) = \sum_{x,y} w(x, y) |I(x + u, y + v) - I(x, y)|^2$$

- ▶  $w$  spécifie le voisinage considérée (valeur 1 à l'intérieur de la fenêtre et 0 à l'extérieur) ;
- ▶  $I(x, y)$  est l'intensité au pixel  $(x, y)$

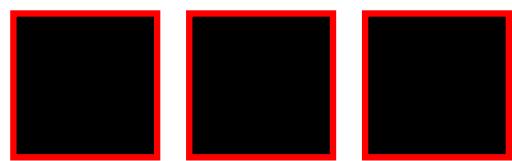
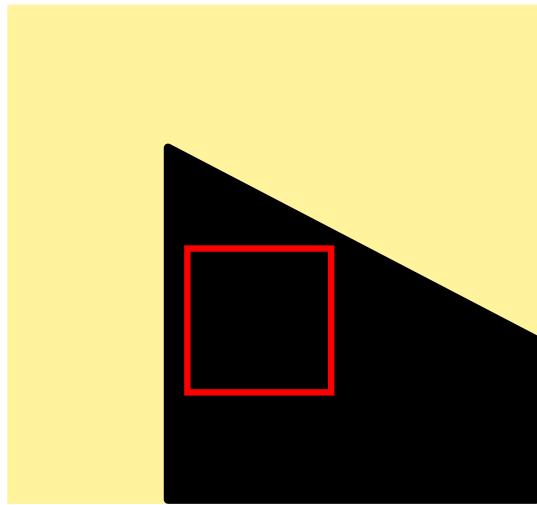
# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Moravec [1]



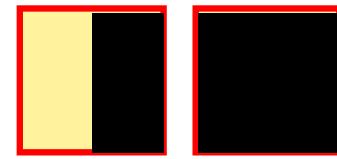
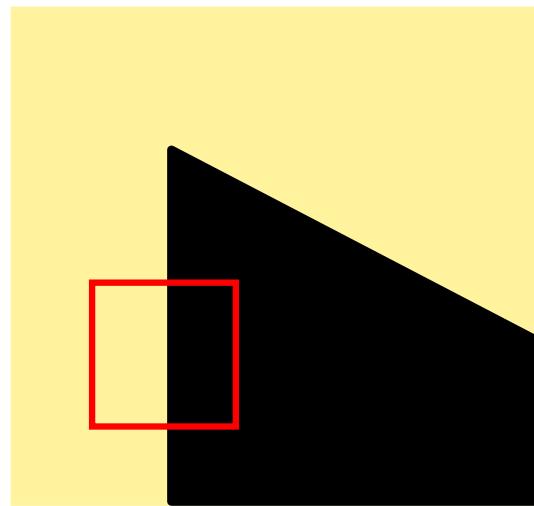
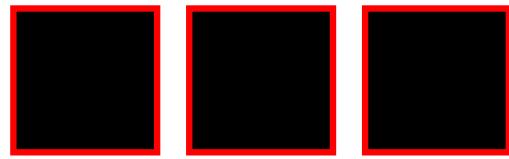
- A. intensité presque constante :  $E(u, v) \approx 0$
- B. contour :  $E(u, v) \approx 0$  pour des déplacements le long du contour ( $v \neq 0$ ) ;  $E(u, v) > 0$  pour des déplacements perpendiculaires
- C. coin :  $E(u, v) > 0$  pour tout  $(u, v) \neq (0, 0)$
- D. pixel seul : idem coin

# MORAVEC CORNER DETECTOR [3]



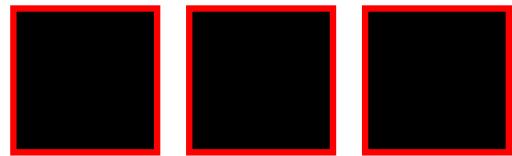
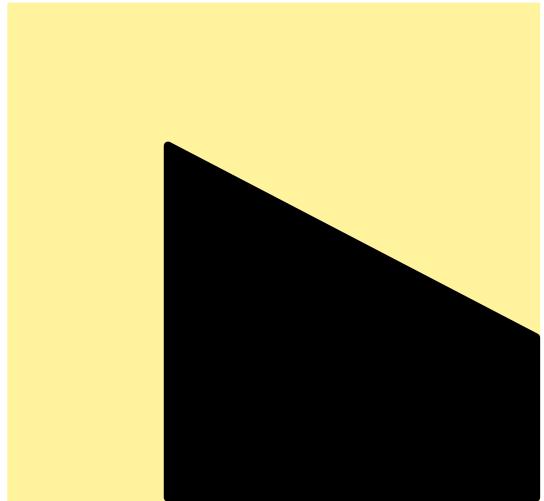
flat

# MORAVEC CORNER DETECTOR [3]

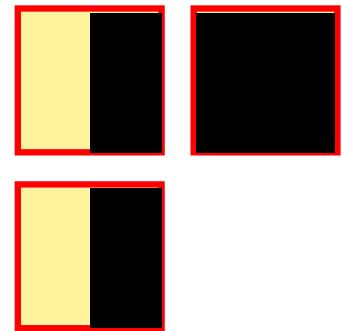
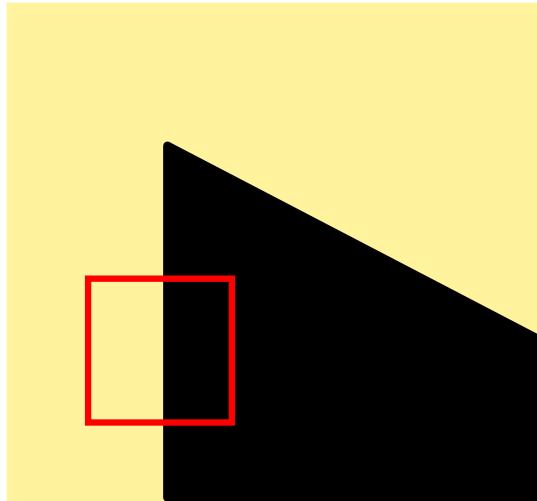


flat

## MORAVEC CORNER DETECTOR [3]

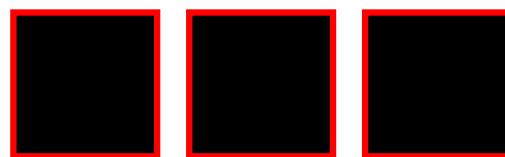
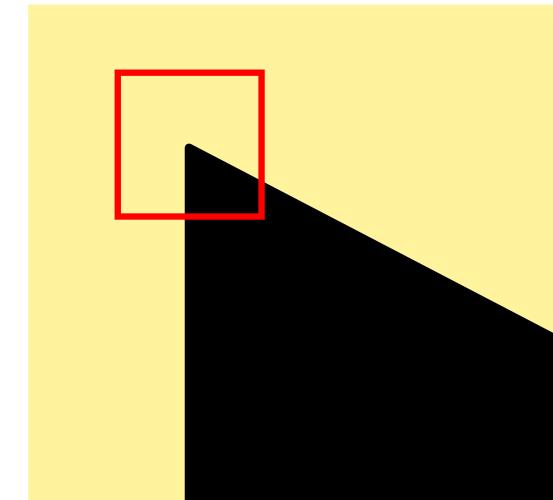


flat

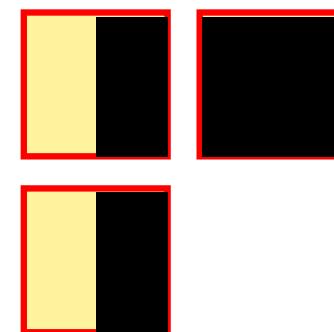


edge

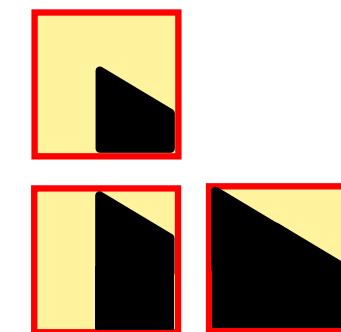
## MORAVEC CORNER DETECTOR [3]



flat



edge



corner  
isolated point <sup>46</sup>

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Moravec – carte de coins [1]

x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	0	0	0	0	0	0	0	0	0	0	1	1	1	x	x
x	x	0	0	0	0	0	1	1	0	0	1	2	1	x	x	
x	x	0	0	0	0	0	2	1	0	0	1	1	1	x	x	
x	x	0	0	0	0	0	0	0	0	0	0	0	0	x	x	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Exemple de carte de coins

# DÉTECTION DE POINTS

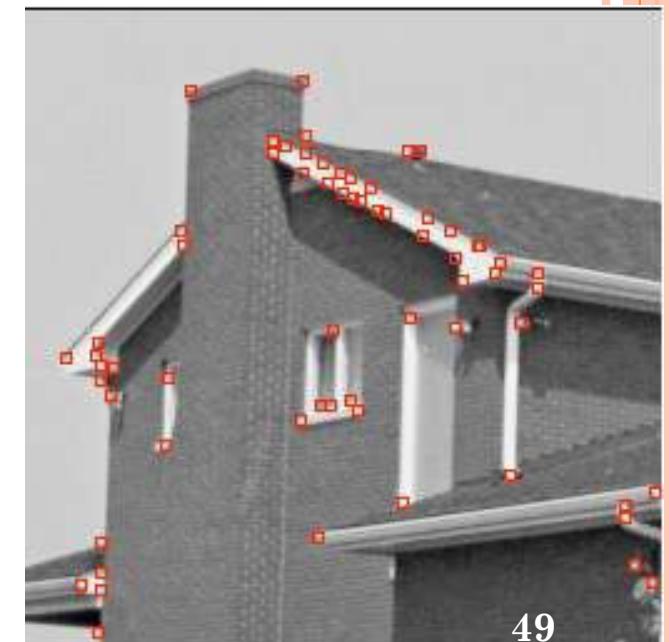
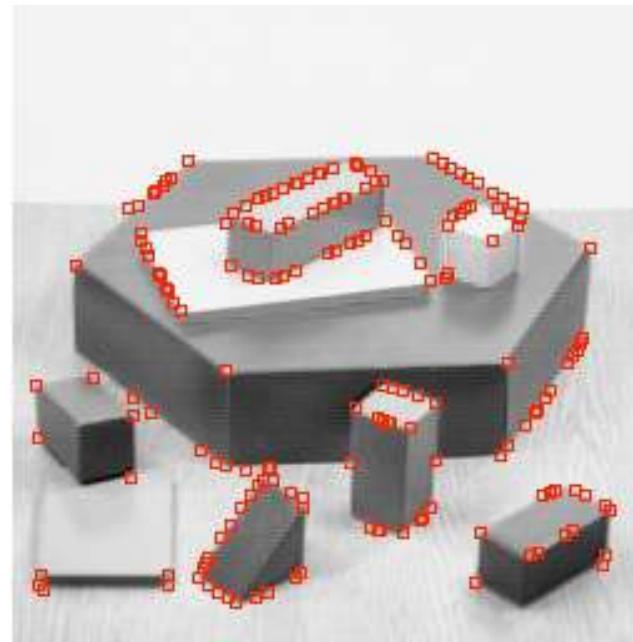
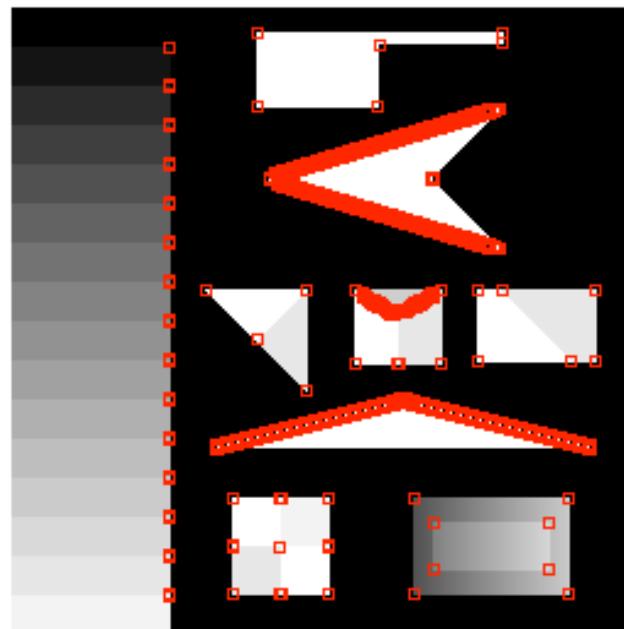
- Détection de coins – Opérateur de Moravec – Algorithme [6]

- ⇒ un coin est un maximum local de  $E$
- ⇒ pbl : la valeur de  $E$  est la même pour un coin que pour un pixel isolé

1. pour chaque pixel  $(x, y)$ , calculer les variations d'intensité  $E(u, v)$  pour  $(u, v) = \{(1, 0), (1, 1), (0, 1), (-1, 1), (-1, 0), (-1, -1), (0, -1), (1, -1)\}$
2. Construire la carte de “coinité” en calculant la mesure  $C(x, y)$  pour chaque pixel  $(x, y)$ :  $C(x, y) = \min E(u, v)$
3. Trouver les maxima de cette carte (correspondent aux points d'intérêt)

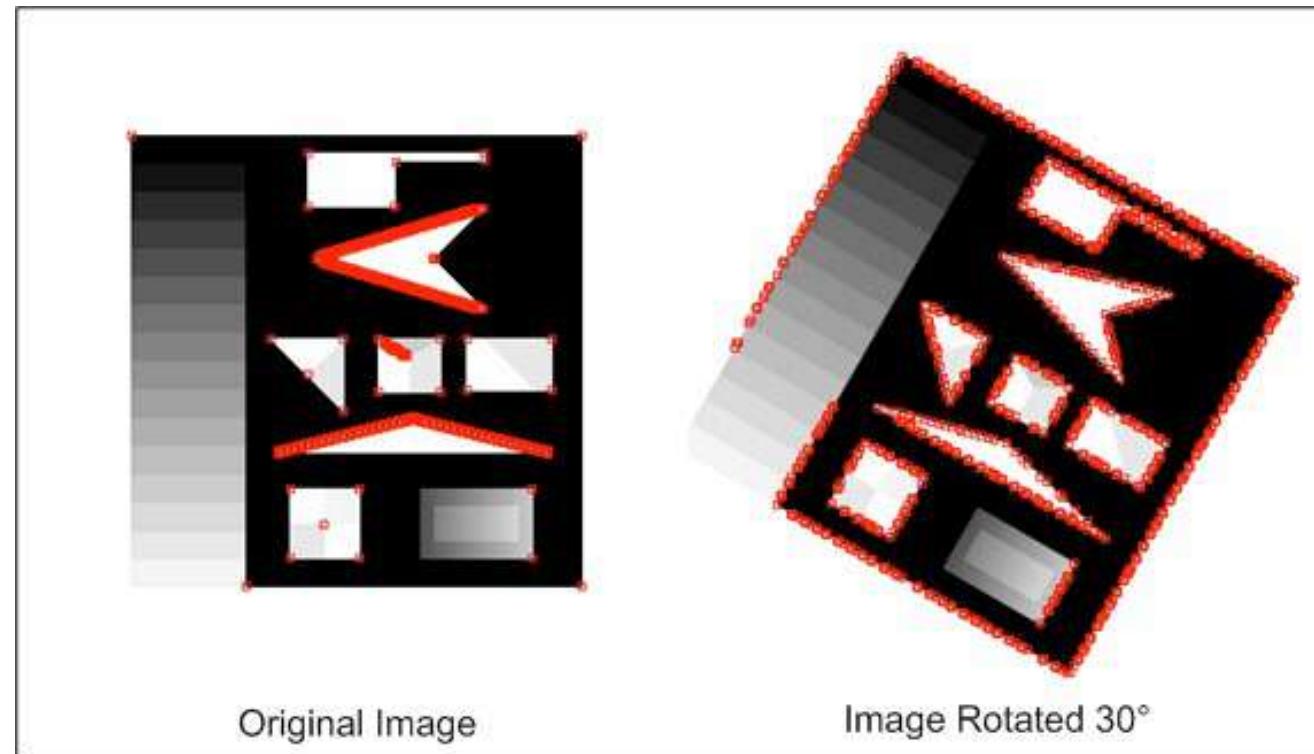
# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Moravec –  
Résultats [1]



## DÉTECTION DE POINTS

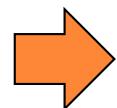
- Détection de coins – Opérateur de Moravec – Résultats [1]



Non invariance par rotation

## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Moravec
  - Algorithme simple
  - Instabilité (diagonales)
  - Non invariance par rotation



Utilisation d'autres algorithmes

## DÉTECTION DE POINTS

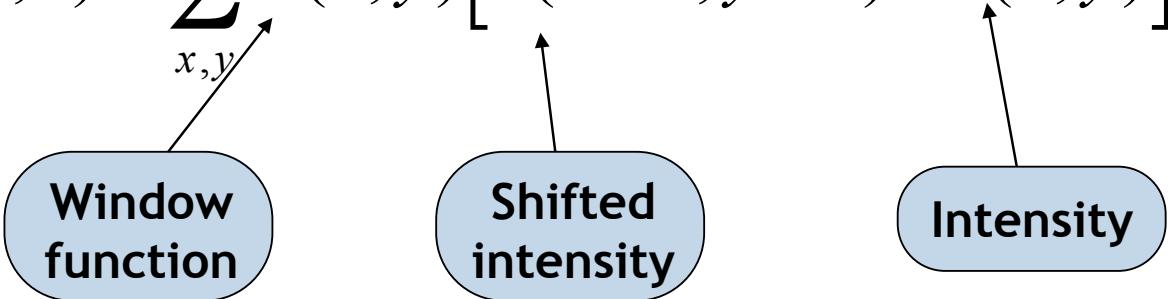
- Détection de coins –Opérateur de Harris / Plessey
  - Différence avec l'opérateur de Moravec : méthode de mesure locale de l'autocorélation
  - Suit les directions principales par utilisation de la dérivation

C.Harris and M.Stephens. "[A Combined Corner and Edge Detector.](#)"  
*Proceedings of the 4th Alvey Vision Conference*, 1988.

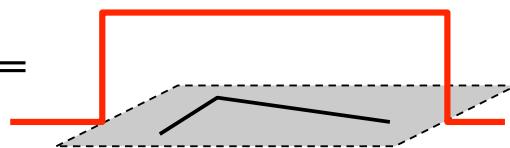
# DÉTECTION DE POINTS

- Détection de coins –Opérateur de Harris / Plessey
  - Basé sur la variation moyenne d'intensité pour un petit déplacement ( $u, v$ )

$$E(u, v) = \sum w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

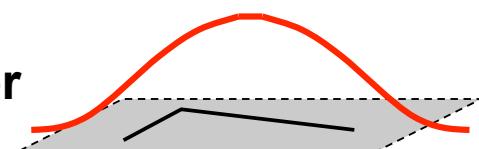


**Window function**  $w(x, y) =$



1 in window, 0 outside

or



Gaussian

[Extrait de [5]]

# DÉTECTION DE POINTS

- Détection de coins –Opérateur de Harris / Plessey (1986)

Variation moyenne de l'intensité pour un petit déplacement  $(u, v)$

$$E(x, y) = \sum_{u, v} w(u, v) |I(x + u, y + v) - I(x, y)|^2$$

Développement de Taylor :

$$I(x + u, y + v) = I(x, y) + u \frac{\partial I}{\partial x}(x, y) + v \frac{\partial I}{\partial y}(x, y) + o(u^2, v^2)$$

En négligeant  $o(u^2, v^2)$

$$E(x, y) = \sum_{u, v} w(u, v) \left\| u \frac{\partial I}{\partial x}(x, y) + v \frac{\partial I}{\partial y}(x, y) \right\|^2$$

# DÉTECTION DE POINTS

- Détection de coins –Opérateur de Harris / Plessey (1986) [7]

Variation moyenne de l'intensité pour un petit déplacement  $(u, v)$

$$\begin{aligned}
 E(x, y) &= \sum_{x,y} w(u, v) \|ul_x(x, y) + vl_y(x, y)\|^2 \\
 &= \sum_{u,v} w(u, v) \left\| \begin{bmatrix} I_x(x, y) & I_y(x, y) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right\|^2 \\
 &= [u \ v] \sum_{u,v} w(u, v) \begin{bmatrix} I_x(x, y)^2 & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y(x, y)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\
 &= [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}
 \end{aligned}$$

Somme sur la  
région d'intérêt  
où nous  
cherchons les  
coins

Gradient selon x      Gradient selon y

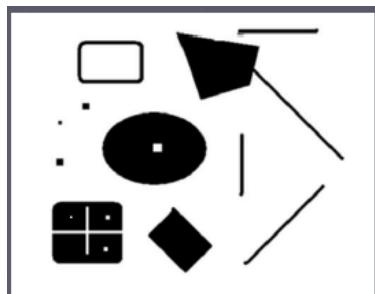
Pour de petites variations  $(u, v)$

$$E(x, y) = [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

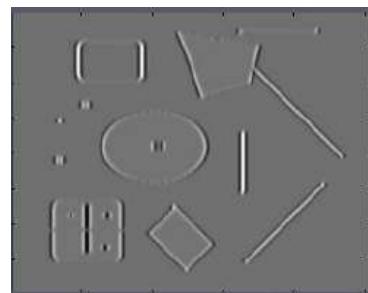
$M$  : symétrique, définie positive  $\Rightarrow$  décomposition en valeurs propres

# DÉTECTION DE POINTS

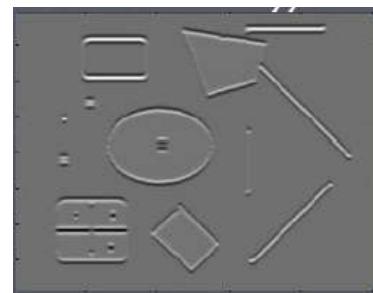
- Détection de coins –Opérateur de Harris / Plessey (1986) [Extrait de [5]]



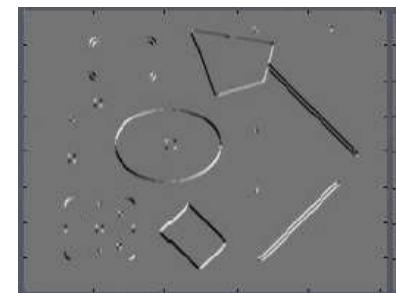
**Image  $I$**



$I_x$



$I_y$



$I_x I_y$

M est une matrice 2x2 calculée à partir des dérivées de l'image

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

↑  
Sum over image region – the area we are checking for corner

Gradient with respect to  $x$ , times gradient with respect to  $y$

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey
  - pour des petits déplacements  $(u, v)$  :

$$E(x, y) = [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

Avec

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey [6]

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Avec  $A = \frac{\delta I^2}{\delta u} \otimes w$  ;  $B = \frac{\delta I^2}{\delta v} \otimes w$  ;  $C = \left( \frac{\delta I}{\delta u} \frac{\delta I}{\delta v} \right) \otimes w$

$w$  : fenêtre gaussienne (+ isotrope)

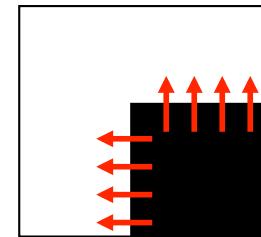
Un coin est caractérisé par une grande variation de E dans toutes les directions de (x y)

→ Calcul des valeurs propres de M

# DÉTECTION DE POINTS – DÉTECTEUR DE HARRIS [5]

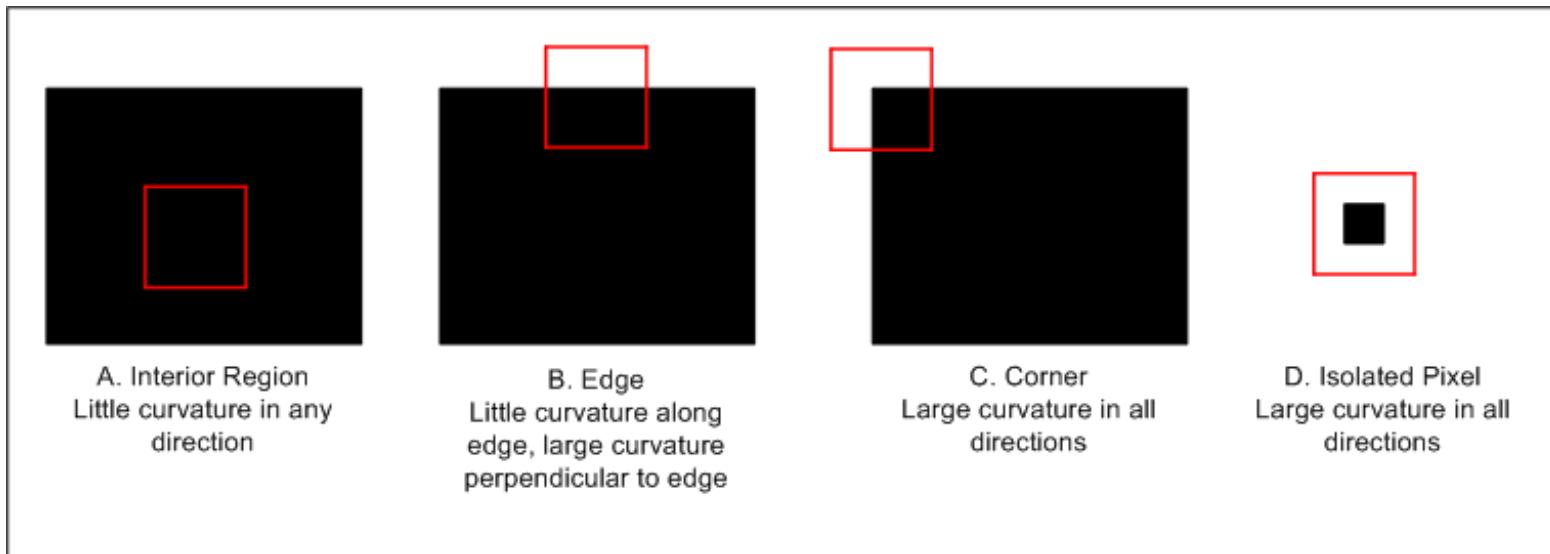
- Si nous considérons un coin aligné sur les x

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



- Cela signifie
  - Les directions dominantes des gradients seront selon x et y
  - Si aucune des valeurs propres ne sont proches de 0, alors ce n'est pas un coin, alors nous recherchons des lieux où elles sont toutes deux grandes

# DÉTECTION DE POINTS – DÉTECTEUR DE HARRIS

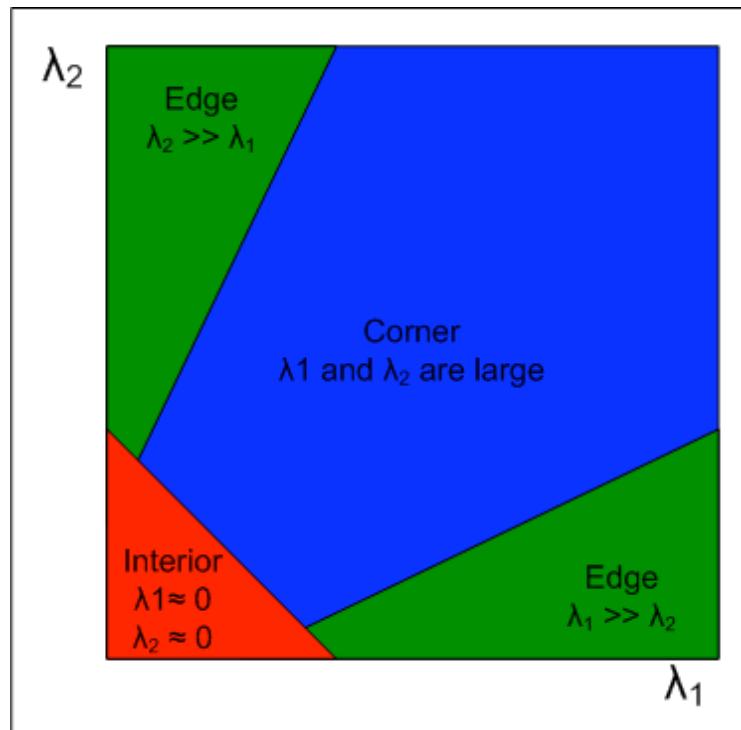


les valeurs propres de  $M$  correspondent aux courbures principales associées à  $E$  :

- A. **intensité presque constante** : les deux courbures sont de faibles valeurs
- B. **contour** : une des courbures est de forte valeur, l'autre est de faible valeur
- C. **point** : les deux courbures sont de fortes valeurs

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey



## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey

Plutôt que de calculer les valeurs propres, il est possible de calculer

$$\det(M) = AB - C^2 = \lambda_1 \cdot \lambda_2$$

$$\text{trace}(M) = A + B = \lambda_1 + \lambda_2$$

et on calcule la réponse :

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

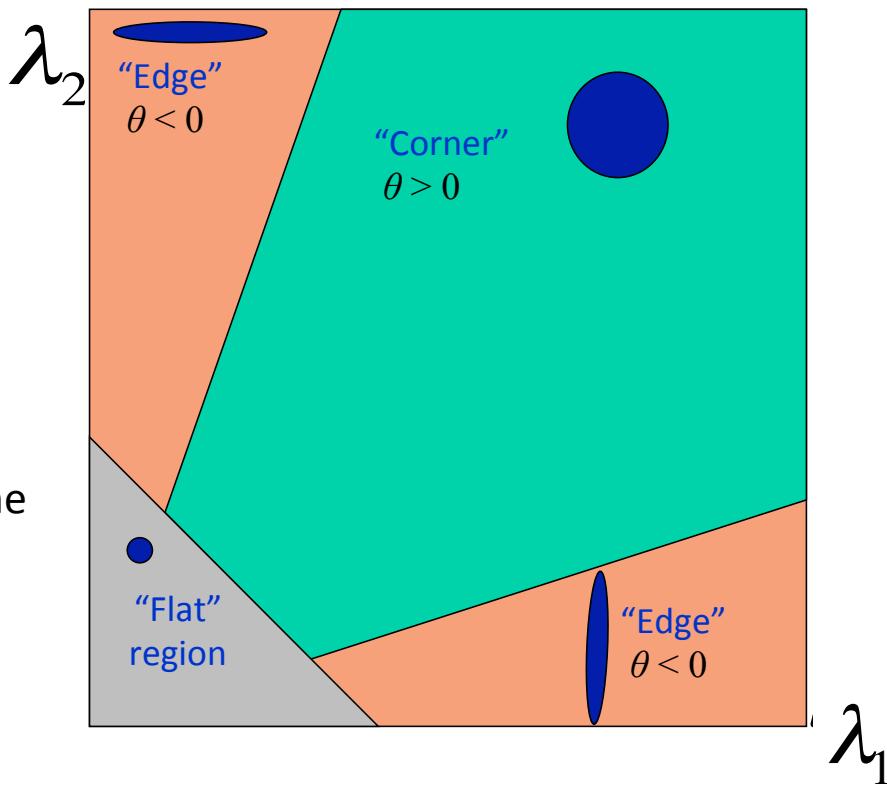
Les valeurs de  $R$  sont positives au voisinage d'un coin, négatives au voisinage d'un contour et faibles dans une région d'intensité constante ( $k = 0.04$ )

⇒ coins/point d'intérêt = max locaux de  $R$

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey

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



- Fast approximation
  - Avoid computing the eigenvalues
  - $\alpha$ : constant (0.04 to 0.06)

[Extrait de [5]]

# DÉTECTION DE POINTS

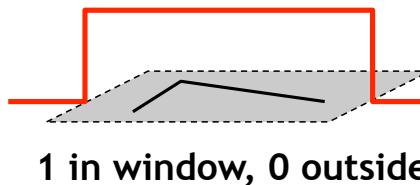
- Détection de coins – Opérateur de Harris / Plessey

## Window Function $w(x,y)$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

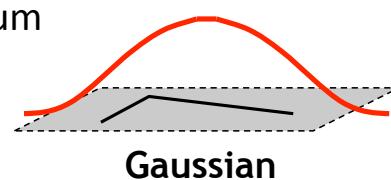
- Option 1: uniform window
  - Sum over square window
  - Problem: not rotation invariant

$$M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

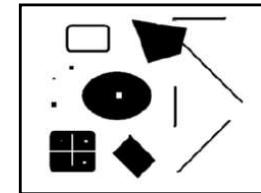


- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum
  - Result is rotation invariant

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



# SUMMARY OF HARRIS DETECTOR [3]

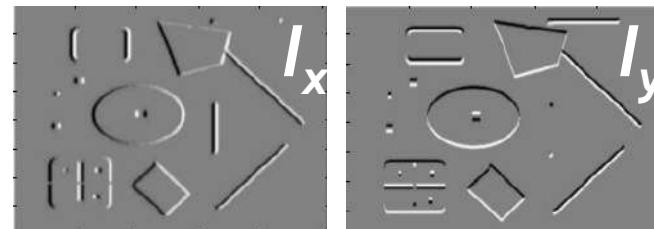


1. Calculer les dérivées x et y de l'image

$$I_x \square G_\sigma^x * I$$

$$I_y \square G_\sigma^y * I$$

Image  
derivatives



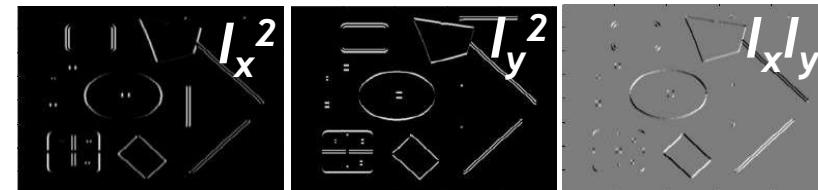
2. Calculez les produits des dérivées à chaque pixel

$$I_{x^2} \square I_x \cdot I_x$$

$$I_{y^2} \square I_y \cdot I_y$$

$$I_{xy} \square I_x \cdot I_y$$

Square of  
derivatives



## SUMMARY OF HARRIS DETECTOR [3]

3. Calculer la somme des produits des dérivées à chaque pixel

$$S_{x^2} \square G_{\sigma'} * I_{x^2} \quad S_{y^2} \square G_{\sigma'} * I_{y^2} \quad S_{xy} \square G_{\sigma'} * I_{xy}$$

Gaussian  
filter  $g(s_I)$



4. Définir la matrice à chaque pixel

$$M(x, y) \square \begin{bmatrix} S_{x^2}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{y^2}(x, y) \end{bmatrix}$$

[images de [5]]

## SUMMARY OF HARRIS DETECTOR [3]

5. Calculer la réponse du détecteur à chaque pixel
6. Seuil sur la valeur de R ; Calculer la suppression nonmax.

$$R \triangleq \det M - k \lVert \text{trace} M \rVert^2$$



[images de [5]]

## DÉTECTION DE POINTS

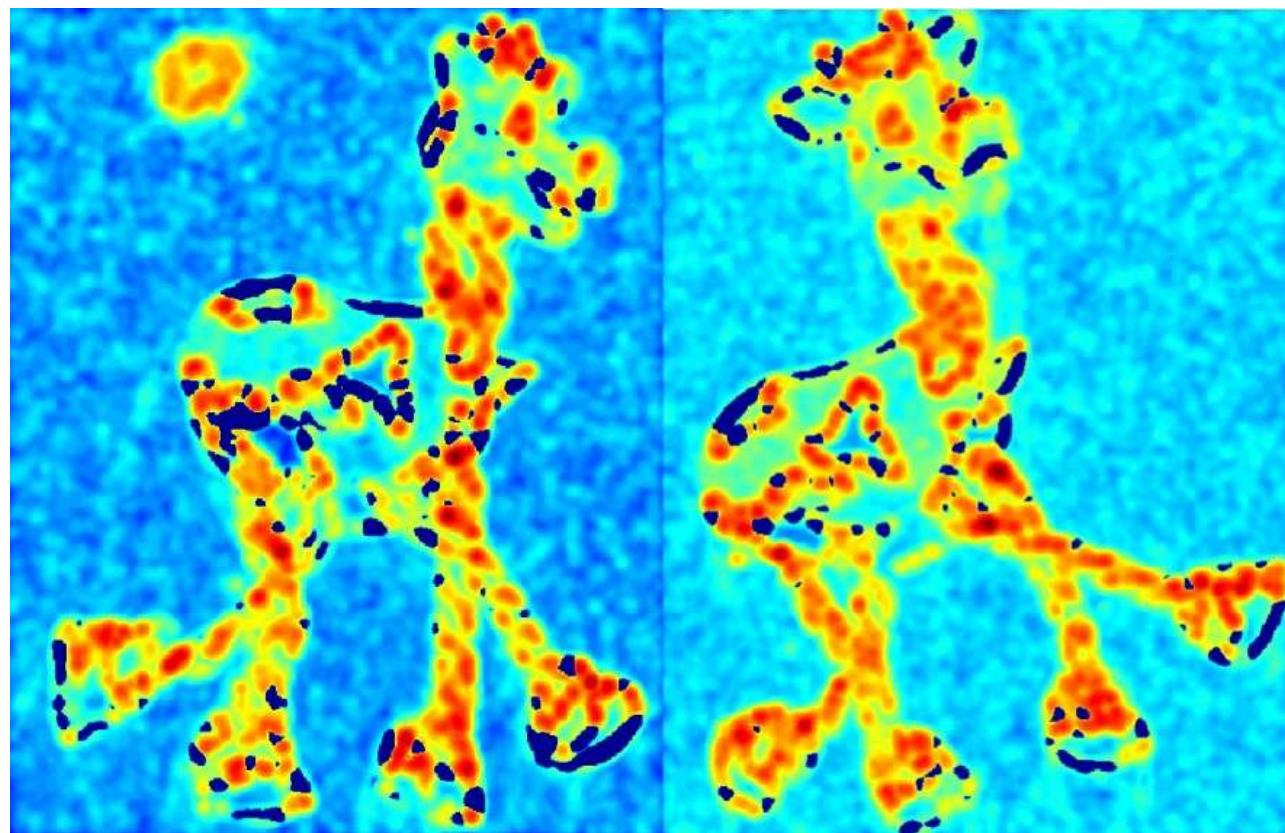
- Détection de coins – Opérateur de Harris / Plessey – Fonctionnement



Images originales

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Fonctionnement



Calcul du détecteur de Harris

## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Fonctionnement



Seuillage des pixels (coins)

## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Fonctionnement



Obtention des maxima locaux

# DÉTECTION DE POINTS

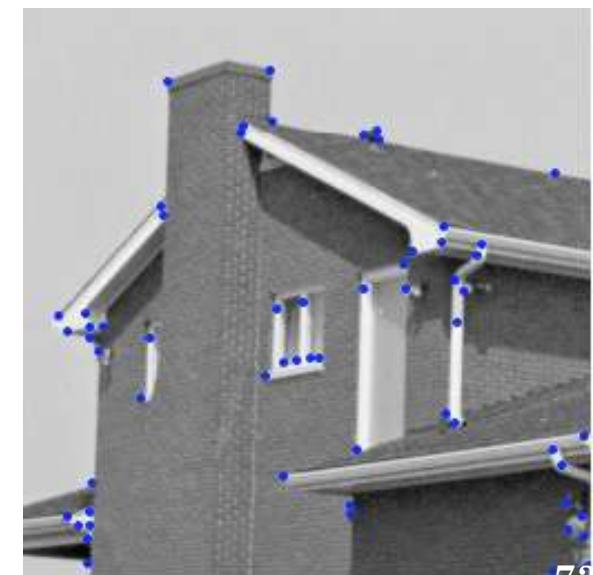
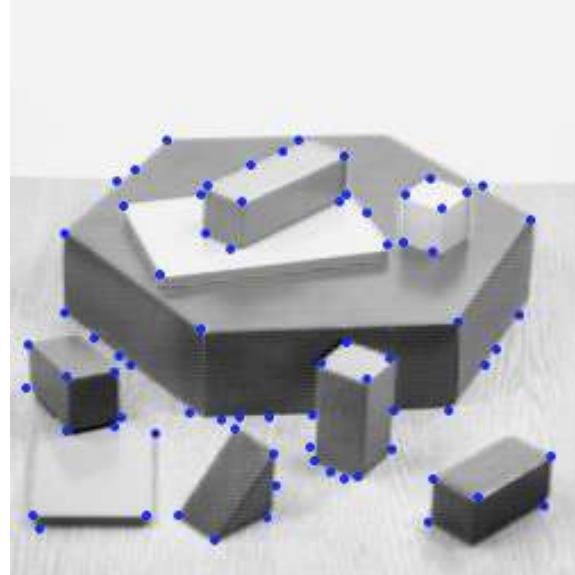
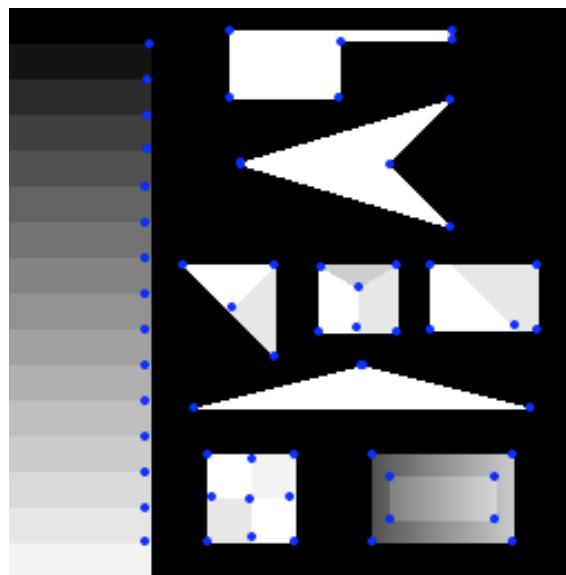
- Détection de coins – Opérateur de Harris / Plessey – Fonctionnement



Superposition sur les images originales

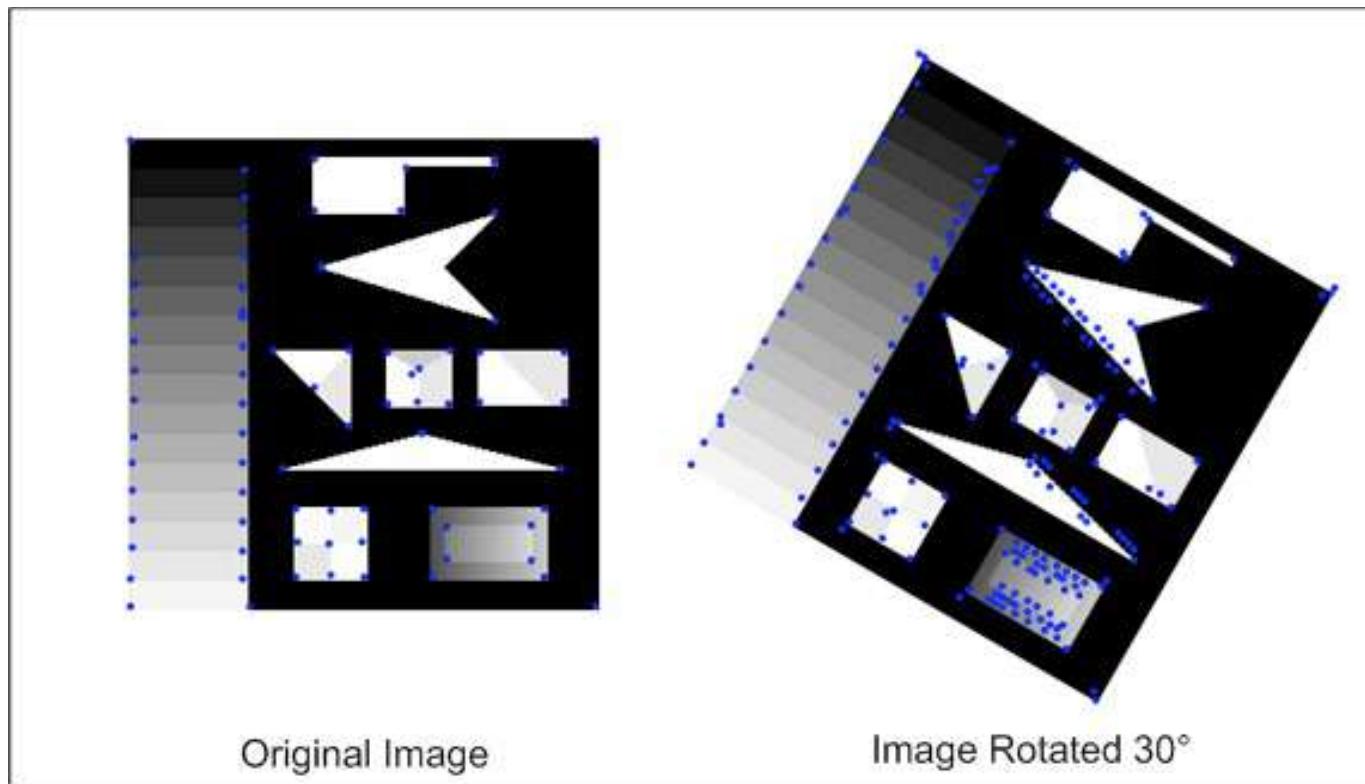
# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Résultats [1]



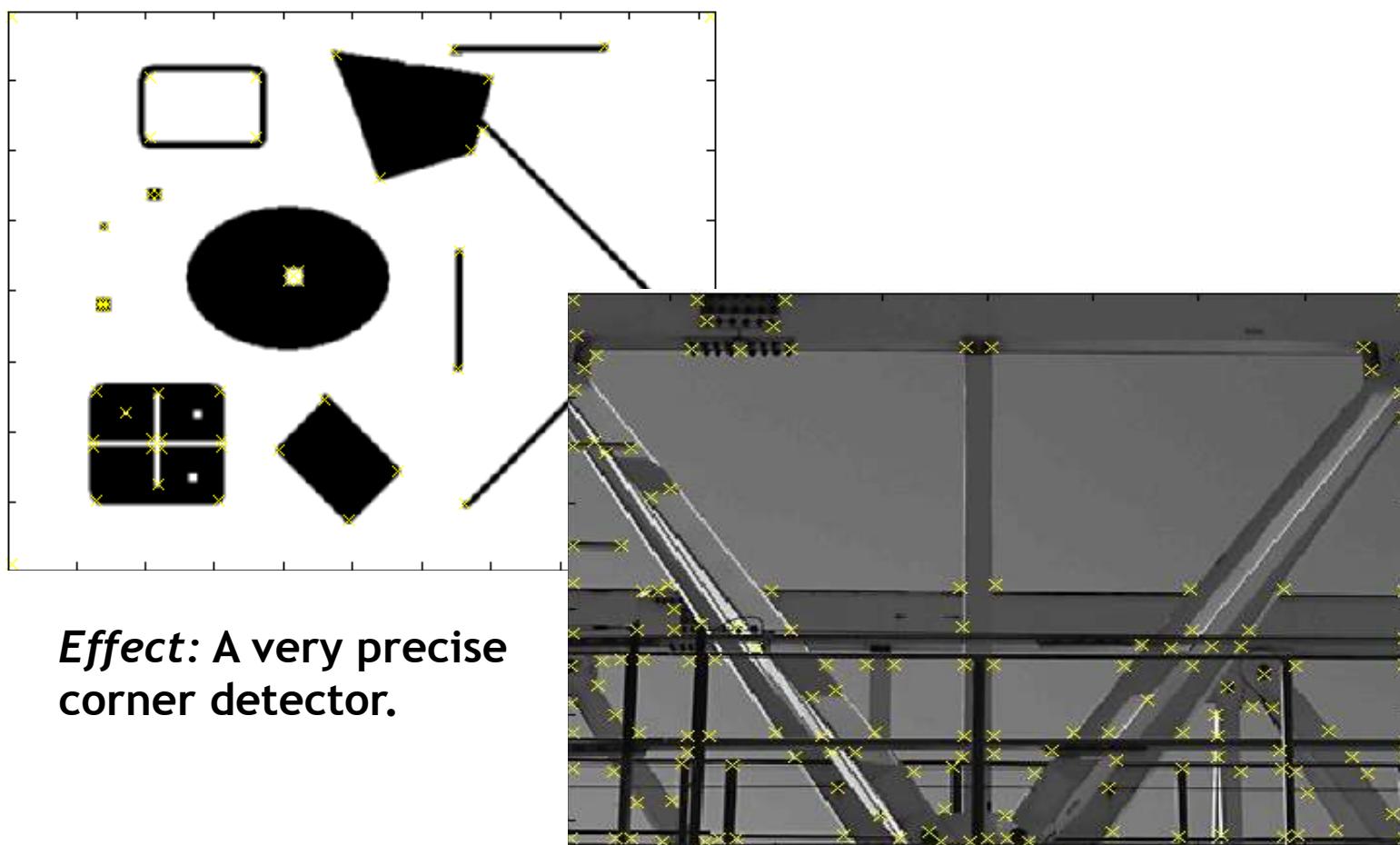
## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Résultats [1]



# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Résultats [5]



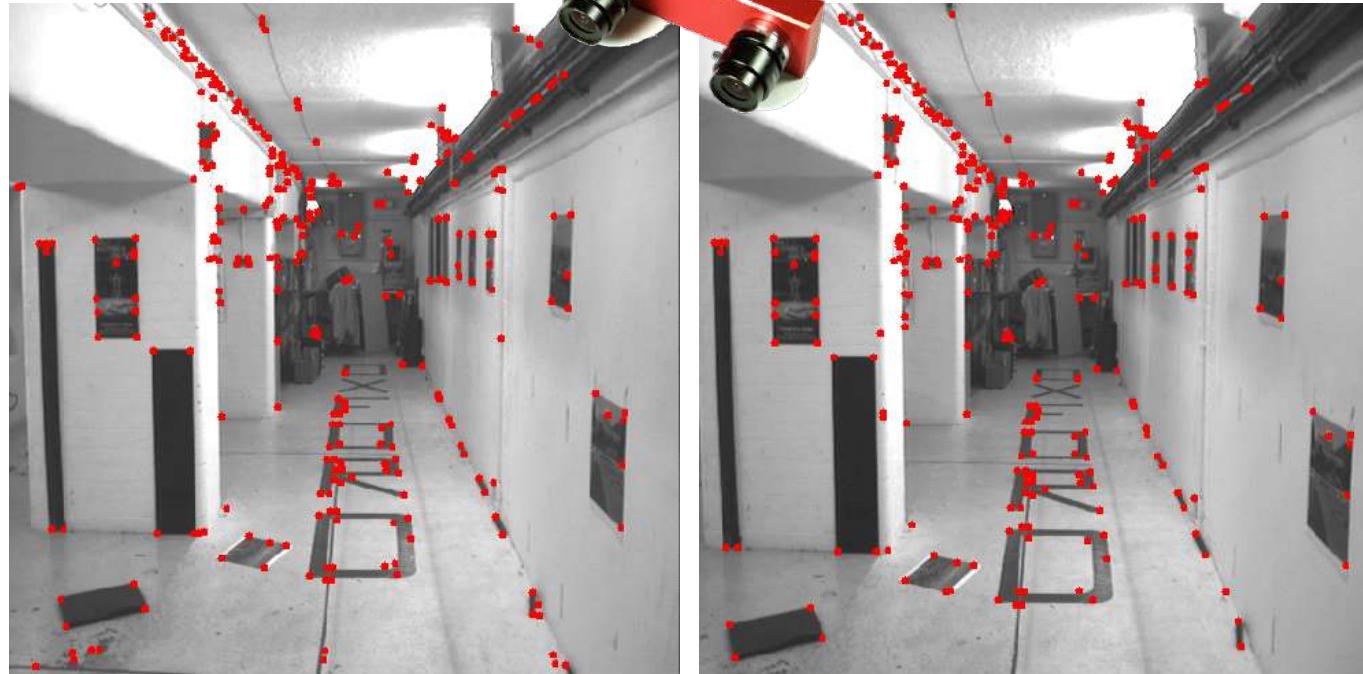
## DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Résultats [5]



# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Résultats [5]

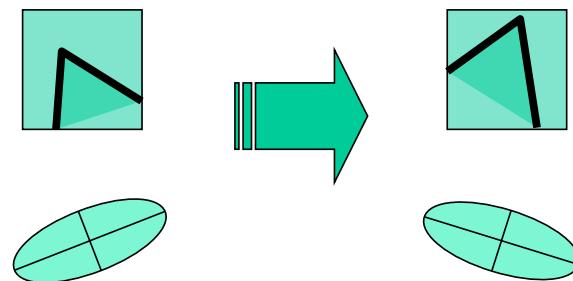


- Results are well suited for finding stereo correspondences

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey – Propriétés

- Invariant en translation
- Invariant en rotation

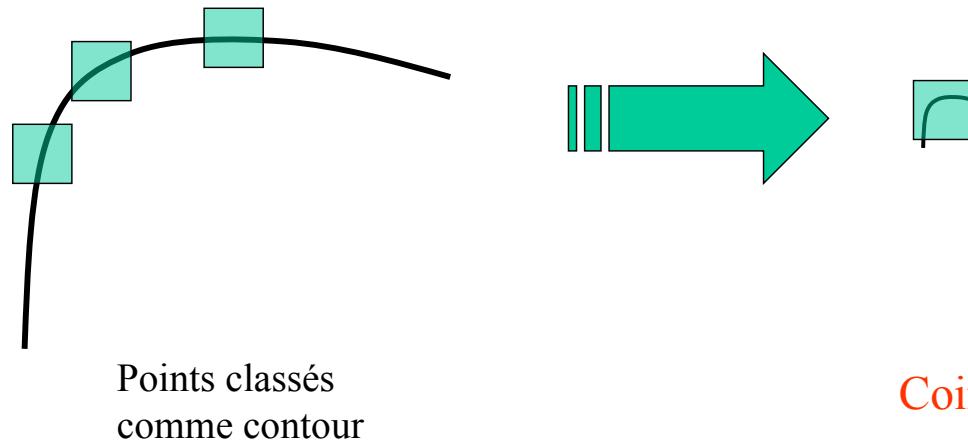


Même après rotation, la forme reste la même, et les valeurs propres de la matrice aussi

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey - Propriétés

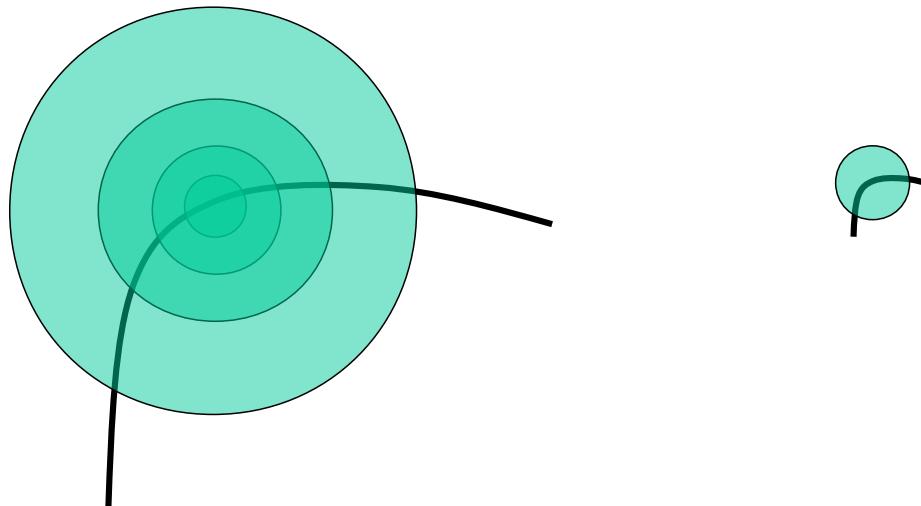
- Le détecteur de Harris n'est pas invariant à l'échelle
  - Ne fonctionne pas si changement d'échelle



Source : Frolova & Simakov, *Invariant Feature Detectors and Descriptors*, Weizmann Ins. (Israel).

# DÉTECTION DE POINTS

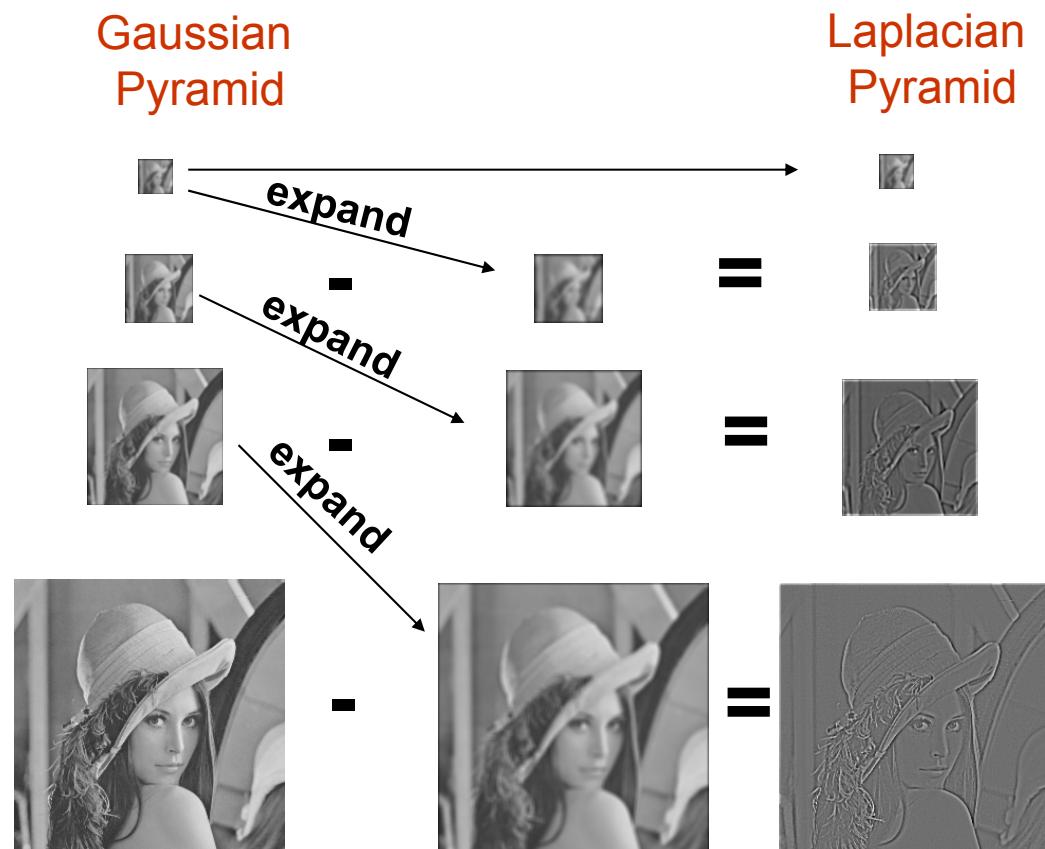
- Détection de coins – Opérateur de Harris / Plessey – DéTECTEURS INVARIANTS À L'ÉCHELLE
  - Il existe des variantes permettant d'être invariant à l'échelle
  - Exemple : Harris-Laplacien



Source : Frolova & Simakov, *Invariant Feature Detectors and Descriptors*, Weizmann Ins. (Israel).

# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey - DéTECTEURS INVARIANTS À L'ÉCHELLE

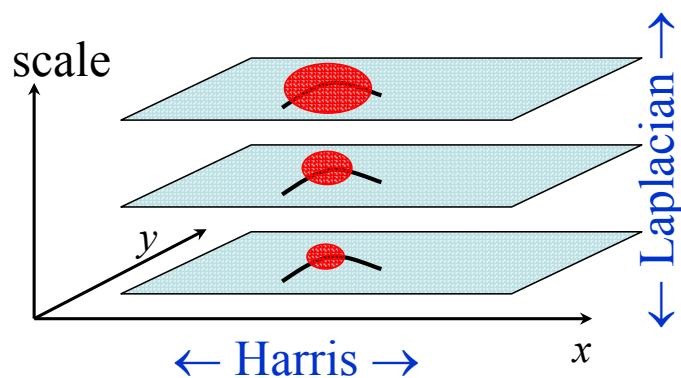


# DÉTECTION DE POINTS

- Détection de coins – Opérateur de Harris / Plessey - DéTECTeurs invariants à l'échelle

- **Harris-Laplacian**

*Find local maximum of:* Harris corner detector  
for a set of Laplacian images.



## DÉTECTION DE POINTS

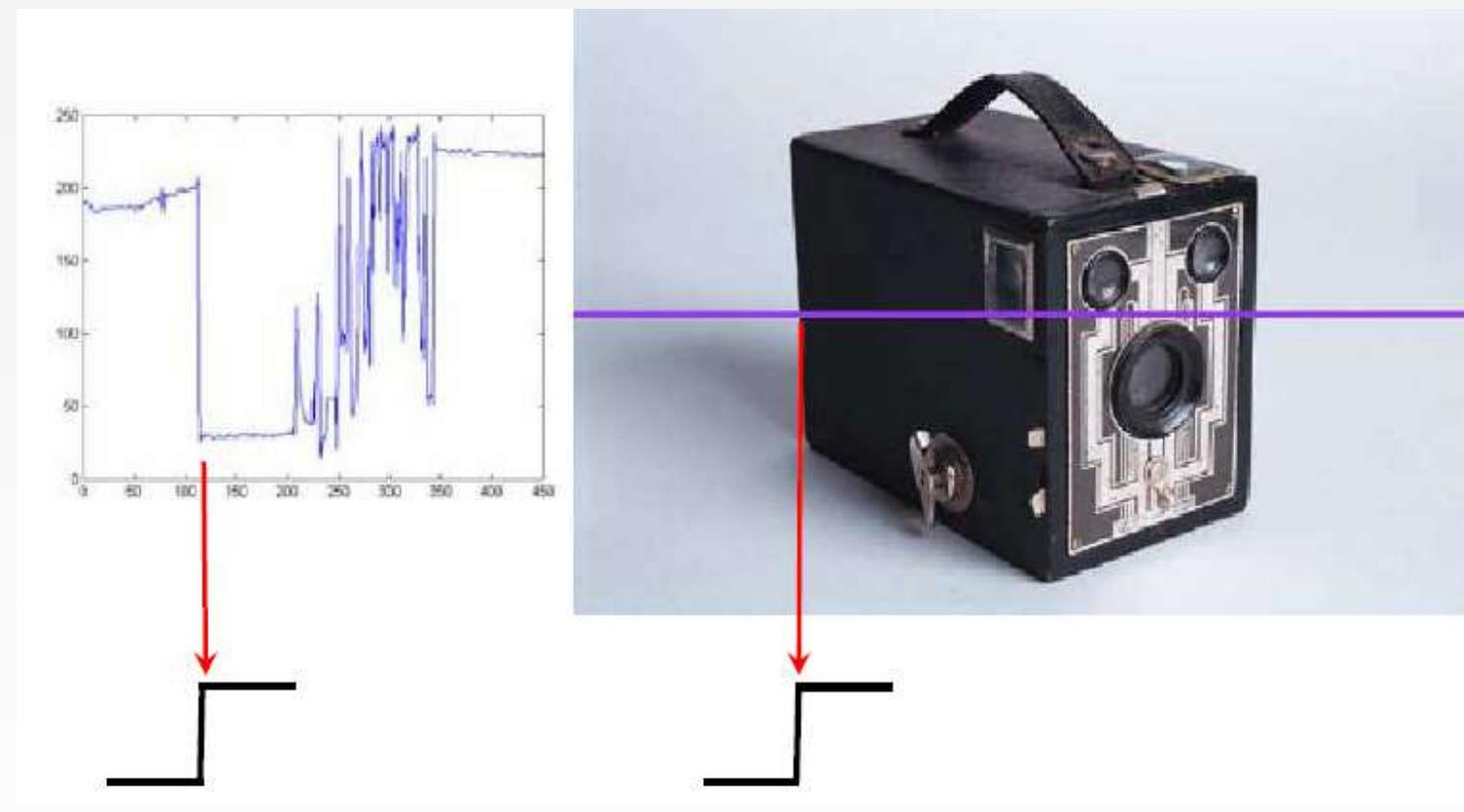
- Détection de coins – Opérateur de Harris / Plessey – Limitations [1]
  - Temps de calcul
  - Sensibilité au bruit
  - Bonne localisation seulement de liaisons en L
  - Opérateur anisotropique

# PLAN

- Introduction
- Détection
  - de points
  - de contours
  - de régions
  - d'objets

# DÉTECTION DE CONTOURS – DÉFINITION [6]

- ▶ Contour : frontière entre deux objets dans une image
- ▶ Définition plus large : discontinuité de l'image (variation brusque d'intensité).



# DÉTECTION DE CONTOURS – DISCONTINUITÉS DE L'IMAGE [6]

## ○ Origine des contours [5]



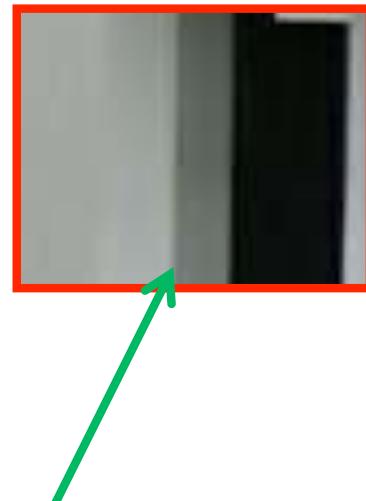
- Discontinuités entre surfaces
- Discontinuités de profondeur
- Discontinuités entre couleurs
- Discontinuités d'éclairement

# DÉTECTION DE CONTOURS – DISCONTINUITÉS DE L'IMAGE [6]

- Origine des contours [5]



- Discontinuités entre surfaces



# DÉTECTION DE CONTOURS – DISCONTINUITÉS DE L'IMAGE [6]

- Origine des contours [5]



- Discontinuités de profondeur



# DÉTECTION DE CONTOURS – DISCONTINUITÉS DE L'IMAGE [6]

- Origine des contours [5]



- Discontinuités entre couleurs



## DÉTECTION DE CONTOURS [7]

étape préliminaire à de nombreuses applications de l'analyse d'images

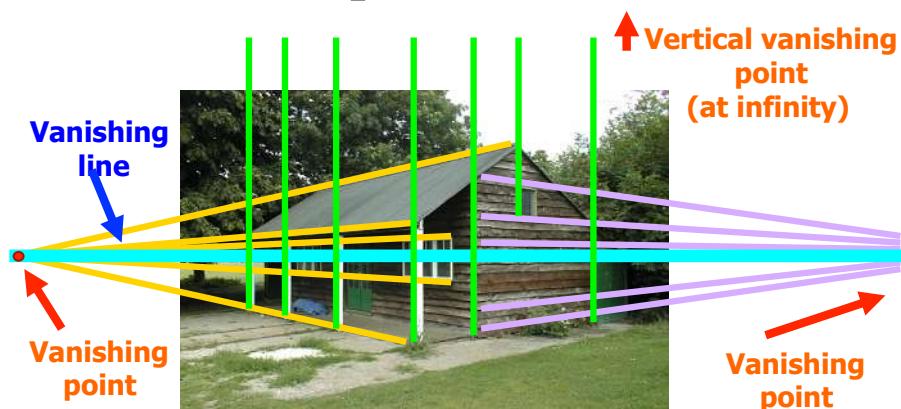
les contours constituent des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image

Les contours dans une image proviennent des :

- ▶ discontinuités de la fonction de réflectance
- ▶ discontinuités de profondeur

# DÉTECTION DE CONTOURS

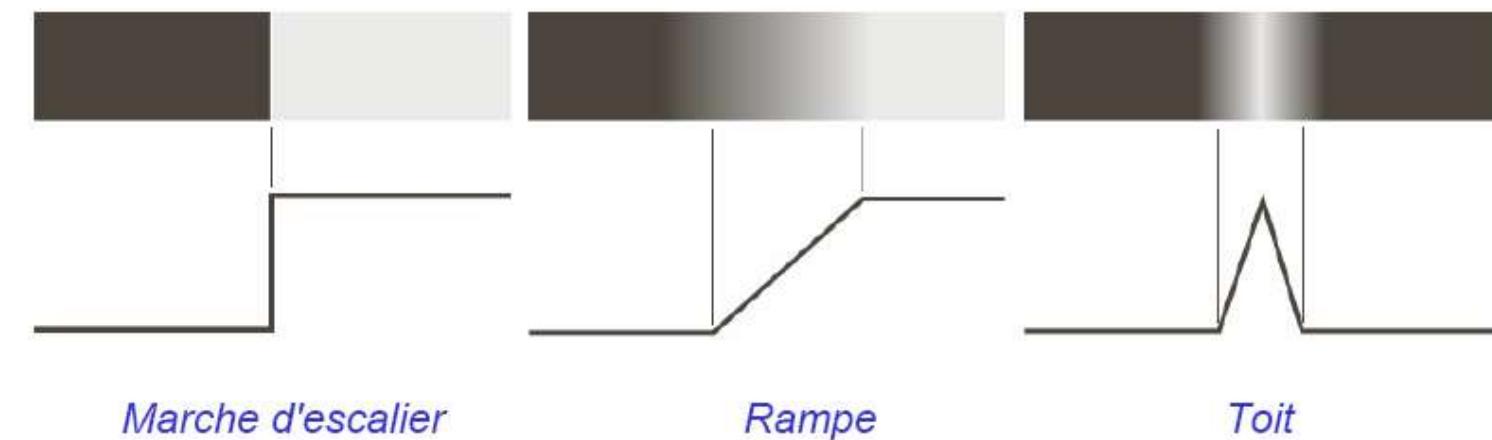
- Utilisation des contours pour :
  - reconnaître des objets présents dans une scène,
  - différencier des zones de l'image,
  - faire de la segmentation d'images,
  - extraire une information réduite souvent pertinente pour caractériser l'image
  - Reconstruire la géométrie et le point de vue



Source: J. Hayes

# DÉTECTION DE CONTOURS

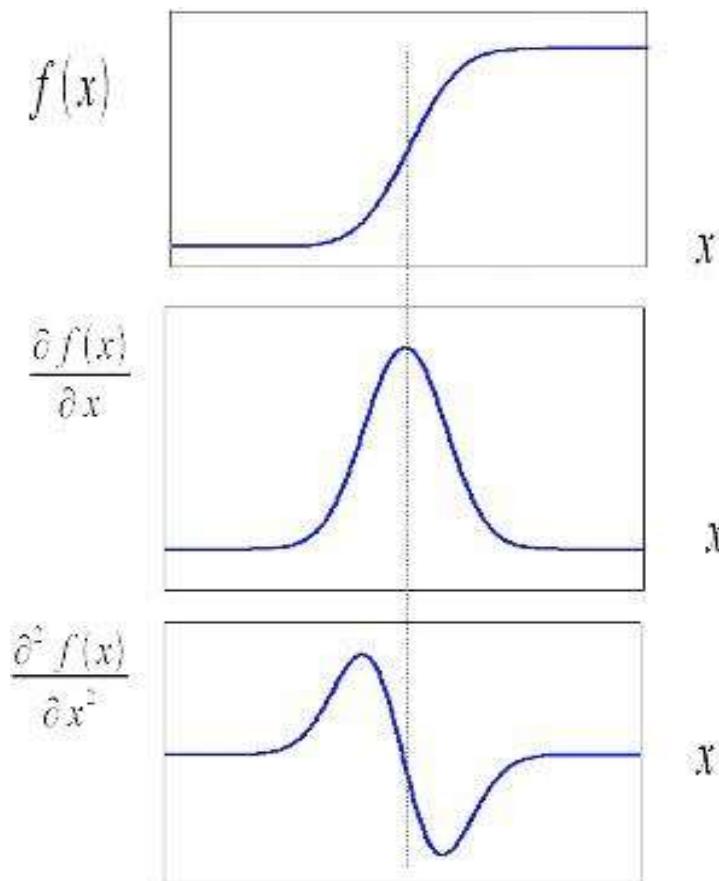
Les contours sont caractérisés par des discontinuités de la fonction d'intensité



Source : Gonzalez and Woods. Digital Image Processing 3ed. Prentice-Hall, 2008.

## DÉTECTION DE CONTOURS [2]

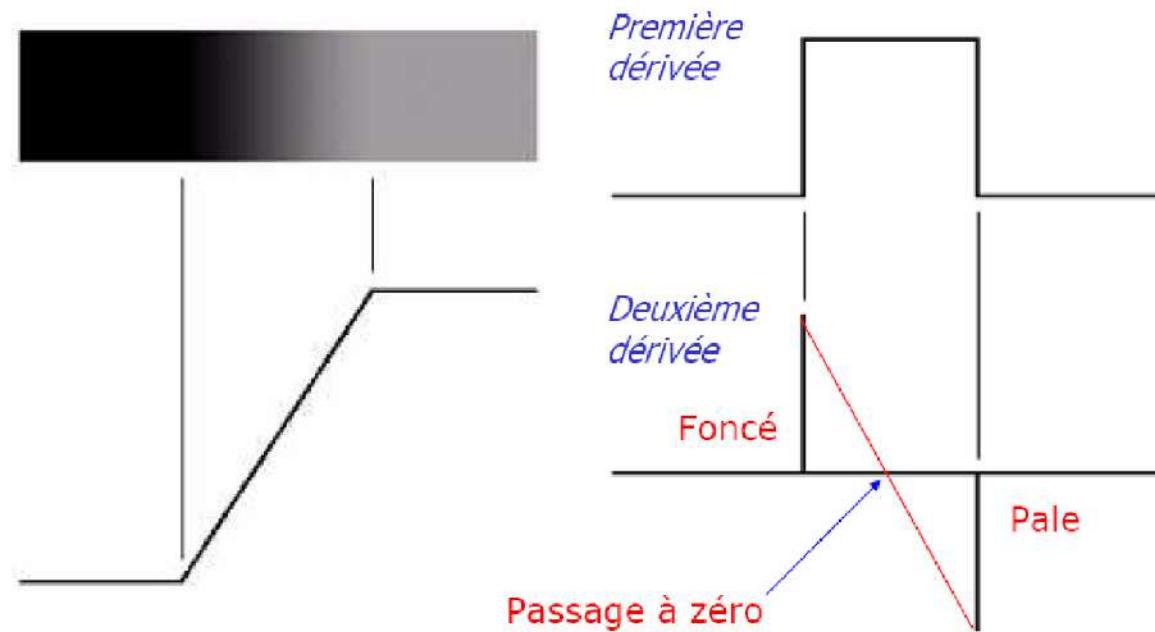
- Un contour matérialise par une rupture d'intensité dans l'image suivant une direction donnée
- Détection en 2 étapes :
  - Utilisation d'un gradient ou d'un Laplacien
  - Seuillage



# DÉTECTION DE CONTOURS

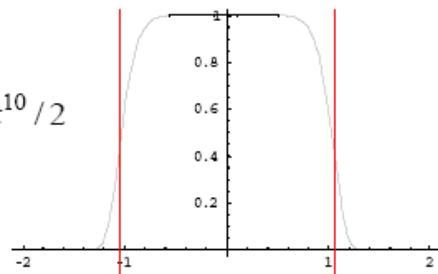
⇒ étude des dérivées de la fonction d'intensité dans l'image

la fonction d'intensité au voisinage d'un contour en *rampe* et ses dérivées première (**gradient**) et seconde (**laplacien**).

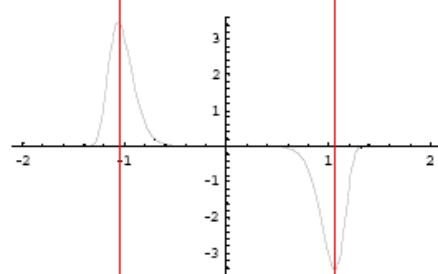


# DÉTECTION DE CONTOURS

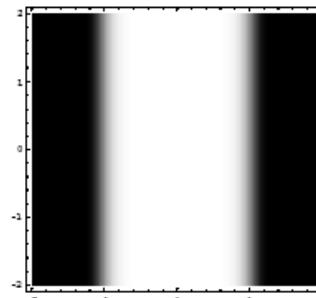
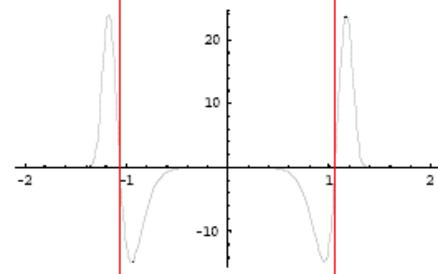
$$f(x, y) = e^{-x^{10}/2}$$



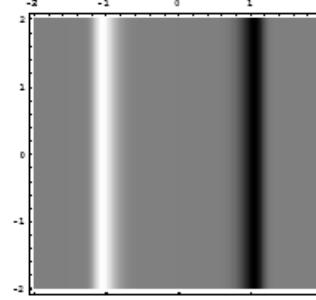
$$\frac{\partial f}{\partial x}$$



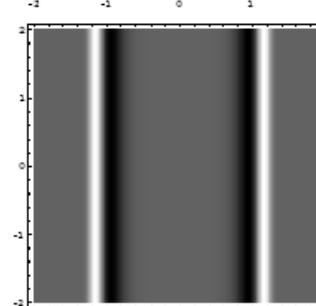
$$\frac{\partial^2 f}{\partial x^2}$$



Image



Première  
dérivée



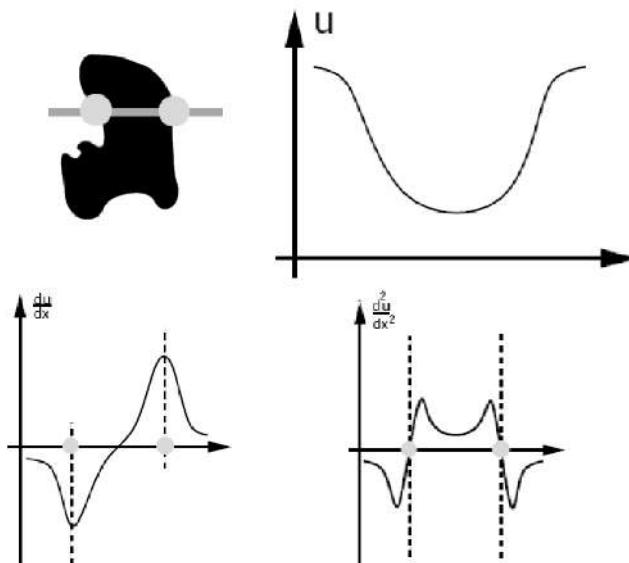
Deuxième  
dérivée

Source : Caroline Rougier. Traitement d'images (IFT2730). Univ. de Montréal.

# DÉTECTION DE CONTOURS

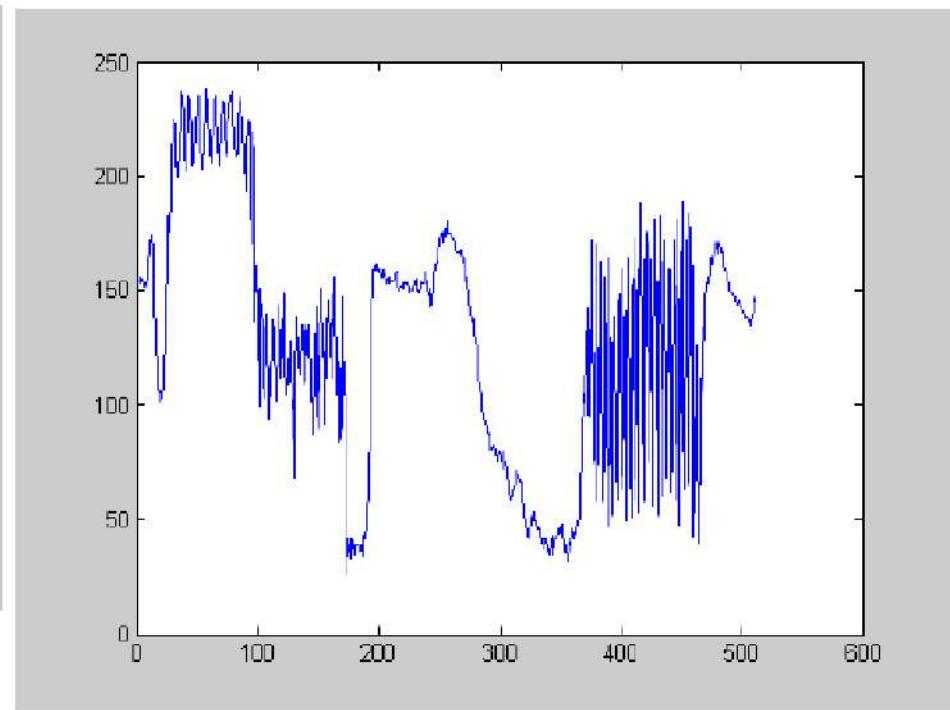
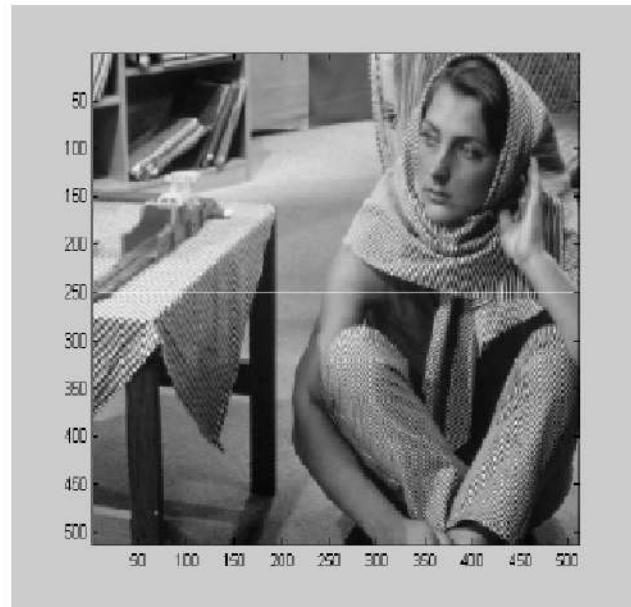
⇒ étude des dérivées de la fonction d'intensité dans l'image

- ▶ les extréma locaux du gradient de la fonction d'intensité
- ▶ les passages par zéro du laplacien
- ▶ difficulté : la présence de bruit dans les images



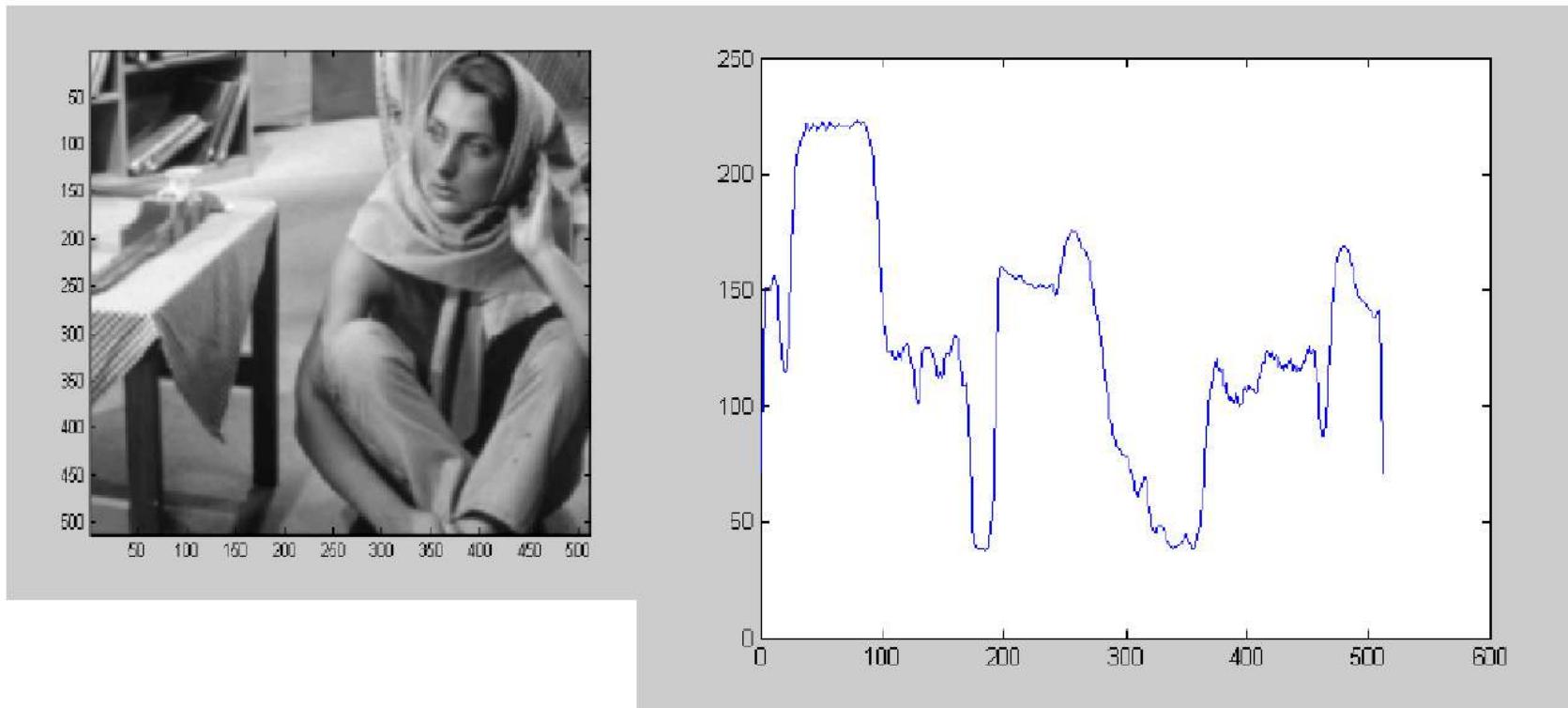
# DÉTECTION DE CONTOURS

- Gradient de l'image – étude du signal en 1D



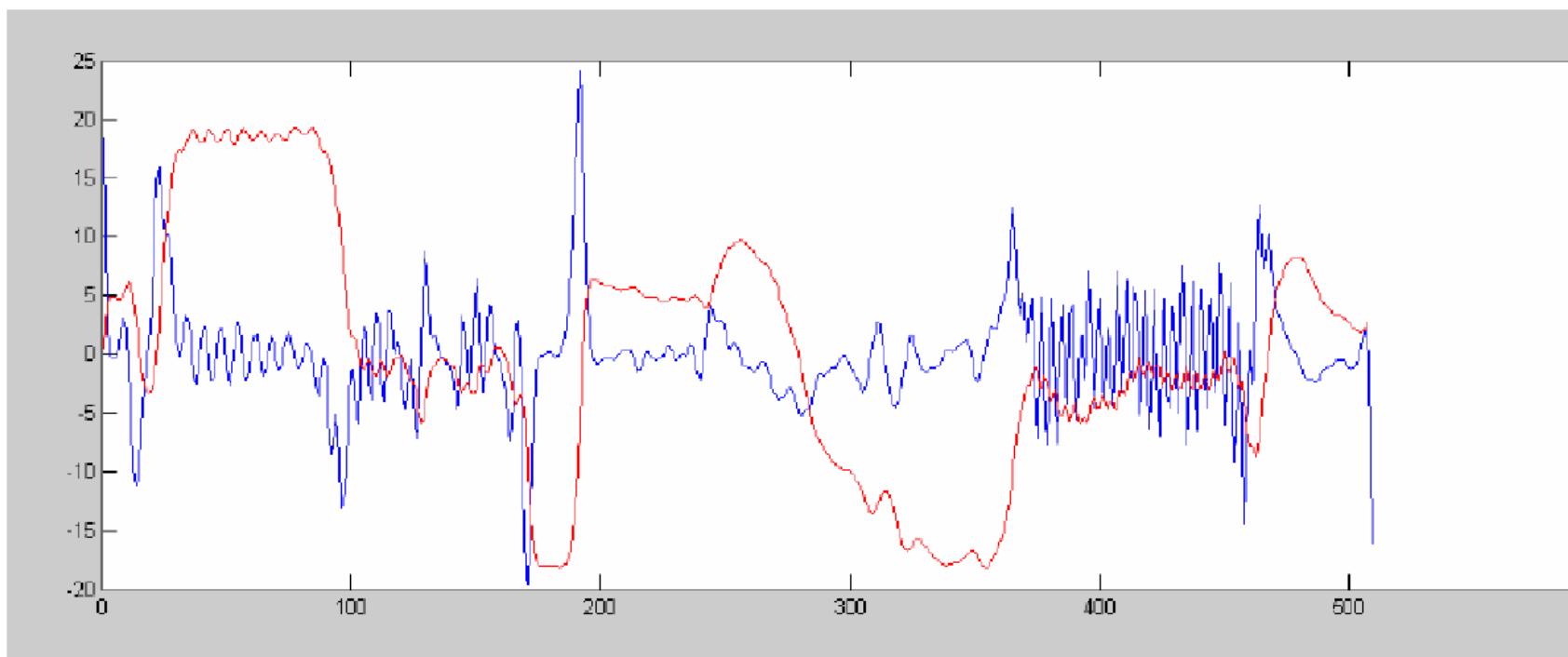
# DÉTECTION DE CONTOURS

- Gradient de l'image – étude du signal en 1D – filtre passe bas pour limiter le bruit



# DÉTECTION DE CONTOURS

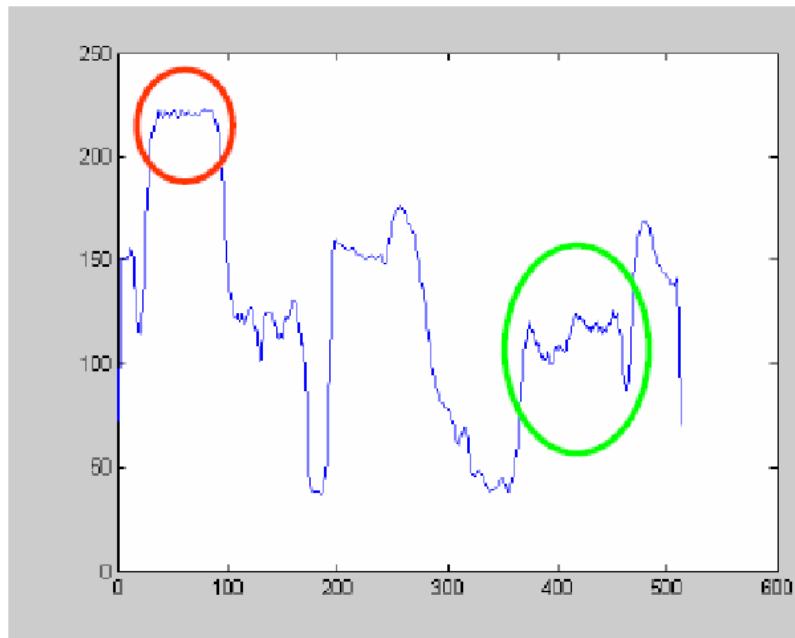
- Gradient de l'image – étude du signal en 1D  
Le signal de Barbara et sa dérivée première



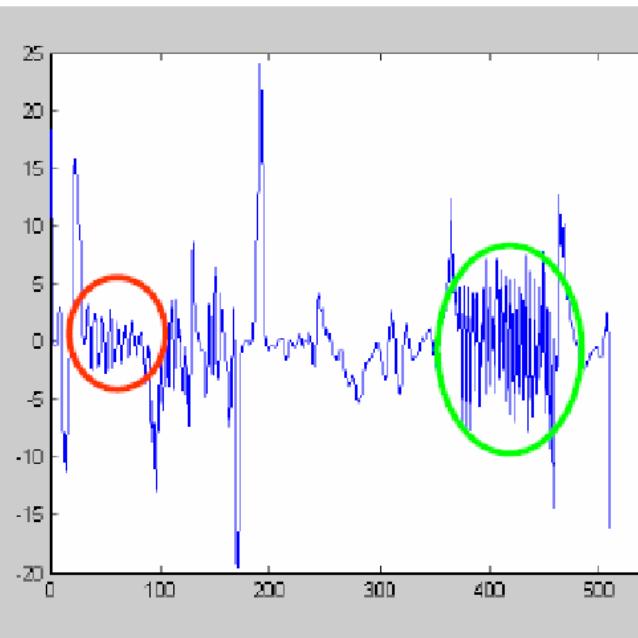
# DÉTECTION DE CONTOURS

- Gradient de l'image – étude du signal en 1D

Le signal de Barbara et sa dérivée première :  
amplitude des perturbations



signal lissé



dérivé première

# DÉTECTION DE CONTOURS

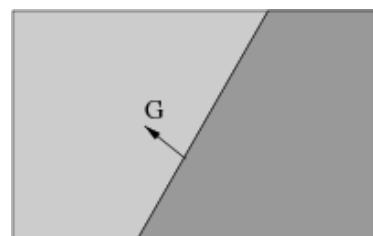
## ○ Opérateur Gradient – dérivée 1ere de l'image [6]

Rappel : l'image est une fonction

$$\begin{aligned} I : \quad S &\rightarrow \Omega \\ (x, y) &\rightarrow I(x, y) \end{aligned}$$

gradient de l'image : vecteur qui représente la variation de la fonction dépendant de plusieurs paramètres par rapport à la variation de ces différents paramètres.

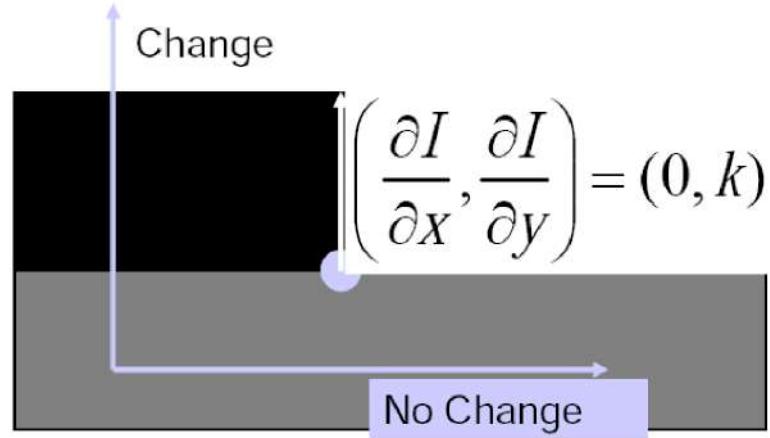
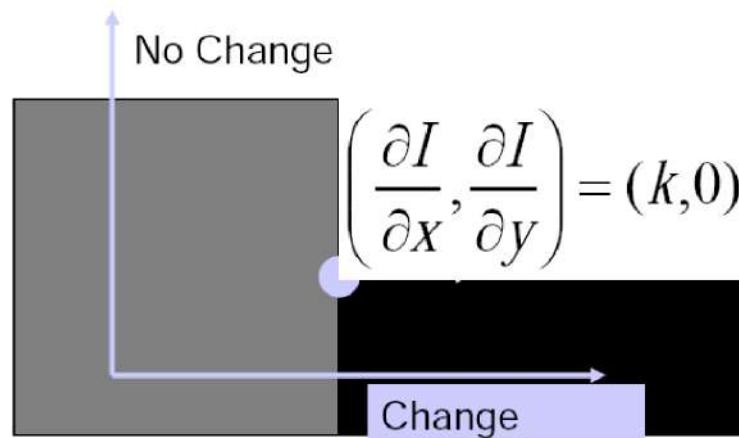
$$\nabla I = \left( \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^t$$



# DÉTECTION DE CONTOURS

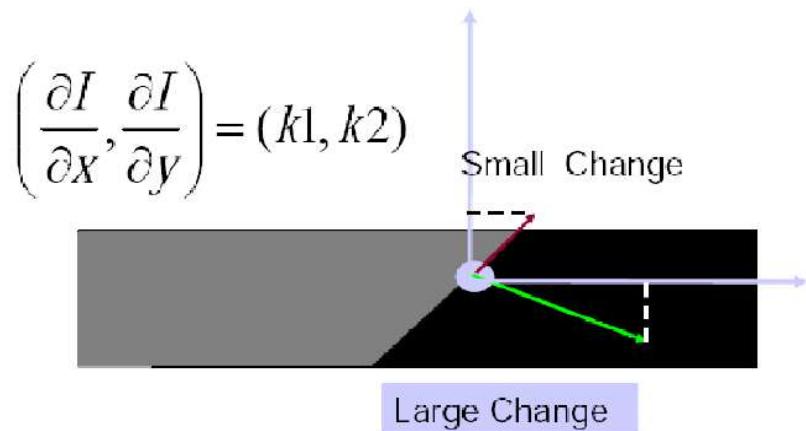
- Opérateur Gradient – dérivée 1ere de l'image [6]

$$\nabla I = \left( \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^t$$



# DÉTECTION DE CONTOURS

- Opérateur Gradient – dérivée 1ere de l'image [6]



- le gradient est un vecteur perpendiculaire au contour - l'amplitude du gradient mesure la force du contour

Le gradient est caractérisé par un module  $m$  et une direction  $\phi$  dans l'image :

$$m = \left( \frac{\partial I(x, y)}{\partial x}^2 + \frac{\partial I(x, y)}{\partial y}^2 \right)^{1/2}$$

$$\phi = \arctan \left( \frac{\partial I(x, y)}{\partial y} / \frac{\partial I(x, y)}{\partial x} \right)$$

## DÉTECTION DE CONTOURS [2]

- Opérateur gradient

- Module du gradient : permet de quantifier l'importance du contour mis en évidence

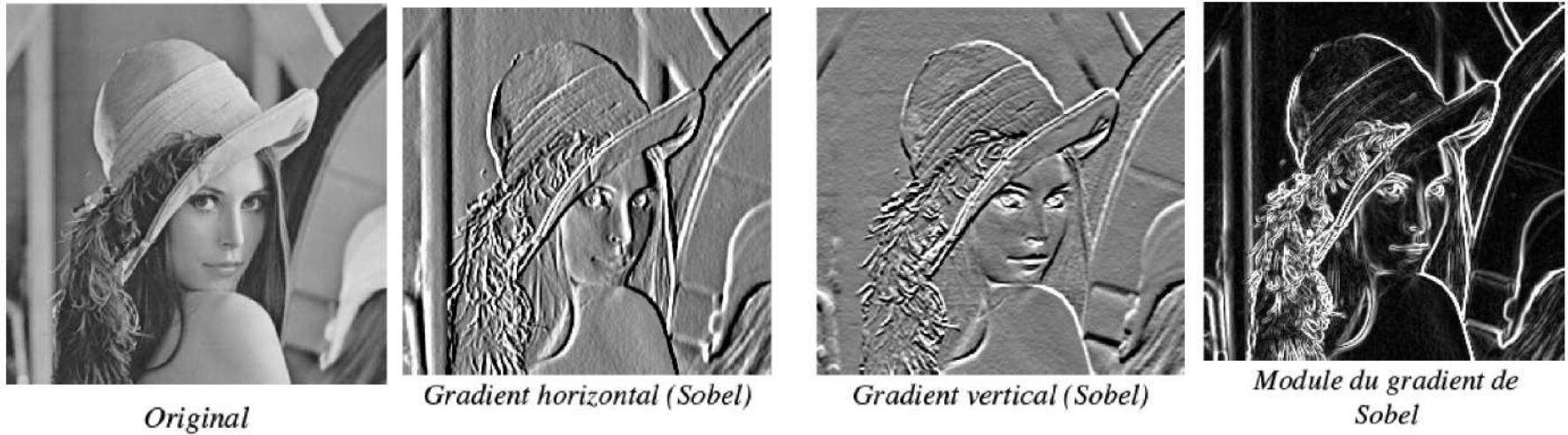
$$m = \left( \frac{\partial I(x, y)}{\partial x}^2 + \frac{\partial I(x, y)}{\partial y}^2 \right)^{1/2}$$

- Direction du gradient : permet de déterminer l'arrête présente dans l'image (la direction du gradient est orthogonale à celle du contour)

$$\phi = \arctan \left( \frac{\partial I(x, y)}{\partial y} / \frac{\partial I(x, y)}{\partial x} \right)$$

# DÉTECTION DE CONTOURS

- Opérateur Gradient



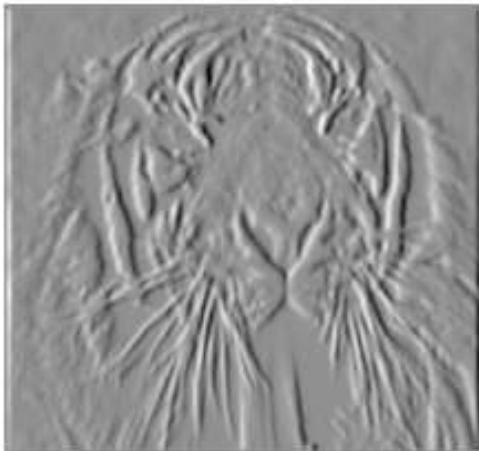
# DÉTECTION DE CONTOURS

- Opérateur Gradient

Original  
Image



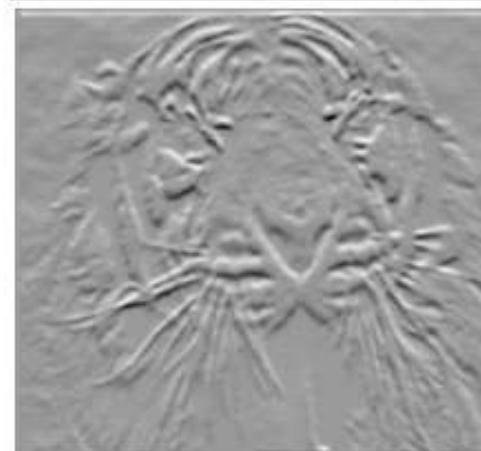
x-direction



Gradient  
magnitude



y-direction

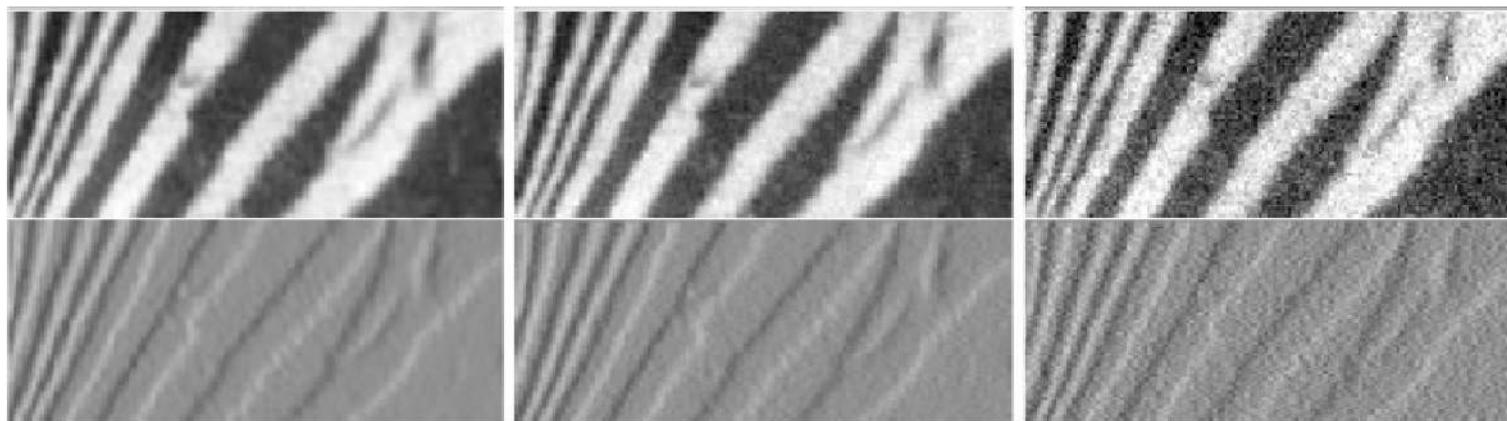


[Extrait de [5]]

# DÉTECTION DE CONTOURS

- Opérateur Gradient

La qualité du gradient se dégrade avec l'accroissement du bruit :



Le gradient d'une image filtrée :

$$\nabla I'(x, y) = \nabla(I(x, y) * h(x, y)) = \nabla I(x, y) * h(x, y) = I(x, y) * \nabla h(x, y).$$

# DÉTECTION DE CONTOURS

## ○ Opérateur Gradient

Approche 1 : par seuillage du gradient

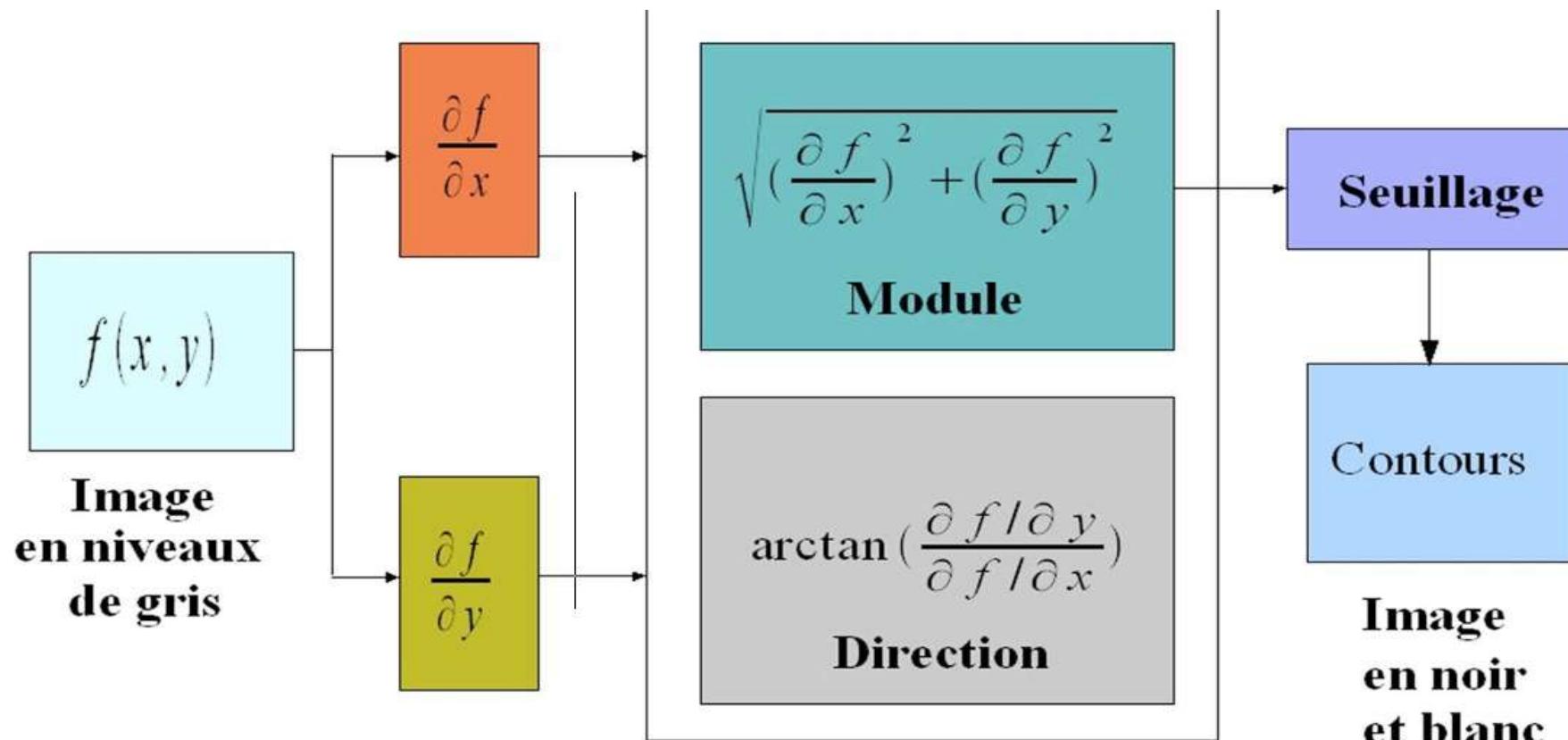
les points de contour dans une image sont caractérisés par des extréma locaux du gradient. Une première approche consiste donc à :

1. calculer la norme du gradient en tous point de l'image,
2. sélectionner les pixels à l'aide d'un seuil fixé *a priori* pour la norme du gradient.

pbl : ne permet pas de différencier efficacement les points de contour du bruit.

# DÉTECTION DE CONTOURS [2]

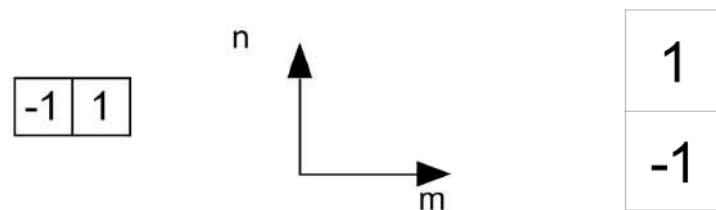
- Opérateur gradient



# DÉTECTION DE CONTOURS [2]

## ○ Opérateur gradient

- Gradients par masque
  - Pour chaque opérateur, deux masques sont utilisés de façon à déterminer le gradient de l'image dans deux directions orthogonales.
- Approximation de base



# DÉTECTION DE CONTOURS

- Opérateur Gradient

## Dérivation par différence finies

Une image est discrète par nature. Les premières approches ont donc consisté à approximer les dérivées par différence :

$$\nabla_x I(x, y) = I(x, y) - I(x - n, y)$$

ou :

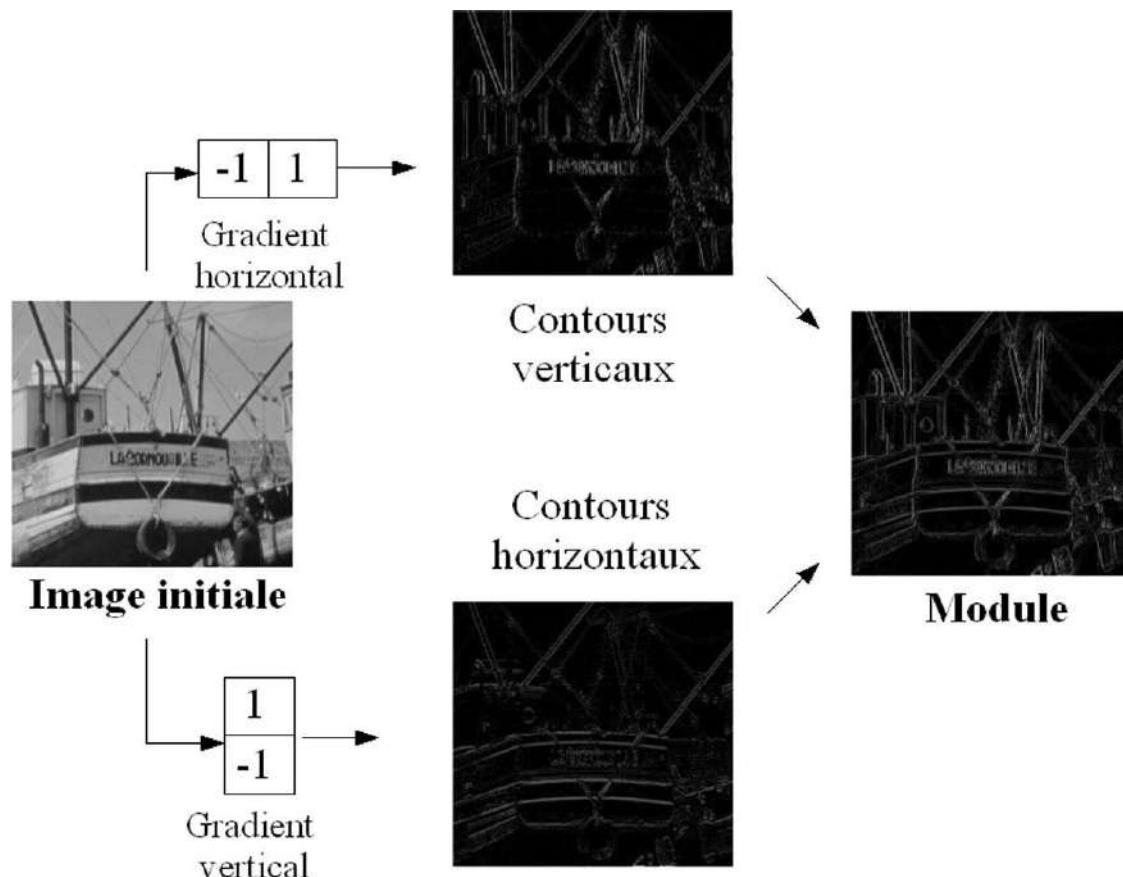
$$\nabla_x I(x, y) = I(x + n, y) - I(x - n, y)$$

avec, en général  $n = 1$ .

Ces dérivées sont calculées par convolution de l'image avec un masque de différences.

# DÉTECTION DE CONTOURS [2]

- Opérateur gradient
  - Approximation de base



# DÉTECTION DE CONTOURS

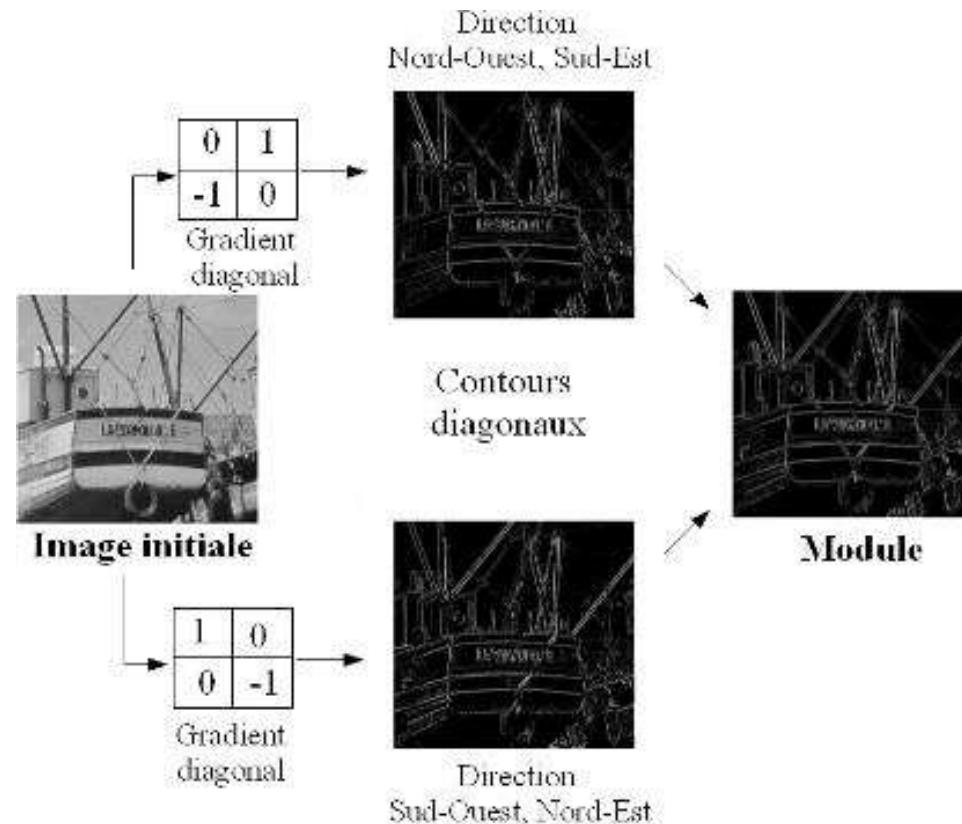
- Opérateur gradient

Proposez un masque pour détecter les contours diagonaux

# DÉTECTION DE CONTOURS [2]

- Opérateur gradient
  - Opérateur de Roberts

Ce masque proposé en 1965 permet de calculer un gradient le long des diagonales de l'image



## DÉTECTION DE CONTOURS [2]

- Opérateur gradient
  - Opérateur de Prewitt et Sobel

1	c	1
0	0	0
-1	-c	-1

-1	0	1
-c	0	c
-1	0	1

Chaque pixel des masques est normalisé par  $1/(c+2)$

# DÉTECTION DE CONTOURS [2]

- Opérateur gradient
  - Opérateur de Prewitt et Sobel

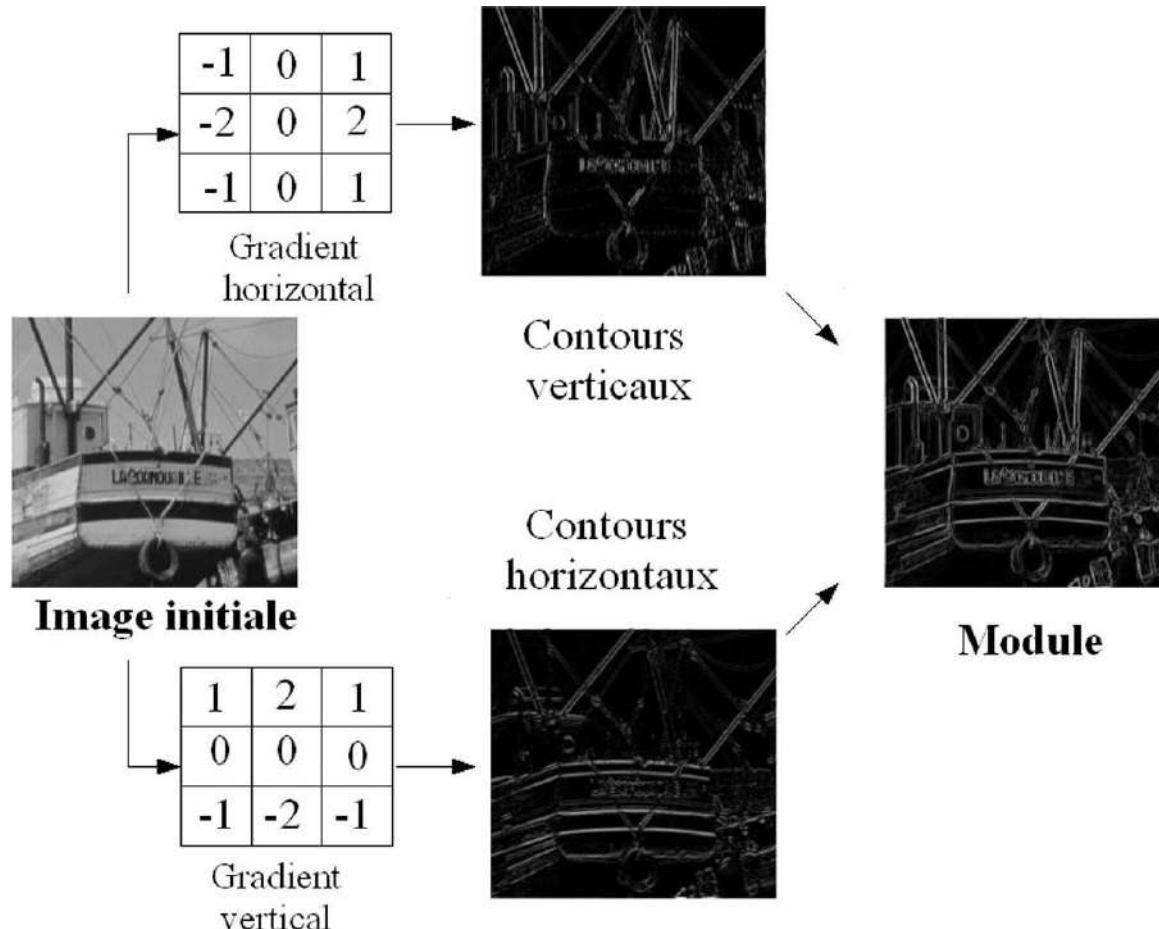
Dérivation par différence finies - Opérateurs de Prewitt et de Sobel

$$h1 = 1/(c+2) \begin{bmatrix} -1 & 0 & 1 \\ -c & 0 & c \\ -1 & 0 & 1 \end{bmatrix} \quad h2 = 1/(c+2) \begin{bmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{bmatrix}$$

- ▶  $c=1$  : opérateur de Prewitt
- ▶  $c=2$ , opérateur de Sobel.
- ▶ ces masques effectuent un lissage dans la direction orthogonale. Ce lissage rend ces masques un peu moins sensibles au bruit que les précédents.

# DÉTECTION DE CONTOURS [2]

- Opérateur gradient
  - Opérateur de Sobel



# DÉTECTION DE CONTOURS

- Opérateur de gradients

Opérateurs de Prewitt et de Sobel séparables

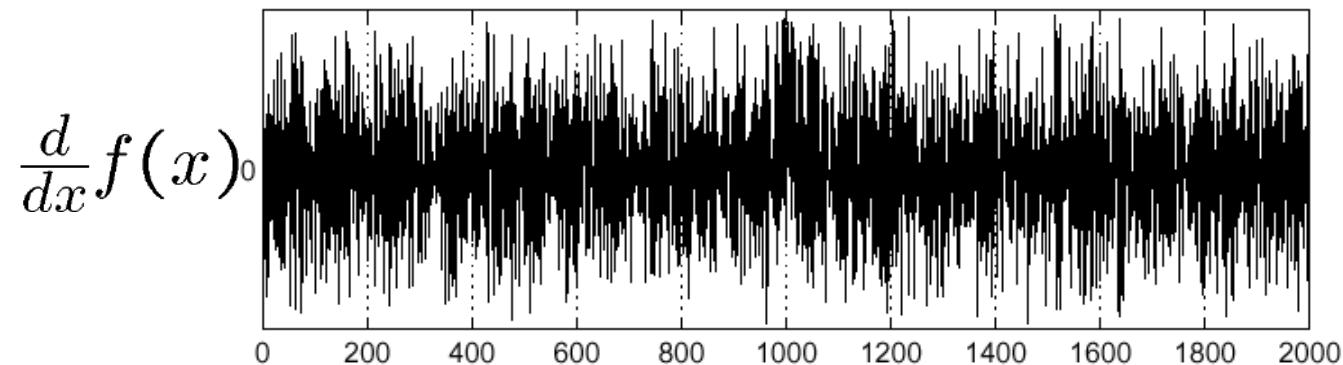
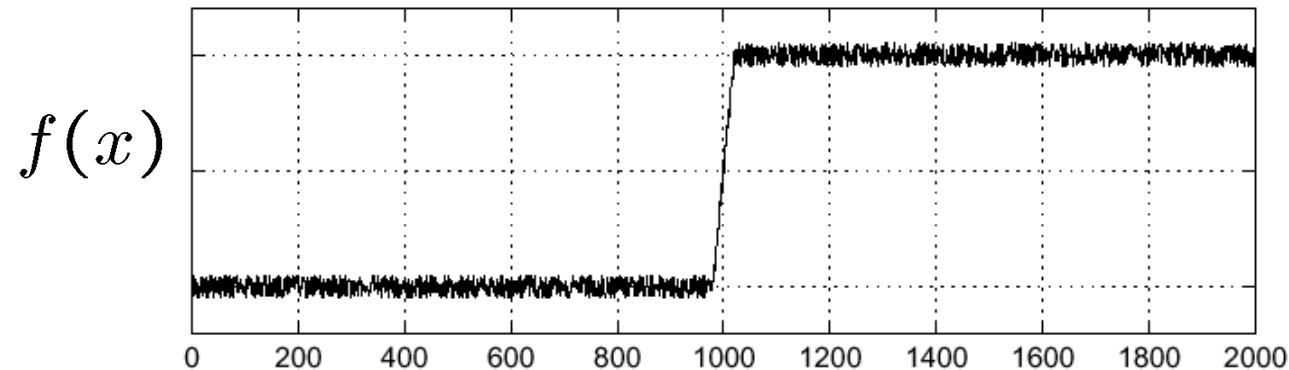
$$h1 = 1/(c+2) \begin{bmatrix} -1 & 0 & 1 \\ -c & 0 & c \\ -1 & 0 & 1 \end{bmatrix} = 1/(c+2) \begin{bmatrix} 1 \\ c \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$h2 = 1/(c+2) \begin{bmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{bmatrix} = 1/(c+2) \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & c & 1 \end{bmatrix}$$

filtres isotropes ou anisotropes ?

# DÉTECTION DE CONTOURS

- Opérateur Gradient
  - Effet du bruit



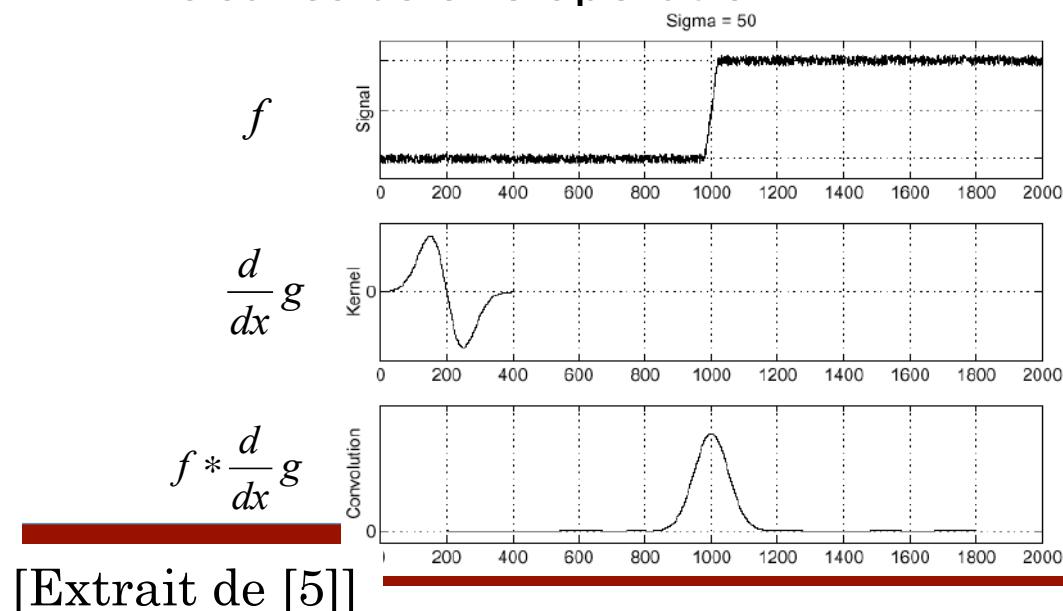
[Extrait de [5]]

# DÉTECTION DE CONTOURS

- Opérateur Gradient
  - Effet du bruit – flouter d'abord
  - This theorem gives us a very useful property:

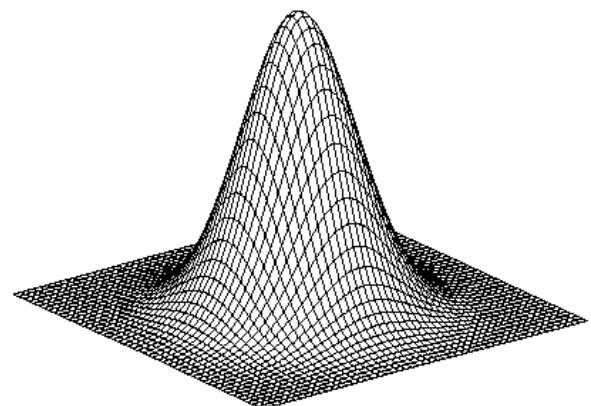
$$\frac{d}{dx}(f * g) = f * \frac{d}{dx}g$$

- This saves us one operation:



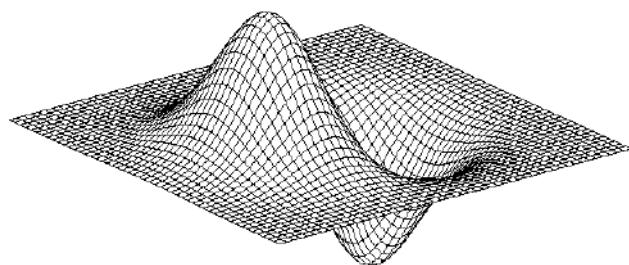
# DÉTECTION DE CONTOURS

- Opérateur Gradient
  - Effet du bruit – Dérivée du filtre Gaussien



2D-gaussian

$$* [1 \ -1] =$$

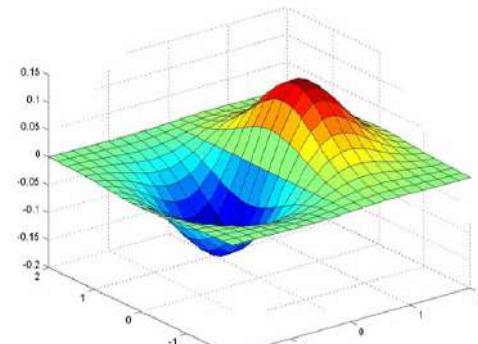


x - derivative

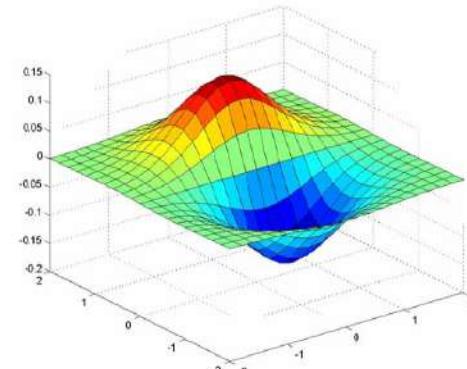
[Extrait de [5]]

# DÉTECTION DE CONTOURS

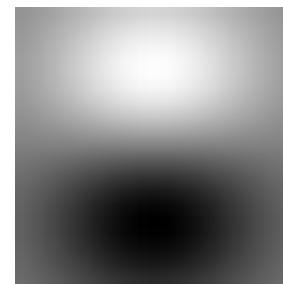
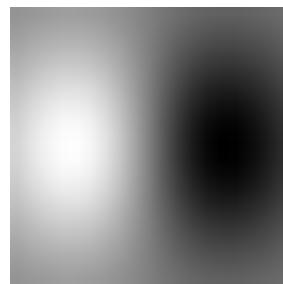
- Opérateur Gradient
  - Effet du bruit – Dérivée du filtre Gaussien



x-direction



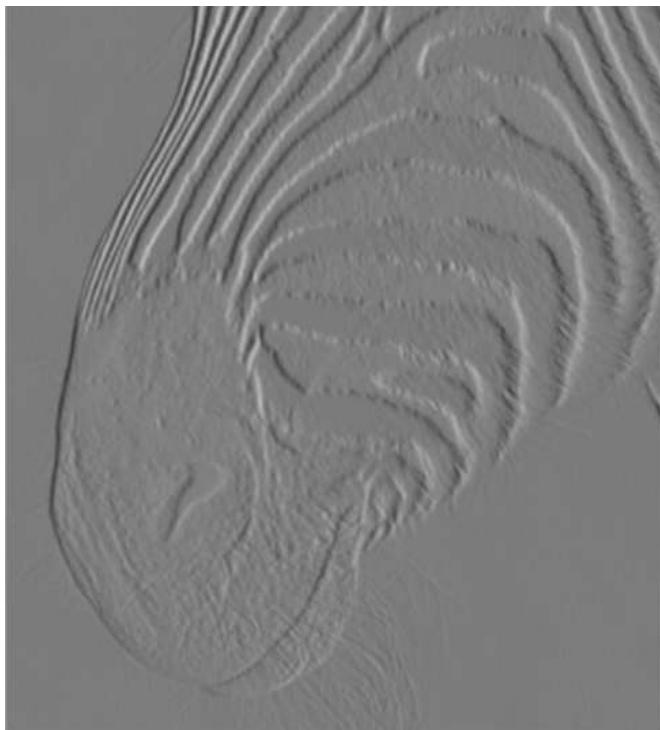
y-direction



[Extrait de [5]]

# DÉTECTION DE CONTOURS

- Opérateur Gradient
  - Effet du bruit – Équilibre à trouver entre flou et localisation

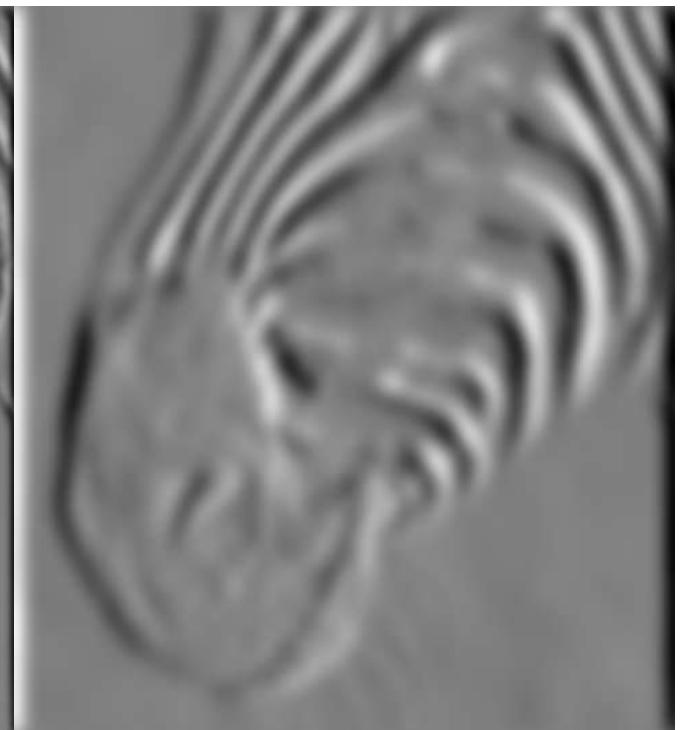


1 pixel

[Extrait de [5]]



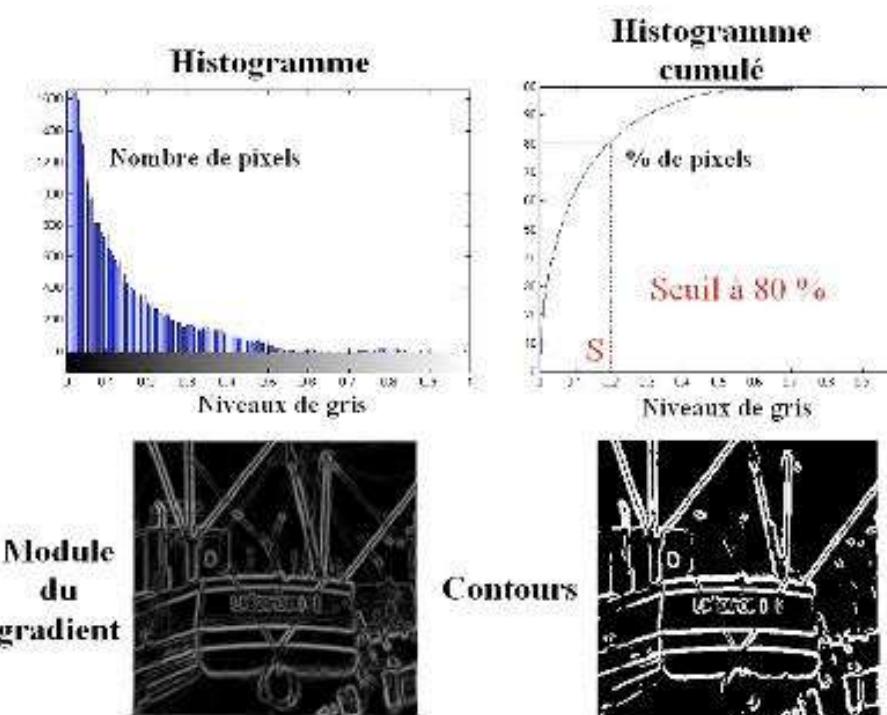
3 pixels



7 pixels

# DÉTECTION DE CONTOURS [2]

- Méthodes de seuillage
  - Seuillage global : fixer un seuil
  - Seuillage global par histogramme



# DÉTECTION DE CONTOURS [2]

- Méthodes de seuillage

Approche 1 : par seuillage du gradient



Gradient



Gradient seuillé ( $|G| > G_{min}$ )



Seuil faible



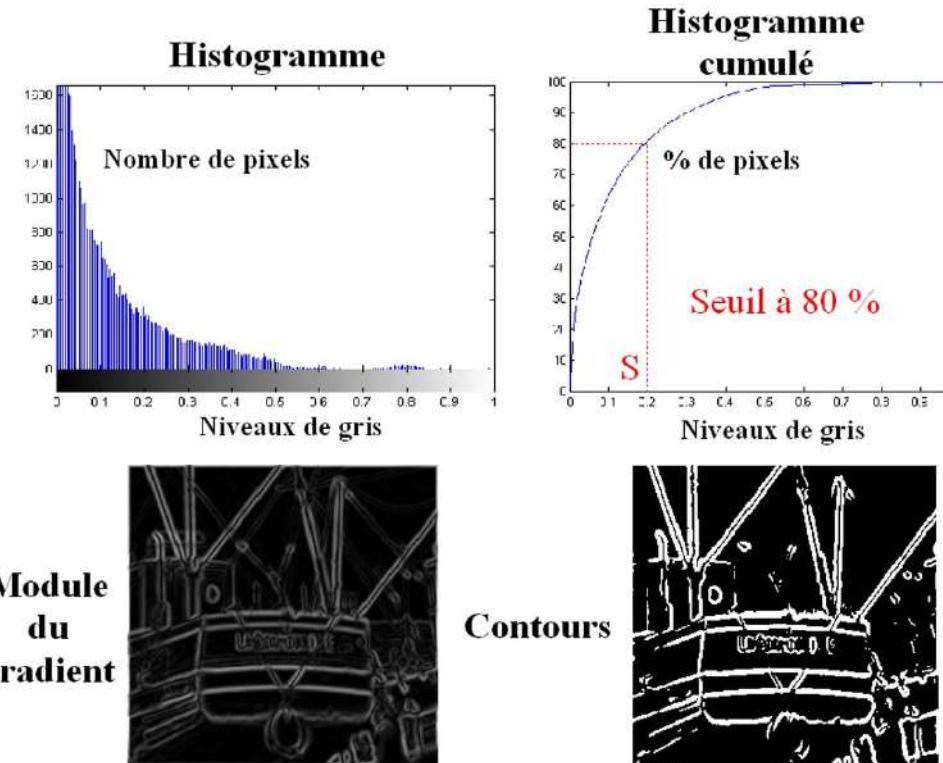
Seuil grand

# DÉTECTION DE CONTOURS [2]

## ○ Méthodes de seuillage

Approche 1 : par seuillage du gradient

possibilité de calculer le seuil en tenant compte de l'histogramme du module du gradient



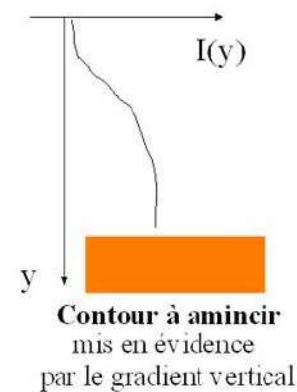
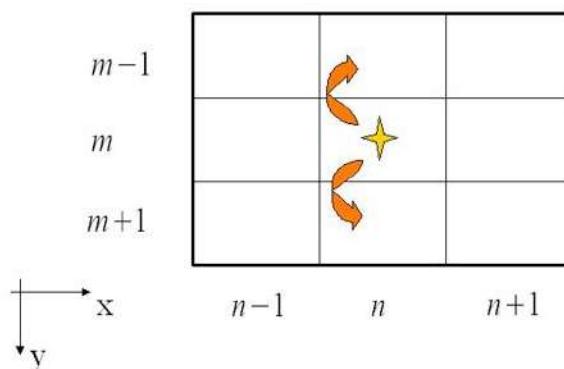
# DÉTECTION DE CONTOURS [2]

- Méthodes de seuillage
  - Par amincissement

Le pixel en  $(m, n)$  est un contour horizontal si :

$$\begin{array}{ll} |g_V| > |g_H| & g_H \text{ gradient horizontal} \\ g_V \text{ gradient vertical} & \\ G(m, n) > S & G \text{ module du gradient} \end{array}$$

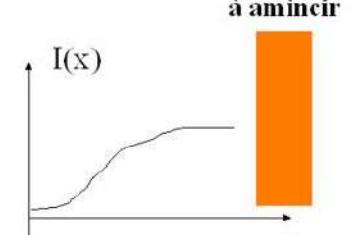
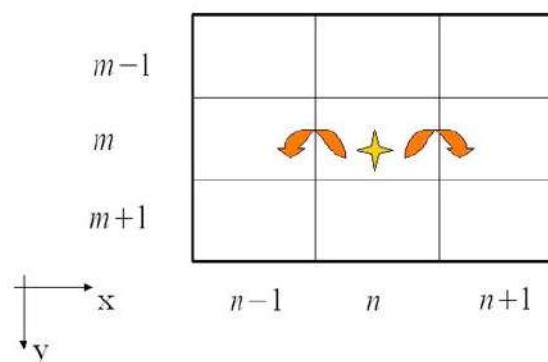
$$(G(m-1, n) \leq G(m, n)) \wedge (G(m, n) > G(m+1, n))$$



Le pixel en  $(m, n)$  est un contour vertical si :

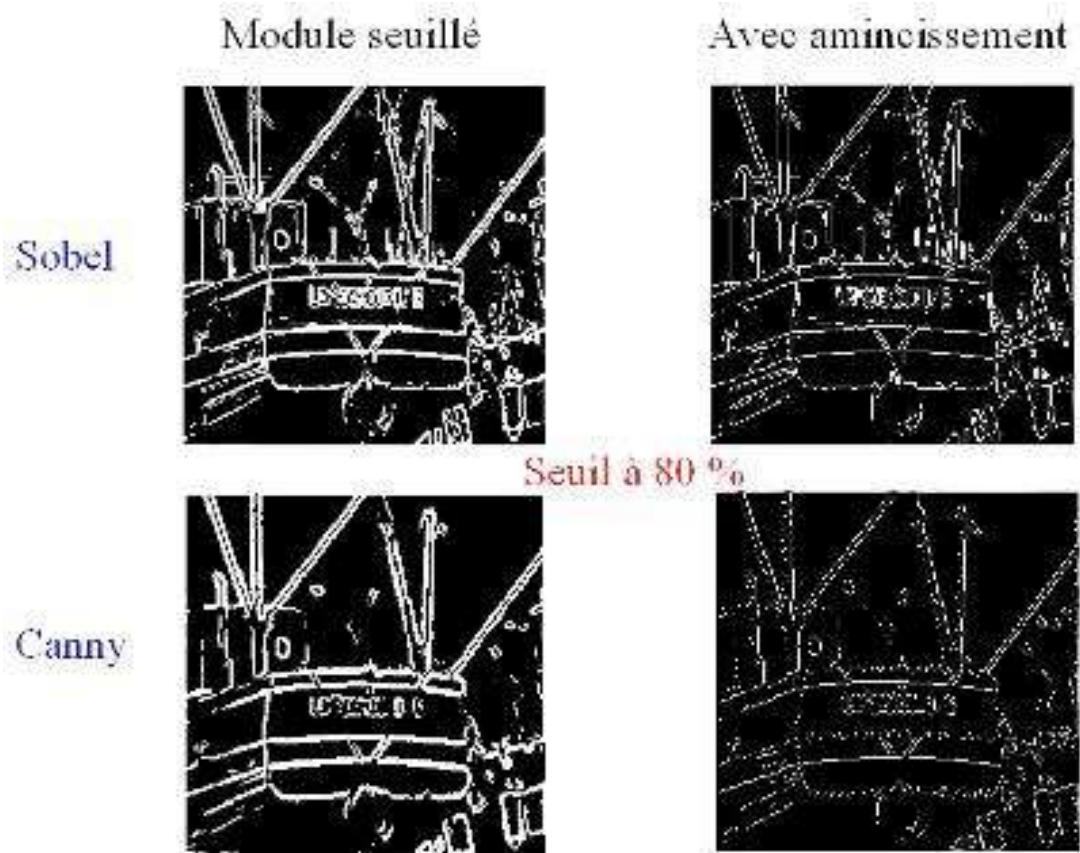
$$\begin{array}{ll} |g_H| > |g_V| & g_H \text{ gradient horizontal} \\ g_V \text{ gradient vertical} & \\ G(m, n) > S & G \text{ module du gradient} \end{array}$$

$$(G(m, n-1) \leq G(m, n)) \wedge (G(m, n) > G(m, n+1))$$



## DÉTECTION DE CONTOURS [2]

- Méthodes de seuillage
  - Par amincissement

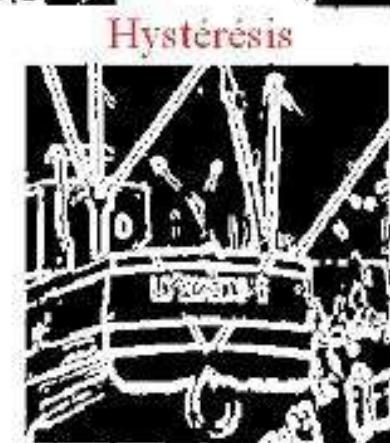
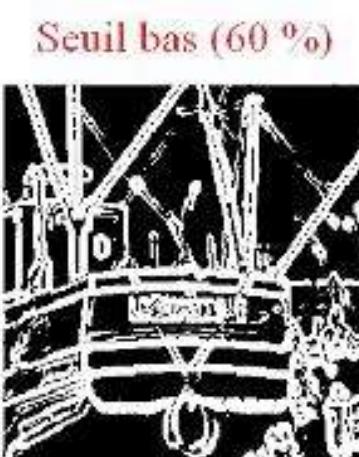
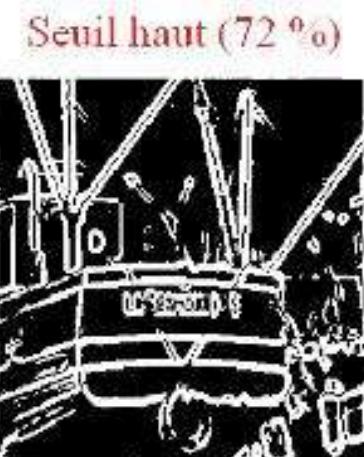


## DÉTECTION DE CONTOURS [2]

- Seuillage local par hystérésis
  - Algorithme
    - Extraction des pixels de niveau supérieur à  $S_h$ . Ce sont les pixels  $P_1$ .
    - Extraction des pixels de niveau compris entre  $S_b$  et  $S_h$ . Ce sont les pixels  $P_2$ .
    - Sélection pour représenter les contours des pixels  $P_1$  et des pixels  $P_2$ , connexes aux pixels  $P_1$ .

## DÉTECTION DE CONTOURS [2]

- Seuillage local par hystérésis

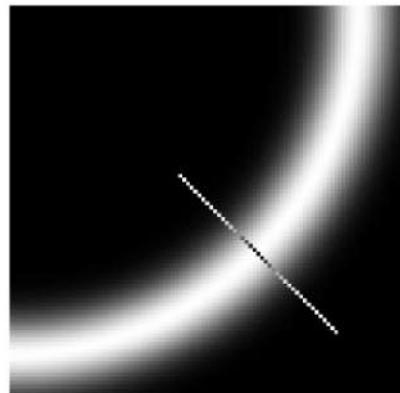
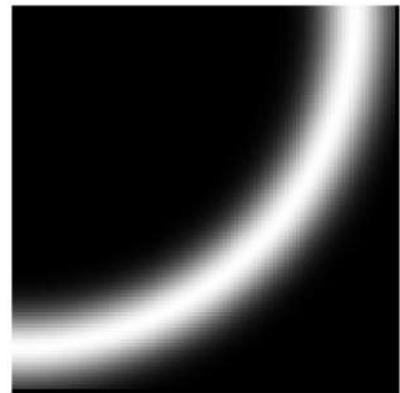
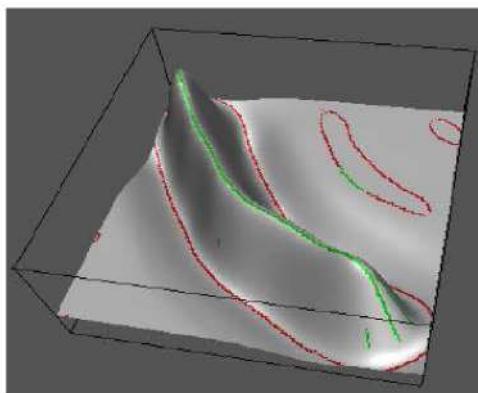


# DÉTECTION DE CONTOURS [2]

- Seuillage local par hystérésis

Approche 2 : par recherche de maxima et seuillage par hystérésis

- (1) Extraction des extréma locaux du gradient dans la direction du gradient. Cela revient à déterminer, pour un pixel  $p$  donné, les valeurs du gradient sur la droite passant  $p$  et de direction celle de son gradient. On vérifie ensuite que le gradient en  $p$  est bien localement maximal sur cette droite.



# DÉTECTION DE CONTOURS [2]

## ○ Seuillage local par hystérésis

Approche 2 : par recherche de maxima et seuillage par hystérésis

(2) **Seuillage par hystérésis des extréma.** Cette étape repose sur une hypothèse de connexité. Le principe est d'utiliser deux seuils pour la norme du gradient :  $s_b$  et  $s_h$  et de sélectionner les pixels pour lesquels :

1. la norme du gradient est supérieure à  $s_h$ ,
2. le pixel donné est connecté, par un chemin constitué de pixels dont la norme du gradient est supérieure à  $s_b$ , à un pixel pour lequel la norme du gradient est supérieure à  $s_h$ .

- remarque : en général, on prend  $s_h/s_b = 2$

## DÉTECTION DE CONTOURS [2]

- Seuillage local par hystérésis

Approche 2 : par recherche de maxima et seuillage par hystérésis



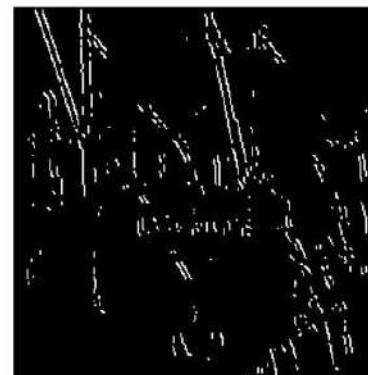
## DÉTECTION DE CONTOURS [2]

- Seuillage local par extraction des maxima du gradient dans une direction
  - Comparer la norme du gradient en tout point  $M(i,j)$  de l'image avec celles de ses deux voisins  $M_1$  et  $M_2$  situés dans la direction du gradient à une distance unitaire de part et d'autre de  $M$

## DÉTECTION DE CONTOURS [2]

- Seuillage local par extraction des maxima du gradient dans une direction

Direction 1



Module gradient



Direction 2



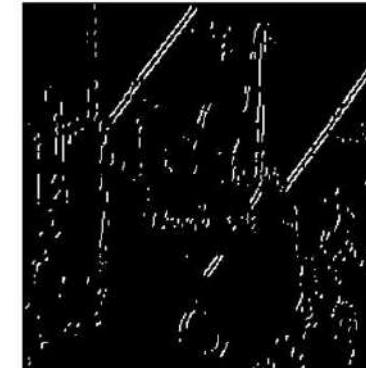
Direction 3



Maxima locaux



Direction 4



## DÉTECTION DE CONTOURS [2]

- Opérateur Laplacien
  - opérateur laplacien : calcul de la dérivée seconde

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	-8	-1
-1	-1	-1

1	-2	1
-2	4	-2
1	-2	1

# DÉTECTION DE CONTOURS

## ○ Opérateur Laplacien

Les points de contour sont caractérisés par des **passages par zéro du laplacien**. La détection de ces points s'effectue en deux étapes :

1. Détection des passages par zéros. Les pixels pour lesquels le laplacien change de signe sont sélectionnés.
2. Seuillage des passages par zéros de fortes amplitudes.

# DÉTECTION DE CONTOURS

- Opérateur Laplacien

dérivées secondes

$$\begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix}$$

laplacien de l'image : somme des dérivées secondes non mixtes

$$\Delta I = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

# DÉTECTION DE CONTOURS

## ○ Opérateur Laplacien

L'estimation du laplacien d'une image se fait de la même manière par convolution de l'image avec un masque. Le laplacien est approché par différences finies :

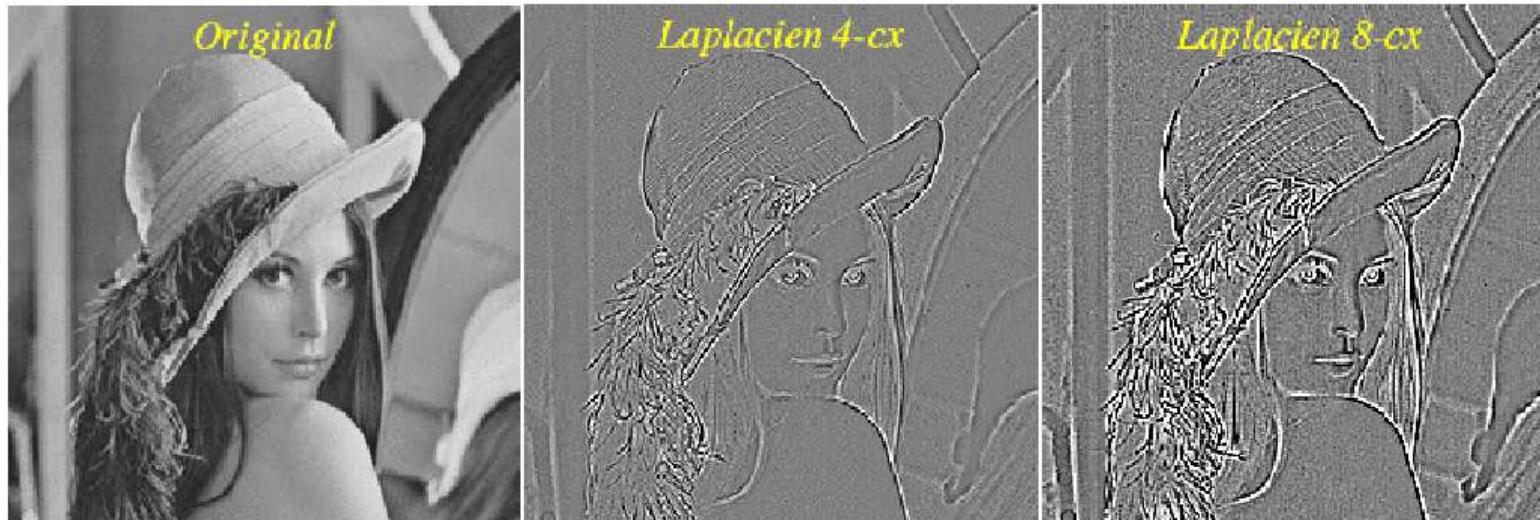
$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

ou :

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

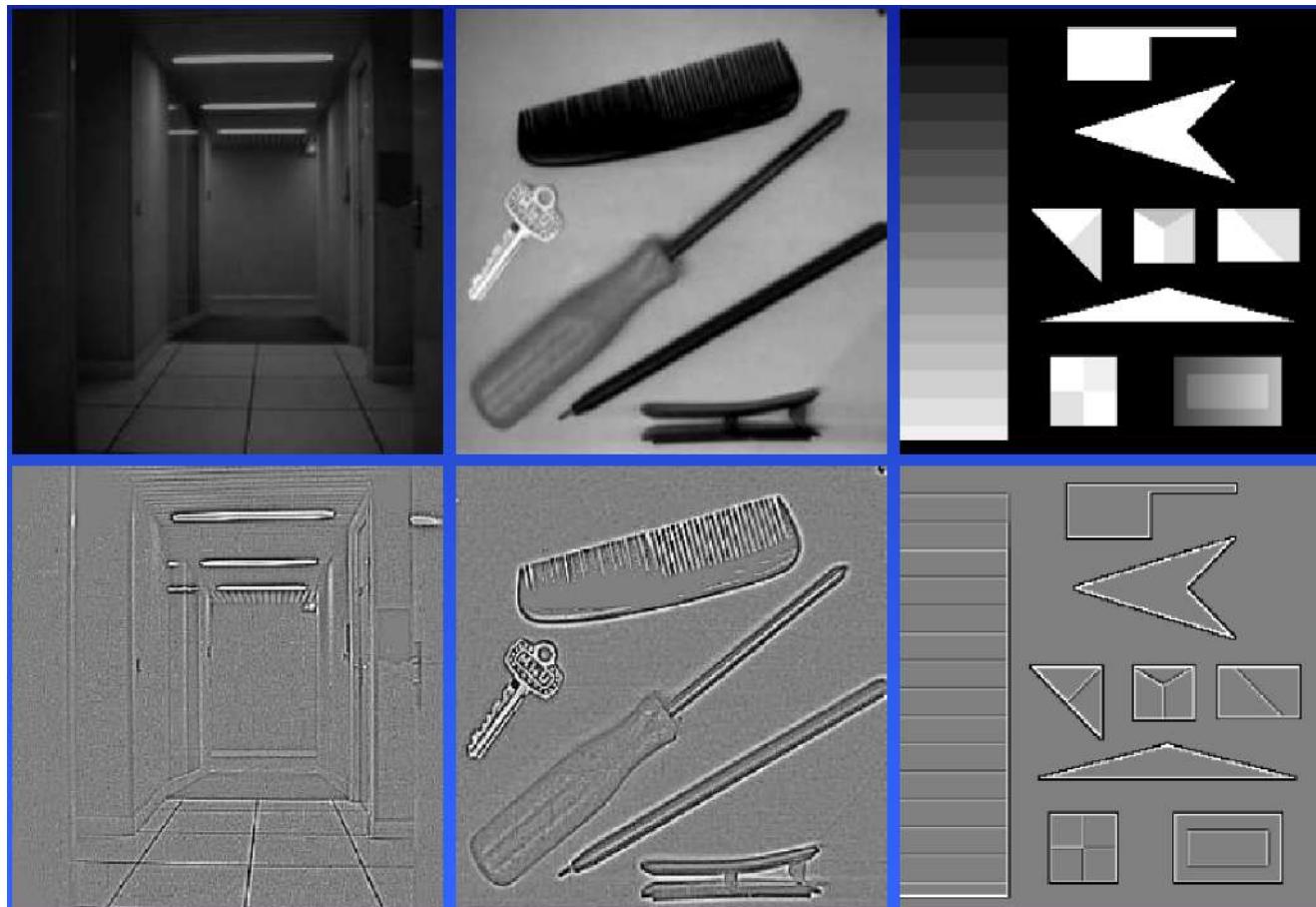
# DÉTECTION DE CONTOURS

- Opérateur Laplacien



# DÉTECTION DE CONTOURS

- Opérateur Laplacien



# DÉTECTION DE CONTOURS

- Opérateur Laplacien

	$\Delta_1$	
$\Delta_2$	$\Delta$	$\Delta_3$
	$\Delta_4$	

Si  $\Delta > 0$  et l'un des autres  $\Delta_i \leq 0$  ou  $\Delta < 0$  et l'un des autres  $\Delta_i \geq 0$  alors on considère qu'il ya changement de signe.

# DÉTECTION DE CONTOURS

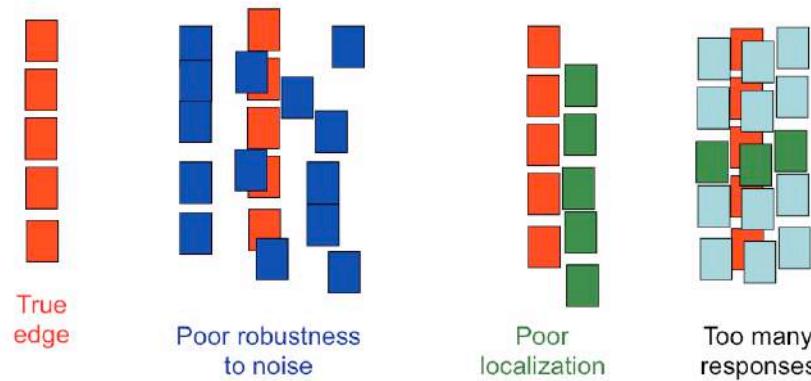
- Opérateur Laplacien



## DÉTECTION DE CONTOURS [5]

- Caractéristiques d'un détecteur de contour optimal

- Bonne détection : minimisation de la probabilité de faux positifs et faux négatifs
- Bonne localisation : détection du contour au plus proche du contour réel
- Réponse unique : le détecteur ne doit retourner qu'un point par point de contour réel en minimisant les maximum locaux sur le contour réel



## DÉTECTION DE CONTOURS [2]

- Filtrage de Canny

- Principe

- approche analytique

- trois critères :

- un critère de bonne détection garantissant une réponse forte en sortie du filtre même en présence de faibles contours sur l'image d'entrée,

- un critère de bonne localisation du contour,

- un critère d'unicité de la réponse permettant d'assurer une seule détection pour un contour et ainsi d'éviter les effets de rebond.

## DÉTECTION DE CONTOURS [2]

- Filtrage de Canny

- Recherche de  $h(x)$  filtre optimal pour la détection d'une marche d'escalier sous l'hypothèse d'un bruit additif indépendant du signal
- Modèle de contour : marche d'escalier

$$e(x) = A Y(x) + n(x)$$

où A représente l'amplitude du saut d'intensité,  $n(x)$  un bruit blanc additif indépendant de l'arête et  $Y(x)$  la fonction d'*heaviside* définie telle que :

$$Y(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$$

# DÉTECTION DE CONTOURS

- Filtrage de Canny – mise en œuvre
  - Réduction du bruit de l'image originale
    - Filtre Gaussien

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Masque 5x5 discret  
avec  $\sigma = 1,4$

$$\frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

## DÉTECTION DE CONTOURS [2]

- Filtrage de Canny
  - Gradient d'intensité (ex Sobel)

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad ; \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

La valeur du gradient en un point est approximée par la formule :

$$|G| = |G_x| + |G_y|$$

## DÉTECTION DE CONTOURS [2]

- Filtrage de Canny
  - Direction des contours

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

- Obtention d'une carte des gradients d'intensité en chaque point de l'image accompagnée des directions des contours

# DÉTECTION DE CONTOURS [2]

## ○ Filtrage de Canny

- Seuillage des contours
  - Différenciation des contours sur la carte générée par seuillage à hystérésis.
  - Nécessité de deux seuils, un haut et un bas, comparés à l'intensité du gradient de chaque point.
  - Critère de décision : pour chaque point, si l'intensité de son gradient est:
    - Inférieur au seuil bas, le point est rejeté;
    - Supérieur au seuil haut, le point est accepté comme formant un contour;
    - Entre le seuil bas et le seuil haut, le point est accepté si il est connecté à un point déjà accepté.
  - Obtention d'une image binaire avec d'un côté les pixels appartenant aux contours et les autres.

## DÉTECTION DE CONTOURS [2]

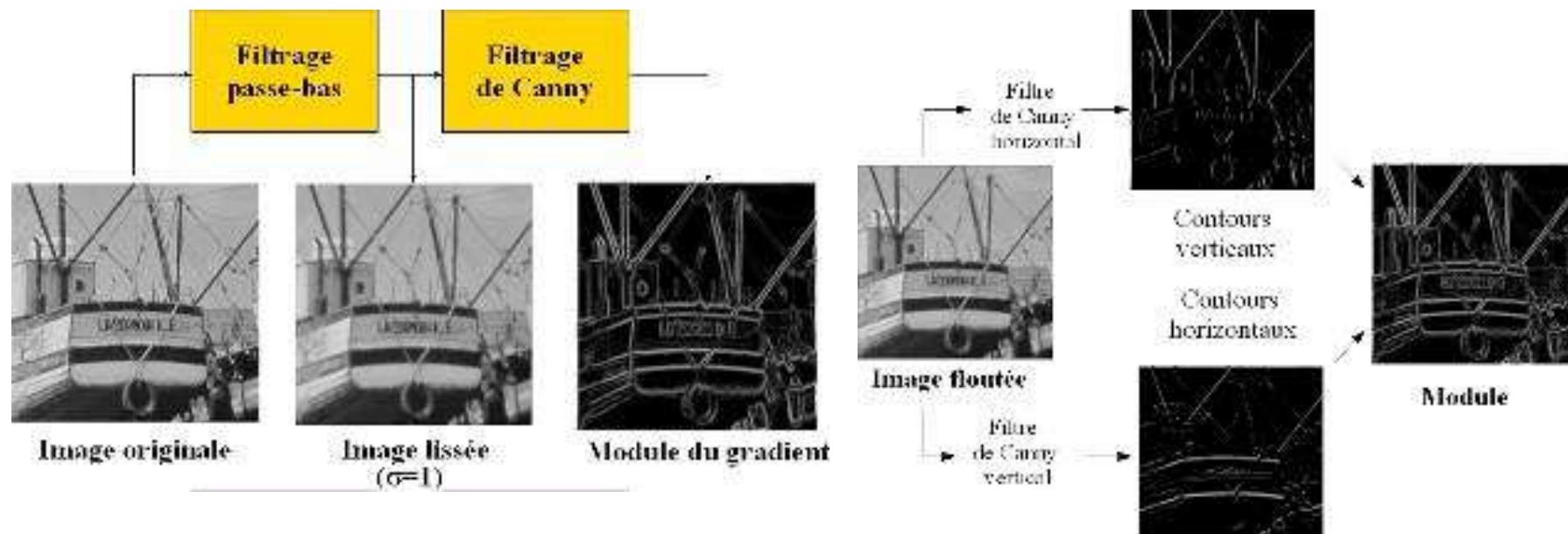
- Filtrage de Canny

- Paramètres
  - Taille du filtre
  - Seuils

Il n'existe pas actuellement de méthode générique pour déterminer des seuils produisant des résultats satisfaisants sur tous les types d'images.

# DÉTECTION DE CONTOURS [2]

## ○ Filtrage de Canny



# CANNY EDGE DETECTION –

EXTRAIT DE

HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT

- Faible taux d'erreur
- Les arêtes sont bien localisées
- Une seule réponse pour un seul contour
- Étapes



Ref: [http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html)

[http://en.wikipedia.org/wiki/Canny\\_edge\\_detector](http://en.wikipedia.org/wiki/Canny_edge_detector)

# CANNY EDGE DETECTION –

EXTRAIT DE

HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT

- Étape 1: Lisser pour débruiter

$\frac{1}{115}$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

$\frac{1}{159}$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Masque Gaussian

Ref:

[http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html)

154

# CANNY EDGE DETECTION –

EXTRAIT DE

HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT

- Étape 1: Lisser pour débruiter
- Étape 2: operator de Sobel

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Ref:

[http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html)

# CANNY EDGE DETECTION –

EXTRAIT DE

HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT

- Étape 1: Lisser pour débruiter
- Étape 2: operator de Sobel
- Étape 3: Amplitude et orientation du gradient

$$g \square \sqrt{g_x^2 + g_y^2}$$
$$\theta \square \arctan\left(\frac{g_y}{g_x}\right)$$

Ref:

[http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html)

# CANNY EDGE DETECTION –

EXTRAIT DE

HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT

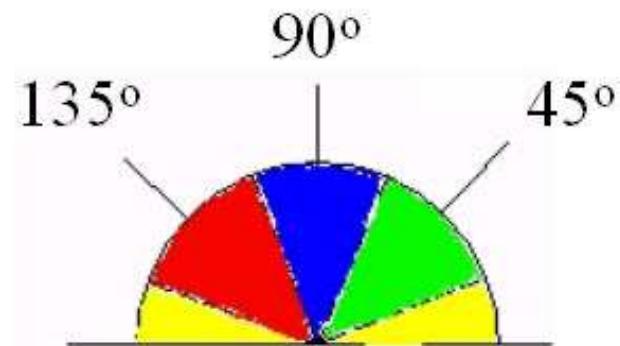
- Étape 1: Lisser pour débruiter
- Étape 2: operator de Sobel
- Étape 3: Amplitude et orientation du gradient
- Étape 4: Quantification de la direction
  - Because there are only four directions when describing the surrounding pixels

Yellow: 0 to 22.5 & 157.5 to 180 degrees

Green: 22.5 to 67.5 degrees

Blue: 67.5 to 112.5 degrees

Red: 112.5 to 157.5 degrees



Ref:

[http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html)

# CANNY EDGE DETECTION –

EXTRAIT DE

HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT

- Étape 1: Lisser pour débruiter
- Étape 2: operator de Sobel
- Étape 3: Amplitude et orientation du gradient
- Étape 4: Quantification de la direction
- Étape 5: Suppression des non maximaux
  - Tracer le long du contour dans la direction du bord et supprimer toute valeur de pixel (la mettre à 0) qui n'est pas considérée comme un bord.
  - Cela donnera une ligne fine dans l'image de sortie.

Ref:

[http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html)

# CANNY EDGE DETECTION –

EXTRAIT DE

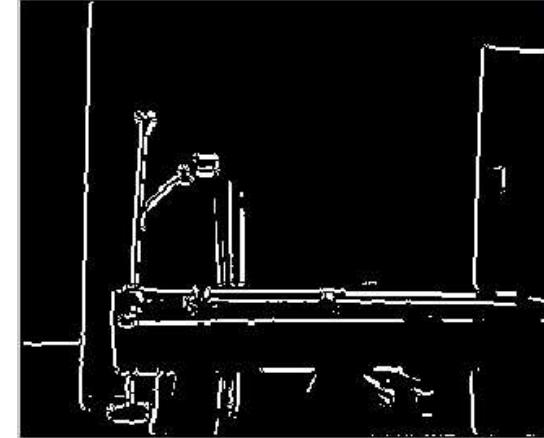
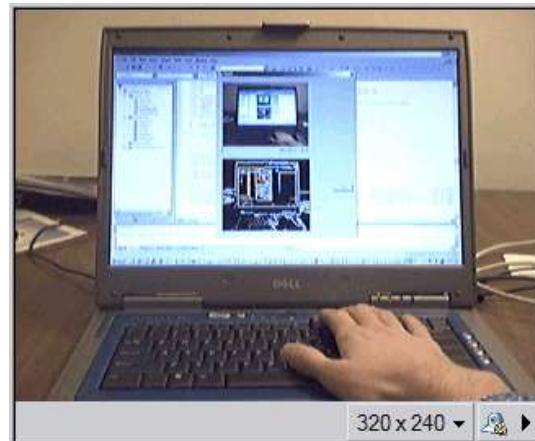
HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT

- Étape 1: Lisser pour débruiter
- Étape 2: operator de Sobel
- Étape 3: Amplitude et orientation du gradient
- Étape 4: Quantification de la direction
- Étape 5: Suppression des non maximaux
- Étape 6: Hysteresis
  - Élimination des stries
  - Deux seuils  $T_1 > T_2$
  - Pixel avec  $g > T_1 \rightarrow$  marqué comme bord immédiatement
  - Pixel connecté à un pixel de bord et  $g > T_2 \rightarrow$  également marqué comme bord

# CANNY EDGE DETECTION –

EXTRAIT DE

HTTP://STAFF.USTC.EDU.CN/~XJCHEN99/TEACHING/IMAGEUNDERSTANDING/CHARTER4\_EDGES.PPT



# CANNY EDGE DETECTION

- Effet de  $\sigma$  



original

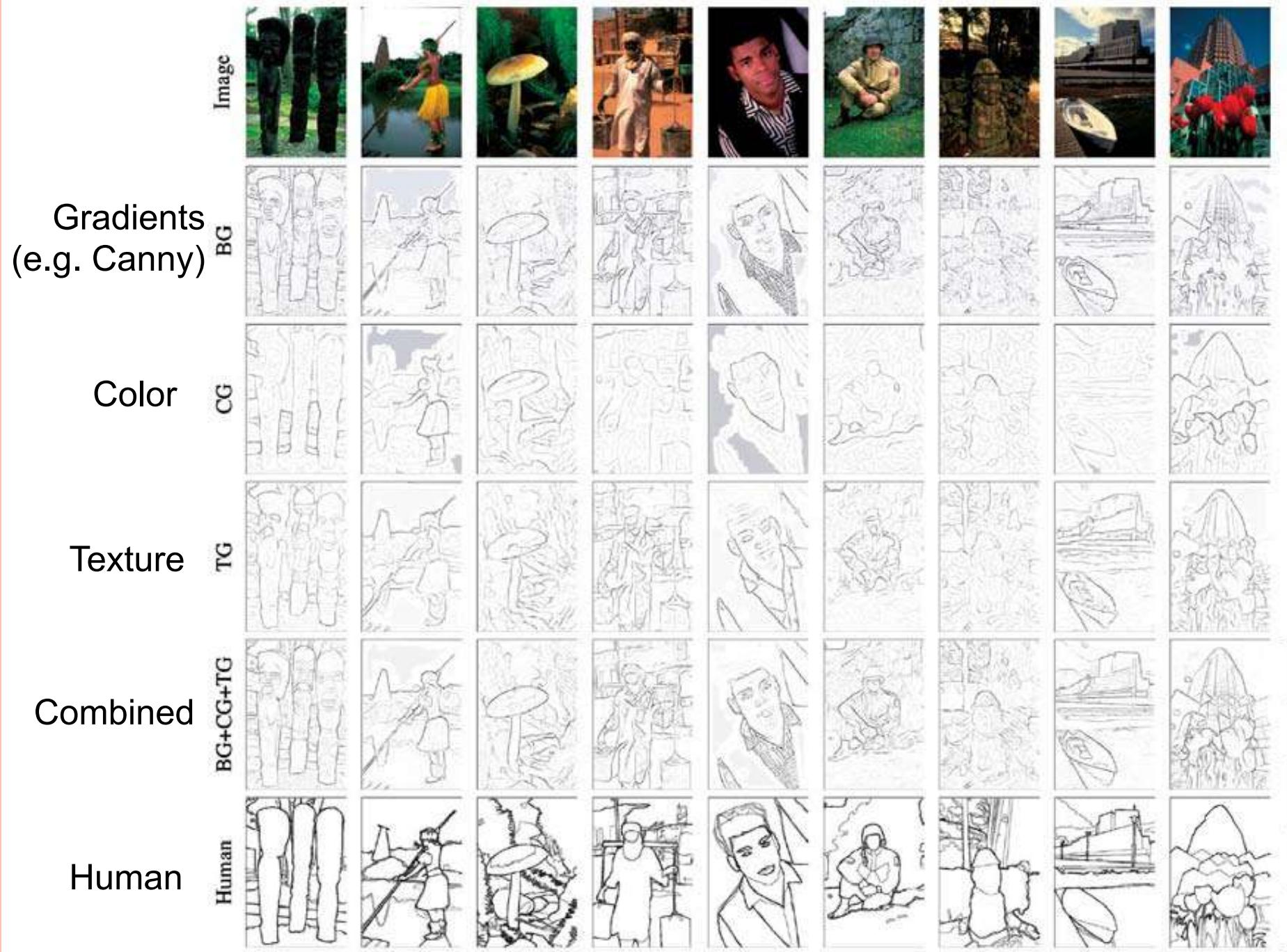


Canny with  $\sigma = 1$

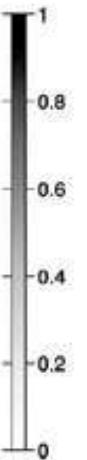


Canny with  $\sigma = 2$

- Grand  $\sigma$  pour détecter les contours les plus importants
- Petit  $\sigma$  pour détecter les caractéristiques plus fines



[Extrait de [5]]

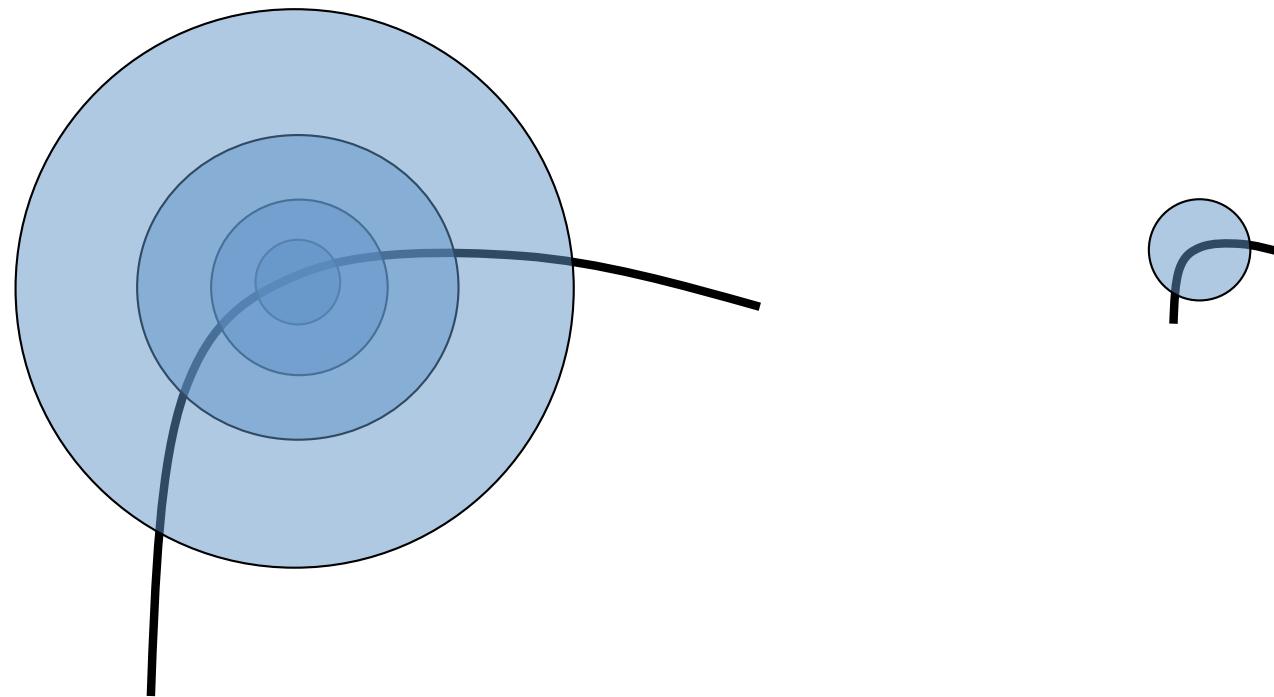


# PLAN

- Introduction
- Détection
  - de points
  - de contours
  - de régions
  - d'objets

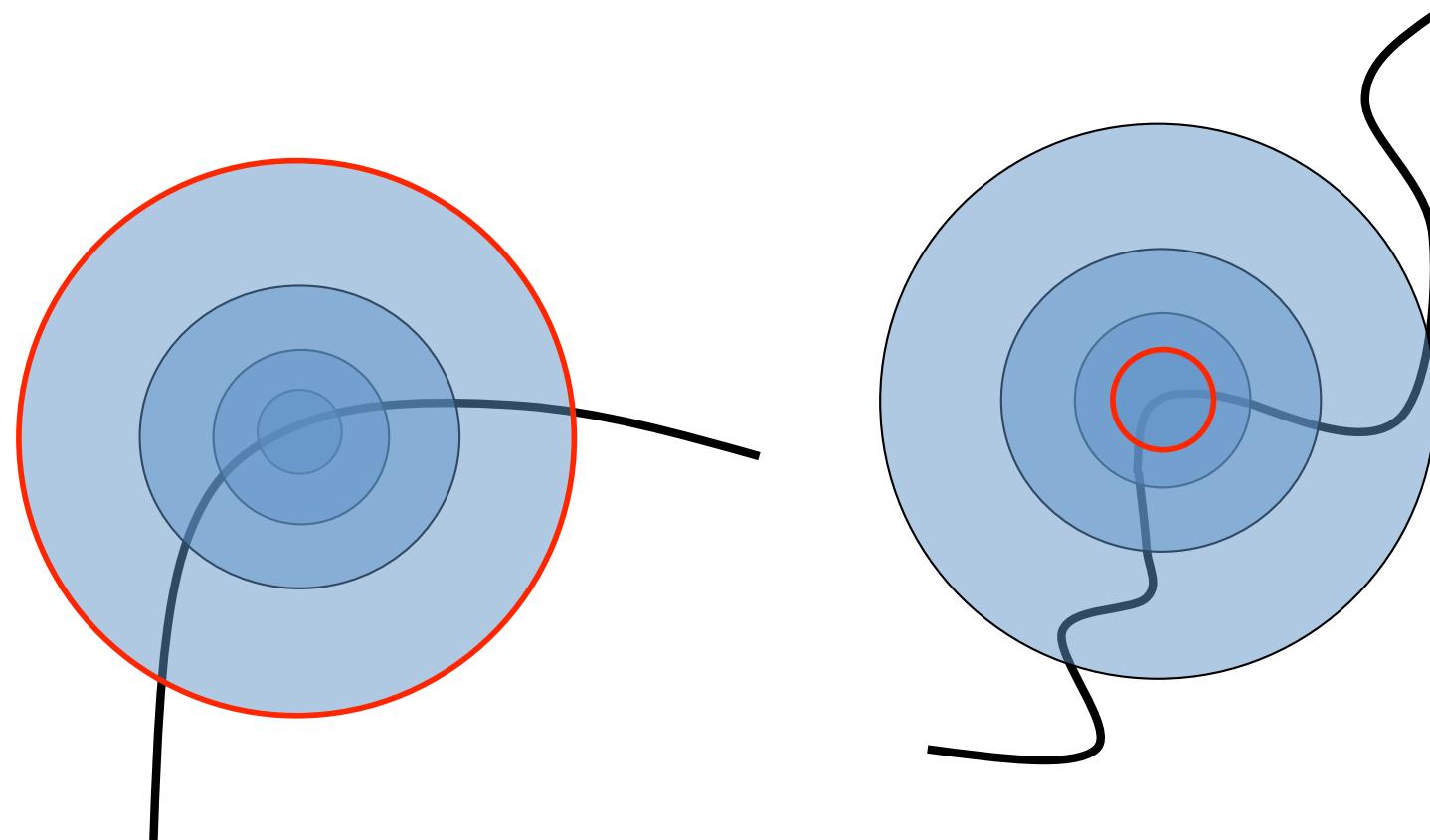
## DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

- Considérons des régions de tailles différentes autour d'un point
- Les régions de taille correspondantes vont avoir le même aspect entre les deux images



## DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

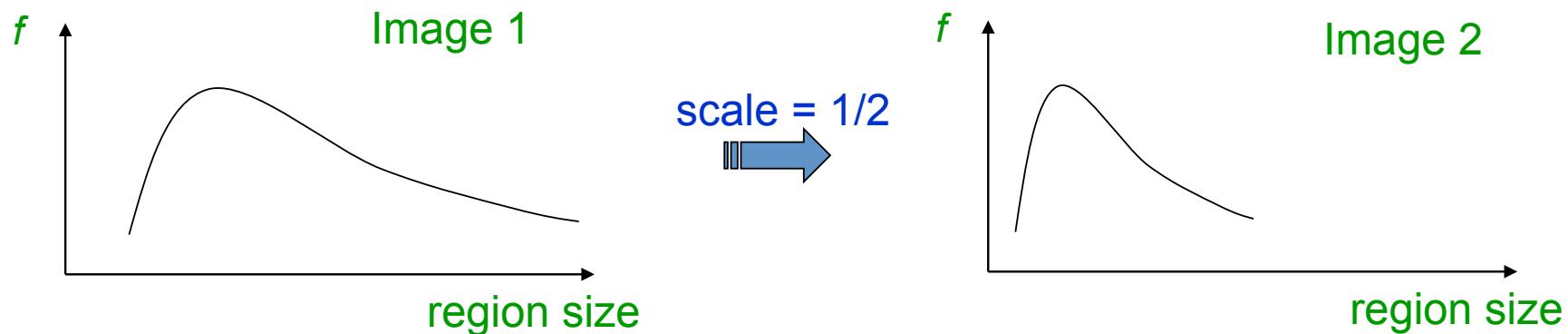
- Problème : comment choisir les cercles correspondant de manière indépendante entre les 2 images ?



# DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

- Solution :

- Faire une fonction de chaque région ( cercle) qui est invariant à l'échelle (la même pour chaque région, même si elles sont à des échelles différentes)
- Pour un point dans une image, nous pouvons le considérer comme une fonction de la taille de la région (rayon du cercle)



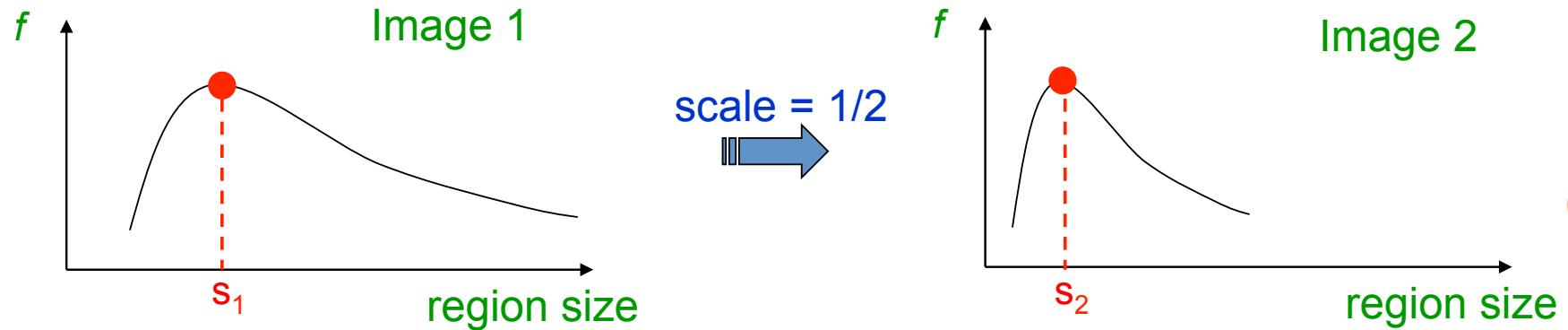
# DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

- Approche classique :

Prendre un maximum local de cette fonction

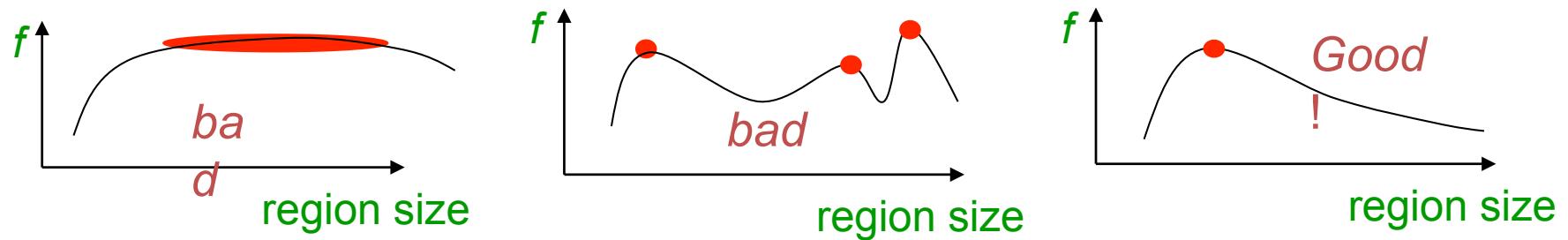
- Remarque : la taille de la région, pour laquelle le maximum est obtenu, doit être co-variant avec l'échelle de l'image

Important : cette taille de région invariante à l'échelle doit être trouvée indépendamment sur chaque image



## DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

- Une bonne fonction de détection à travers les échelles à un seul pic



- Pour les images classiques, une bonne fonction pourra simplement répondre aux contrastes

# DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

- Fonction d'échelle

$$f = \text{Kernel} * \text{Image}$$

Kernels:

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

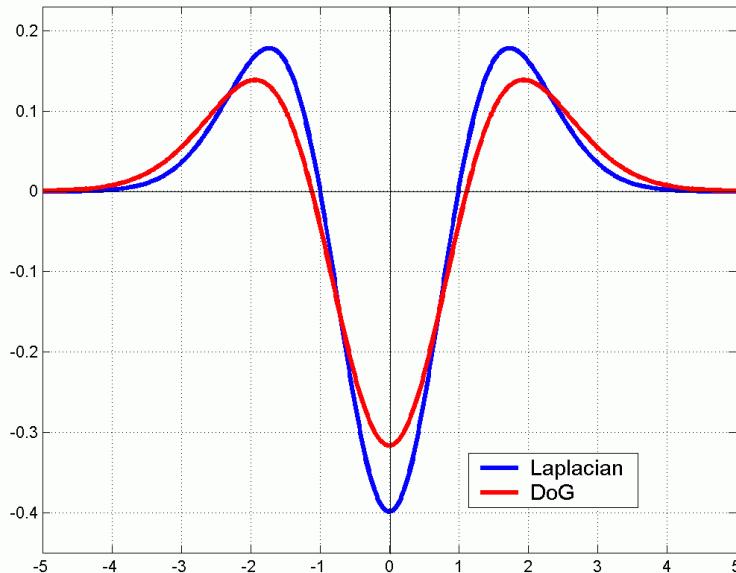
(Laplacian)

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

(Difference of Gaussians)

Gaussienne :

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



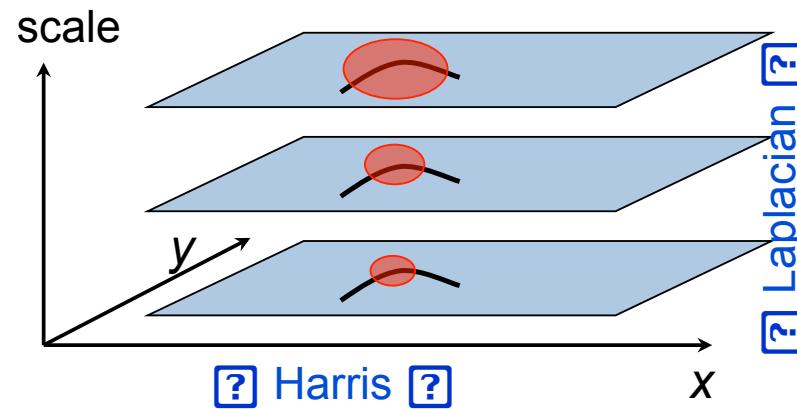
Remarque : les noyaux sont invariants à l'échelle et à la rotation

# DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

- **Harris-Laplacian**<sup>1</sup>

*Find local maximum of:*

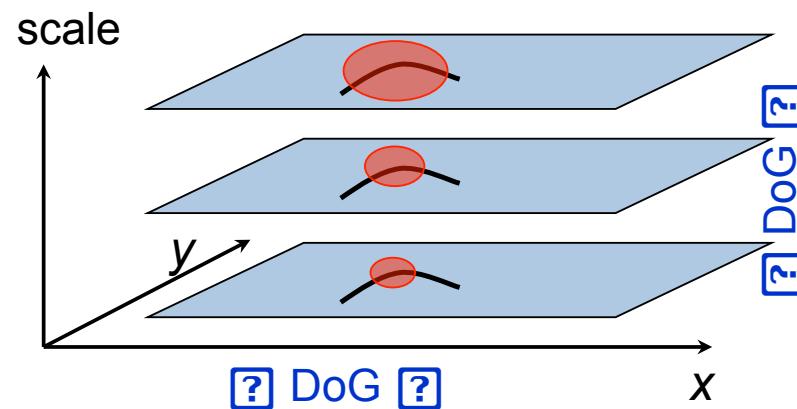
- Harris corner detector in space (image coordinates)
- Laplacian in scale



- **SIFT (Lowe)**<sup>2</sup>

*Find local maximum of:*

- Difference of Gaussians in space and scale



<sup>1</sup> K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

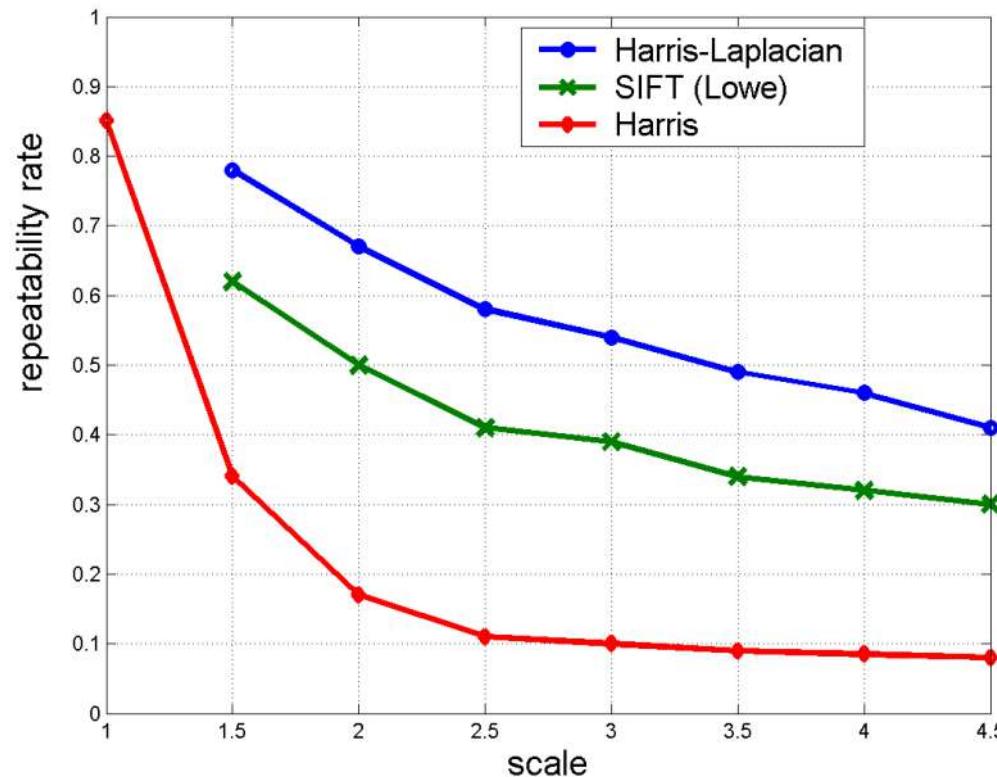
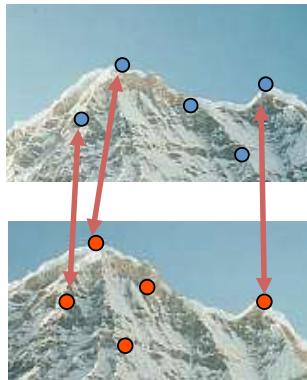
<sup>2</sup> D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

# DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\# \text{ correspondences}}{\# \text{ possible correspondences}}$$



K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

# DÉTECTION DE RÉGIONS INVARIANTES À L'ÉCHELLE [5]

## ○ Résumé :

- Entrées : 2 images de la même scène avec une grande différence d'échelles
- But : trouver le même point d'intérêt de manière indépendante sur chaque image
- Solution : chercher le max. d'une fonction appropriée en échelle et dans l'espace (sur l'image)

Methods:

1. **Harris-Laplacian** [Mikolajczyk, Schmid]: maximize Laplacian over scale, Harris' measure of corner response over the image
2. **SIFT** [Lowe]: maximize Difference of Gaussians over scale and space

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- « transformation de caractéristiques visuelles invariante à l'échelle »
  - Identifie les éléments similaires entre des images
  - David G. Lowe, « Object recognition from local scale-invariant features », dans *Proceedings of the International Conference on Computer Vision*, vol. 2, 1999 p. 1150–1157
  - Calcul et mise en correspondance de descripteurs

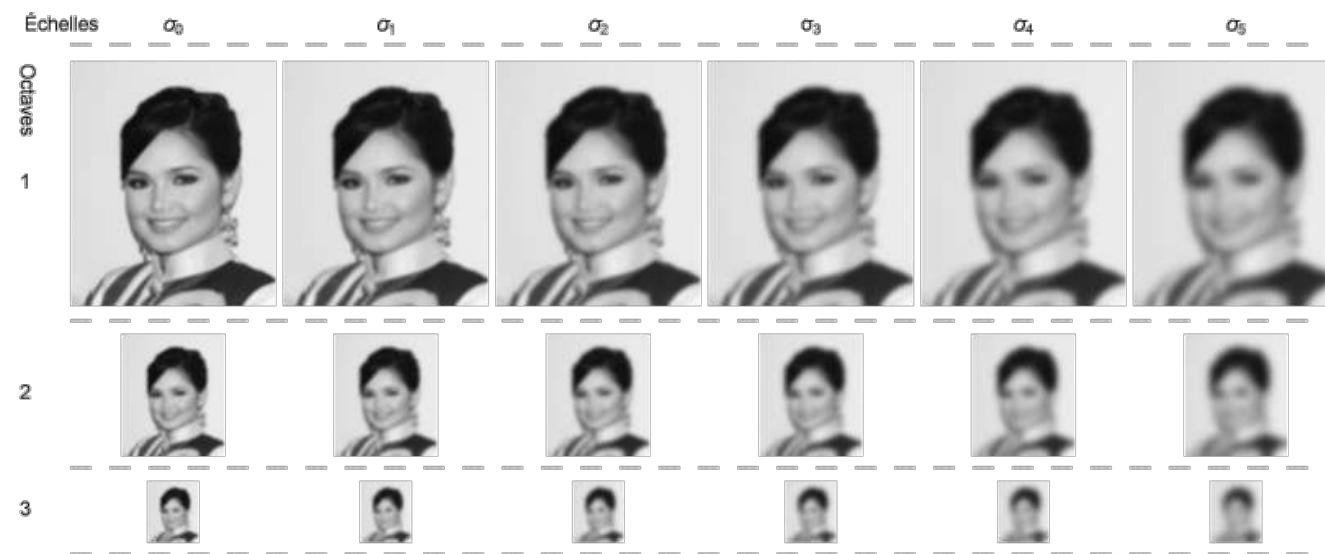
# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Deux parties :
  - un algorithme de détection de caractéristiques et de calcul de descripteurs ;
  - un algorithme de mise en correspondance proprement dit.
- Descripteurs SIFT :
  - Point d'intérêt avec un rayon associé (facteur d'échelle)
  - Détermination d'une orientation intrinsèque pour construire une histogramme des orientations locales des contours
  - Cet histogramme sous la forme d'un vecteur à 128 dimensions constitue le descripteur SIFT du point d'intérêt

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Détection des points d'intérêt
  - définis par ses coordonnées sur l'image (x et y) et par son facteur d'échelle caractéristique ( $\sigma$ )
  - Détection dans l'espace des échelles

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$



175

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Détection des points d'intérêt
  - Différence de Gaussiennes

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

k est un paramètre fixe de l'algorithme qui dépend de la finesse de la discrétisation de l'espace des échelles voulue

Valeurs typiques :  $\sigma = 1.6$ ;  $k = \sqrt{2}$

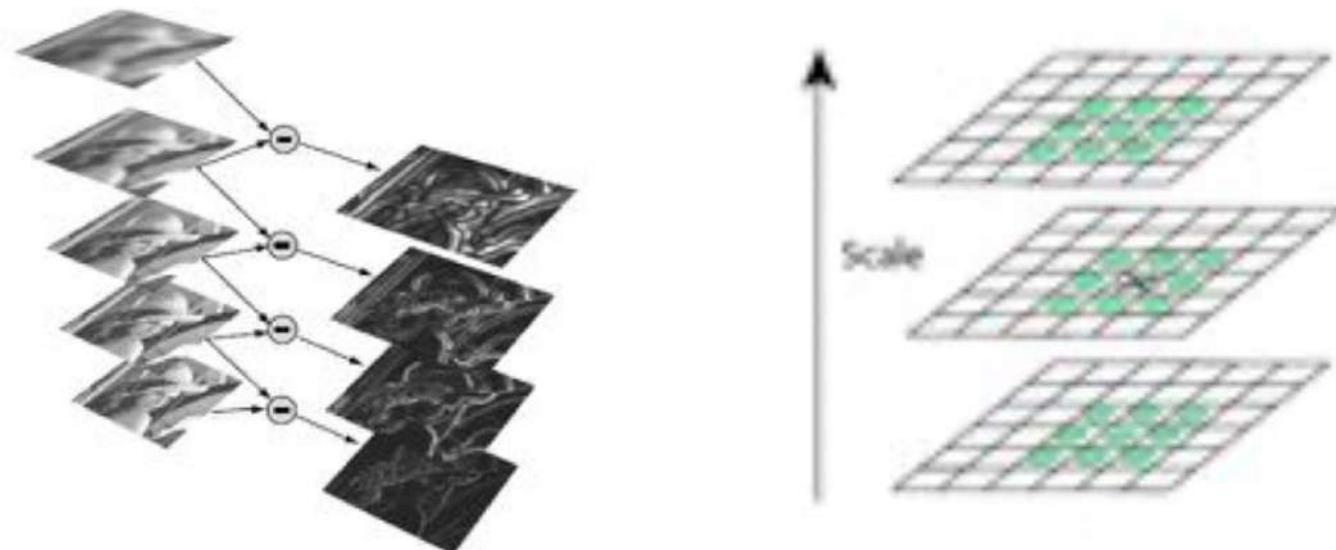
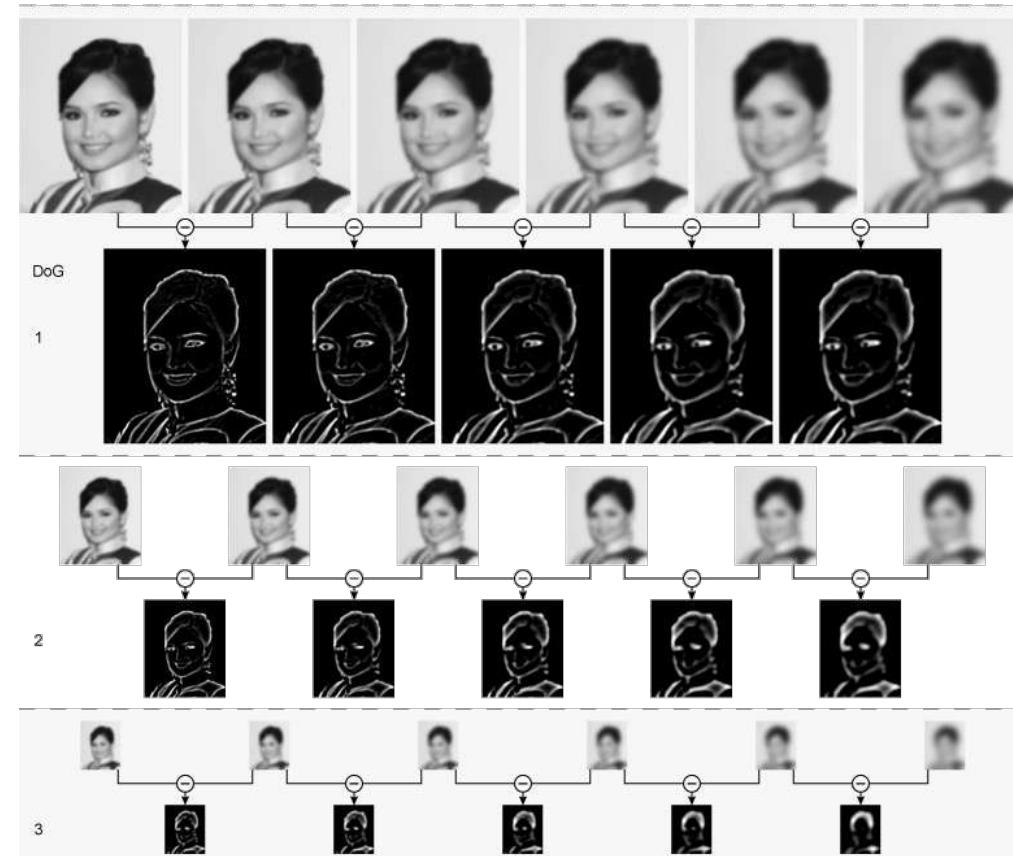


Figure 9.1: A Difference-of-Gaussian octave. The five images in the left stack are incrementally smoothed versions of the input image. The right stack shows the resulting DoG.

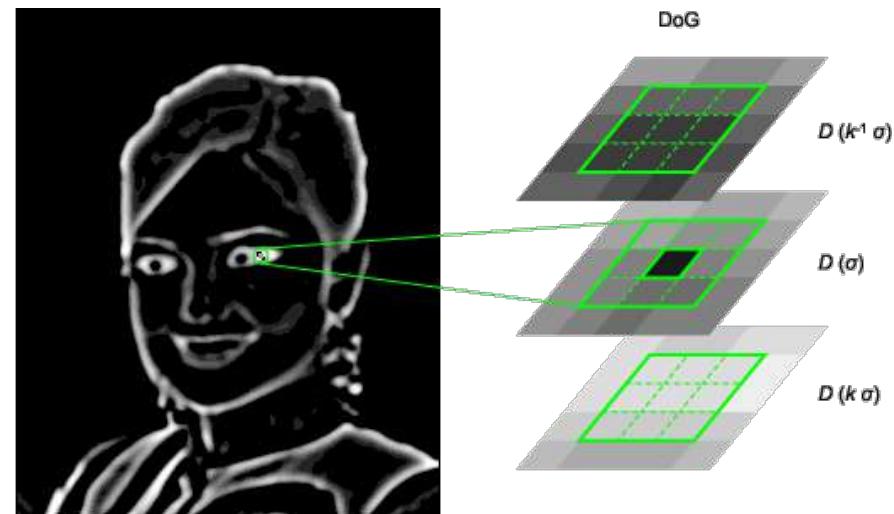
# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Détection des points d'intérêt
  - Construction de la pyramide de différences de gaussiens (DoG) à partir de la pyramide de gradients.



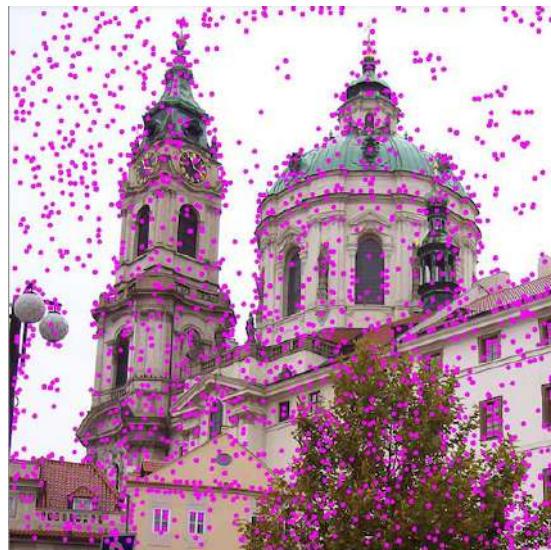
# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Détection des points d'intérêt
  - Exemple de détection d'extremums dans l'espace des échelles.

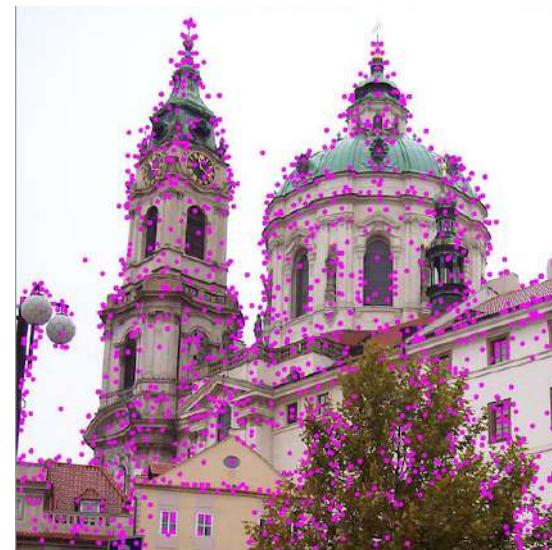


# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Localisation des points d'intérêt



a



b



c

Après la détection des extrema dans l'espace des échelles (a),  
l'algorithme élimine les points de faible contraste (b), puis les points situés  
sur les arêtes (c)

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

## ○ Assignation d'orientation

- attribution à chaque point-clé une ou plusieurs orientations déterminées localement sur l'image à partir de la direction des gradients dans un voisinage autour du point
- Calcul de la norme et de l'orientation du gradient

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)$$

$\forall (x, y)$  dans un voisinage de  $(x_0, y_0)$ .

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Assignation d'orientation
  - Construction de l'histogramme des orientations

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

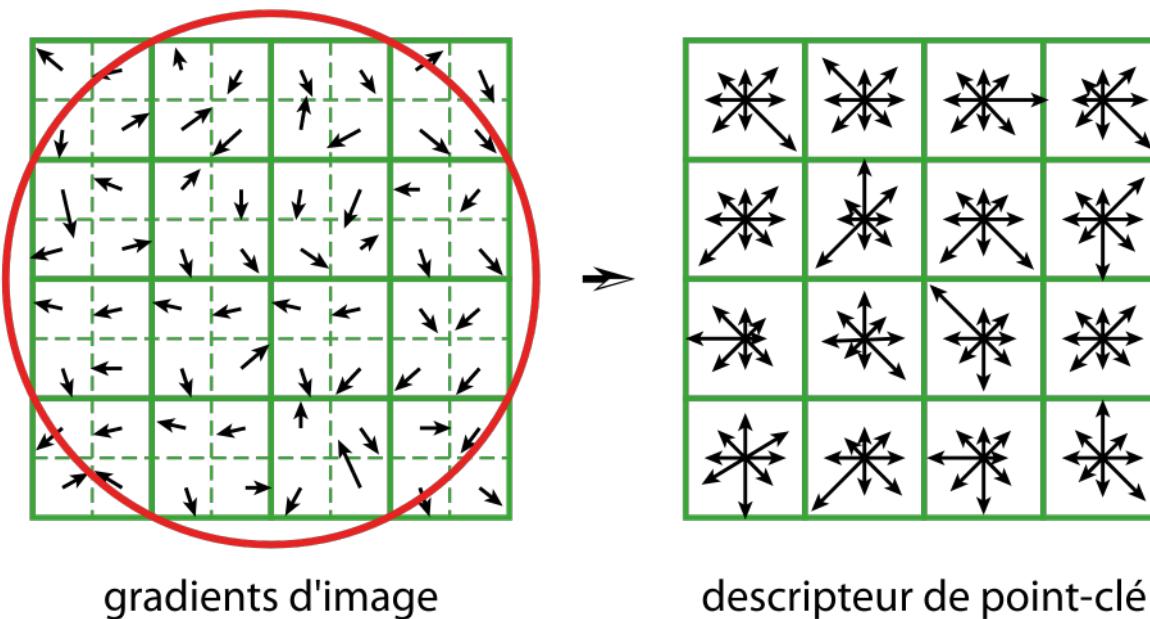
$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)$$

$\forall (x, y)$  dans un voisinage de  $(x_0, y_0)$ .

À l'issue de cette étape, un point caractéristique est défini par quatre paramètres  $(x, y, \sigma, \theta)$

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

- Descripteur de points caractéristiques
  - Construction d'un descripteur SIFT



# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

## ○ Appariement des points caractéristiques

- Algorithme du plus proche voisin basé sur la distance L2
- Comment écarter les mauvais appariements?  
Seuil sur L2 => mauvaise performance  
Solution : Seuil sur le rapport

**meilleur appariement / second meilleur appariement**

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

Reconnaissance malgré des occlusions



# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

## Test de robustesse à l'illumination

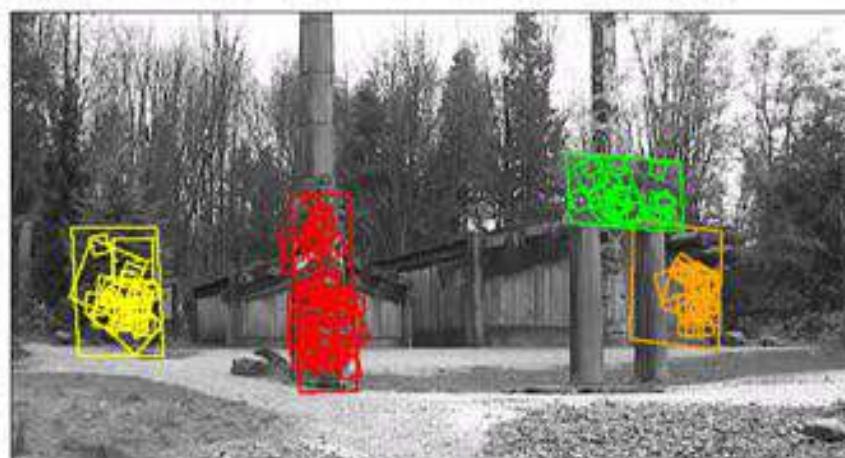
- Même **image** sous différents éclairages



273 clés vérifiées dans l'appariement final

# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

Reconnaissance en localisation



# SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

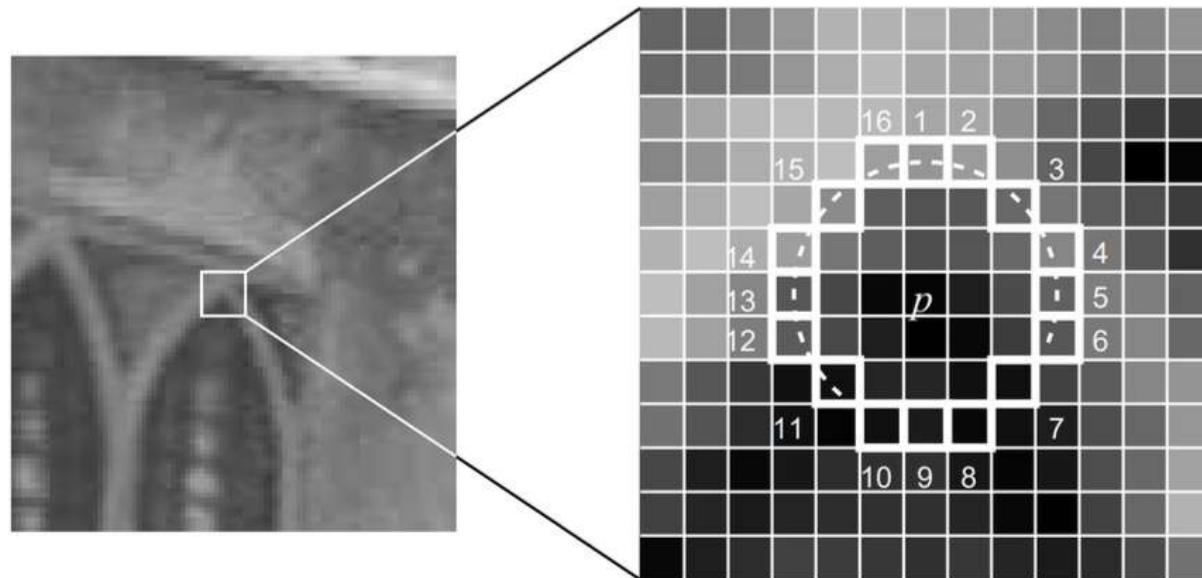
Image panoramique



[Brown & Lowe 2003]

# FEATURES FROM ACCELERATED SEGMENT TEST (FAST)

- Détection de point par algorithme FAST



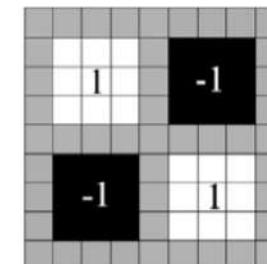
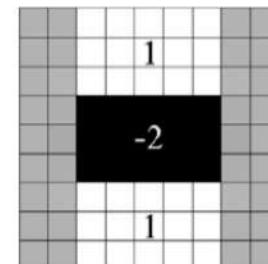
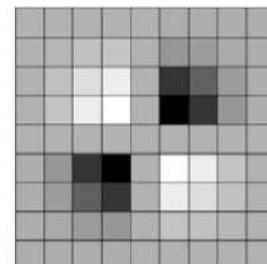
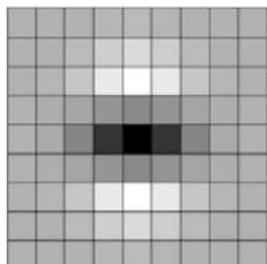
# SPEEDED UP ROBUST FEATURES (SURF)

- Descripteur basé sur des histogrammes en 2 parties :
  - extraction de caractéristiques
  - description de caractéristiques
- Extraction basée sur le calcul approximatif de la matrice hessienne d'un pixel p à l'échelle  $\sigma$

$$\mathcal{H}(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix}$$

Où L est la dérivée seconde

- Les auteurs considèrent des filtres 9x9 and  $\sigma = 1.2$  pour approximer les dérivées gaussiennes d'ordre 2



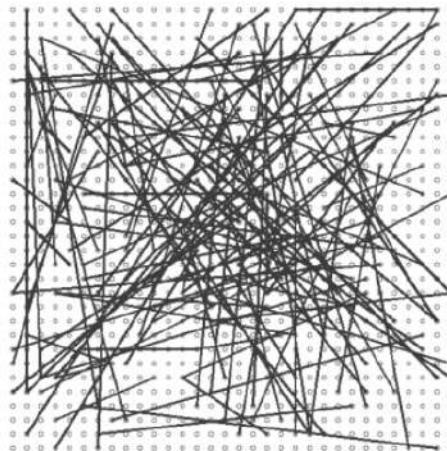
# SPEEDED UP ROBUST FEATURES (SURF)

- En nommant les approximation comme  $D_{xx}$ ,  $D_{yy}$  et  $D_{xy}$ , les auteurs obtiennent les déterminant Hessian en calculant

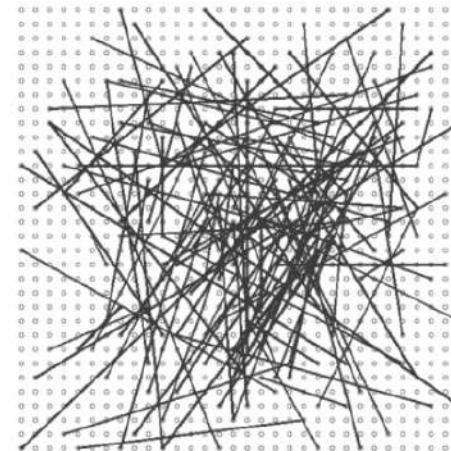
$$\det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2$$

- Localisation des points d'intérêt à partir d'une approche sur pyramide d'image où un filtre gaussien est appliqué répétitivement et est sous échantillonné au niveau suivant de la pyramide
- Différents niveau de masques sont appliqués sur l'image, permettant la localisation des point d'intérêts
- Puis création de descripteur pour chaque point d'intérêt

# BINARY ROBUST INDEPENDENT ELEMENTARY FEATURES (BRIEF)



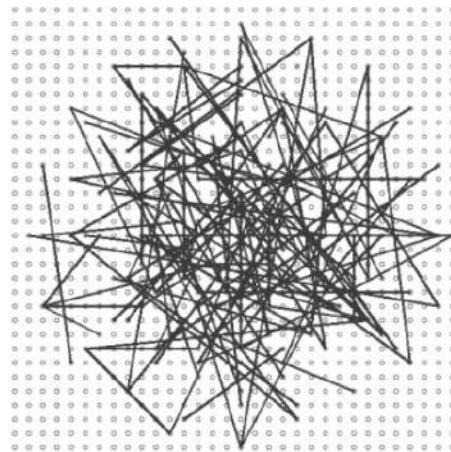
G I



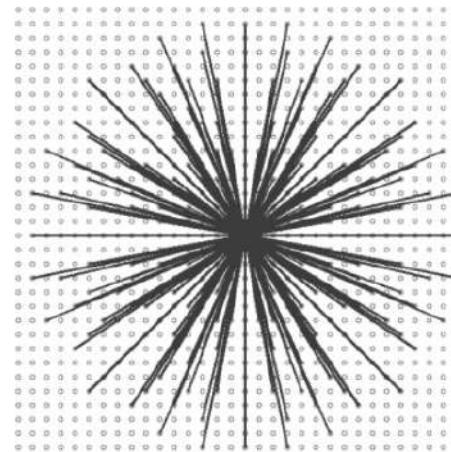
G II



G III



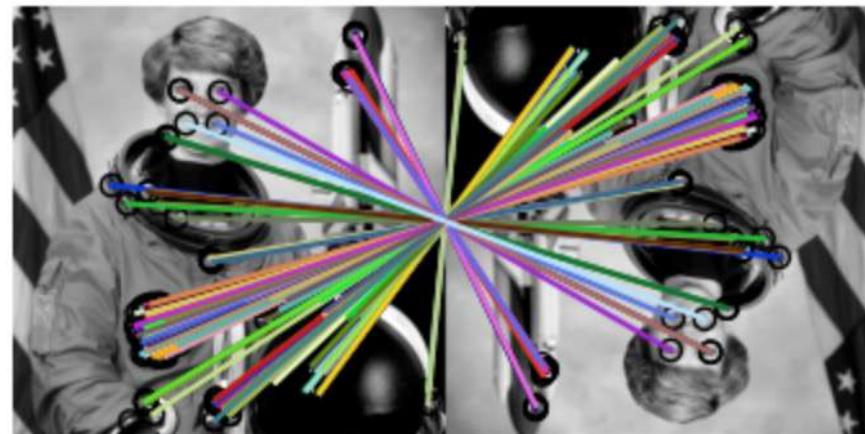
G IV



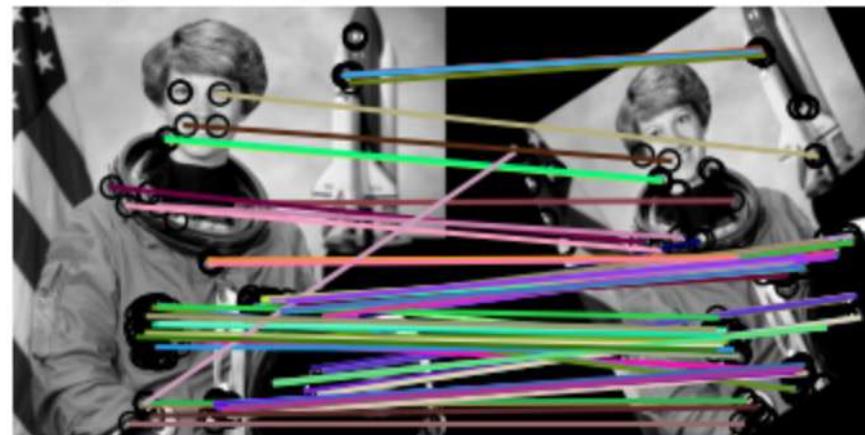
G V

# ORIENTED FAST AND ROTATED BRIEF (ORB)

Original Image vs. Transformed Image



Original Image vs. Transformed Image



# PLAN

- Introduction
- Détection
  - de points
  - de contours
  - de régions
  - d'objets

# RECONNAISSANCE D'OBJETS [9]

- Il existe différents types de reconnaissance d'objets :
  - La classification d'images consiste à prédire la classe d'un objet dans une image.
    - Entrée : une image avec un seul objet.
    - Sortie : une étiquette de classe.



CHAT

## RECONNAISSANCE D'OBJETS [9]

- Il existe différents types de reconnaissance d'objets :
  - La localisation d'objets consiste à identifier l'emplacement d'un ou plusieurs objets dans une image et à dessiner un cadre de délimitation autour de leur étendue.
    - Entrée : une image avec un ou plusieurs objets.
    - Sortie : un ou plusieurs cadres de délimitation.



## DÉTECTION D'OBJETS [9]

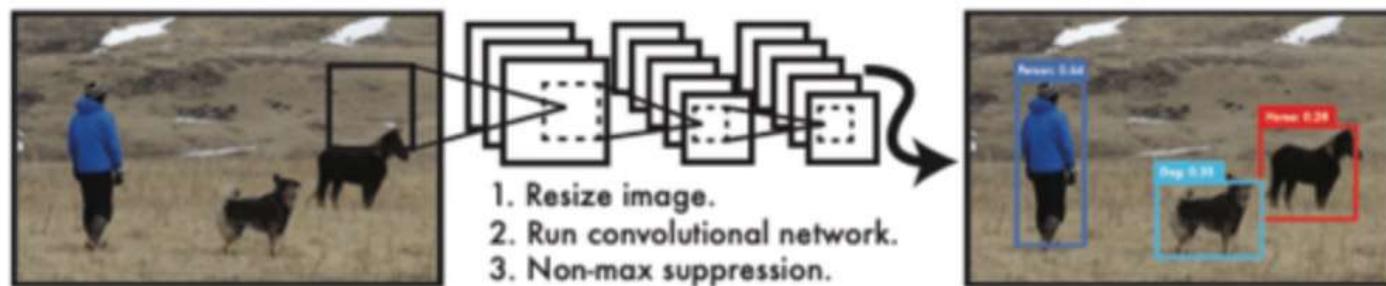
- La détection d'objets combine ces deux tâches et dessine un cadre de délimitation autour de chaque objet dans l'image et leur attribue une classe.
  - Entrée : une image avec un ou plusieurs objets.
  - Sortie : un ou plusieurs cadres de délimitation et une étiquette de classe pour chaque cadre de délimitation



**CHAT**

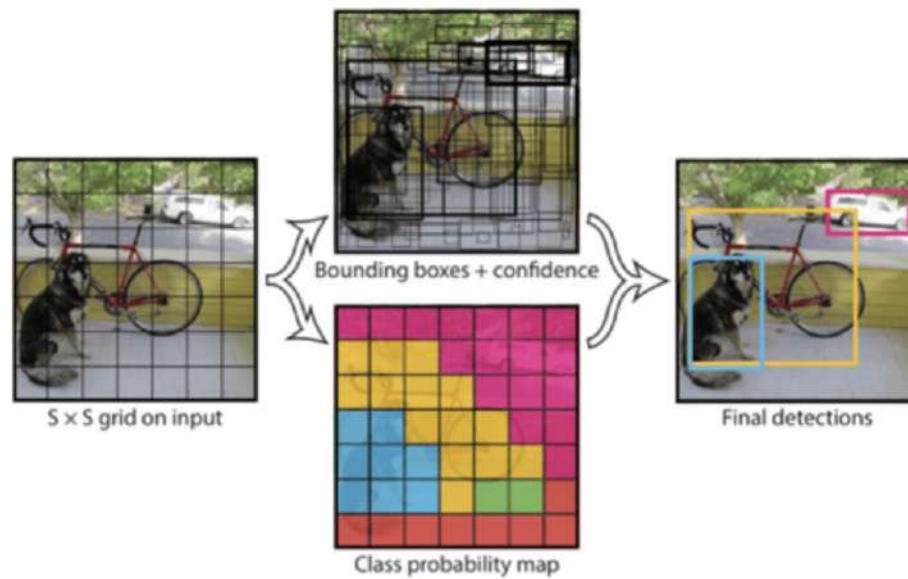
# YOLO - YOU ONLY LOOK ONCE

- “You Only Look Once: Unified, Real-Time Object Detection.” (<https://arxiv.org/abs/1506.02640>)
  - détection d’objets en temps réel (30 fps)



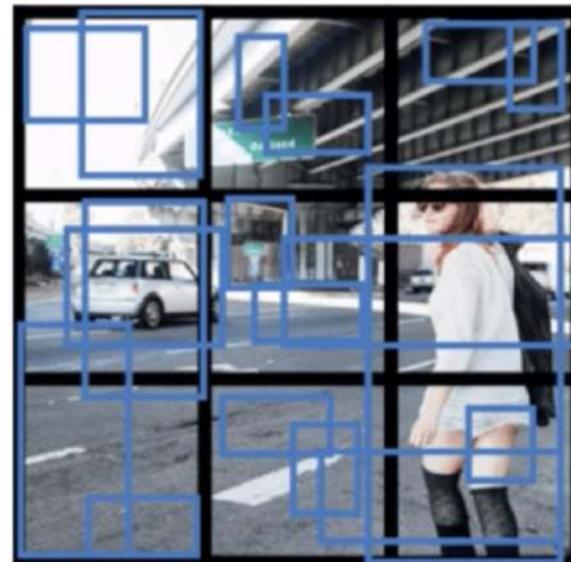
# YOLO – FONCTIONNEMENT [9]

- Entrée : Photo
- Sortie : bounding box et labels de classes
- Division de l'image d'entrée en une grille de cellules - chaque cellule est responsable de la prédiction d'une bounding box. Et d'un label de classe



# YOLO – DÉTAILS [9]

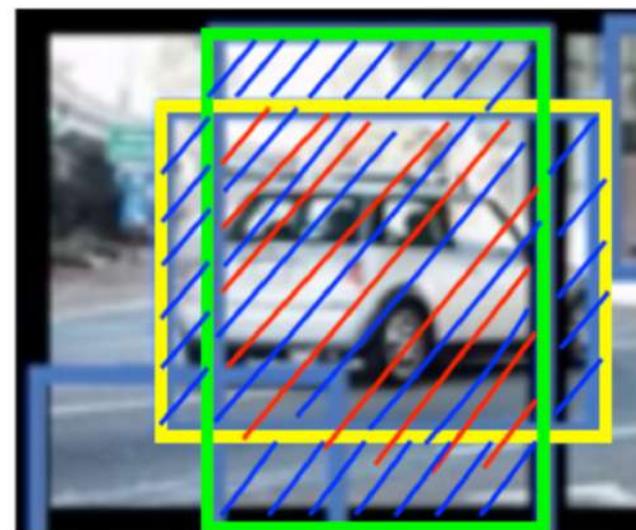
- Application d'une grille sur l'image pour analyser séparément chaque portion
- Création d'**anchor box**
- Suppression de toutes les *anchor boxes* qui ont une faible probabilité qu'un objet soit détecté
- Suppression non max



# YOLO – DÉTAILS [9]

- Suppression non max :
  1. Suppression des anchor boxes ayant une probabilité inférieure à un certain seuil (0.6 par exemple)
  2. Sélection de l'anchor box avec la probabilité de détection la plus élevée;
  3. Suppression des anchor boxes qui intersectent l'anchor box sélectionnée en 2) ayant un IoU supérieur à un certain seuil (0.5 généralement)
  4. on répète 2) et 3).

L'IoU est calculé en divisant l'intersection des 2 anchor boxes par leur union (diviser la partie rouge par la partie bleue).



## YOLO – DÉTAILS [9]

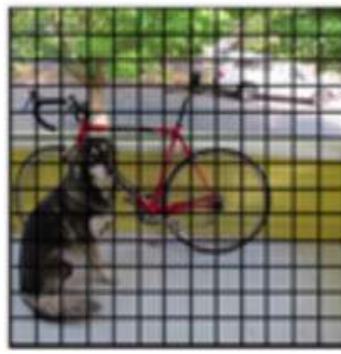


Image1



Image2

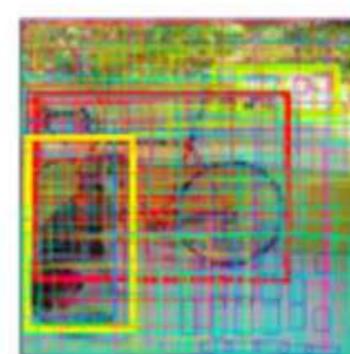


Image3

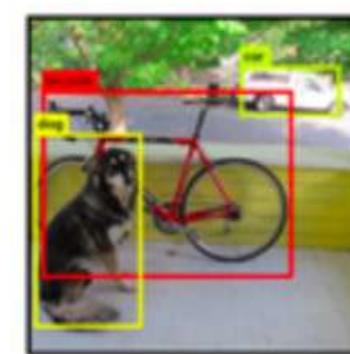
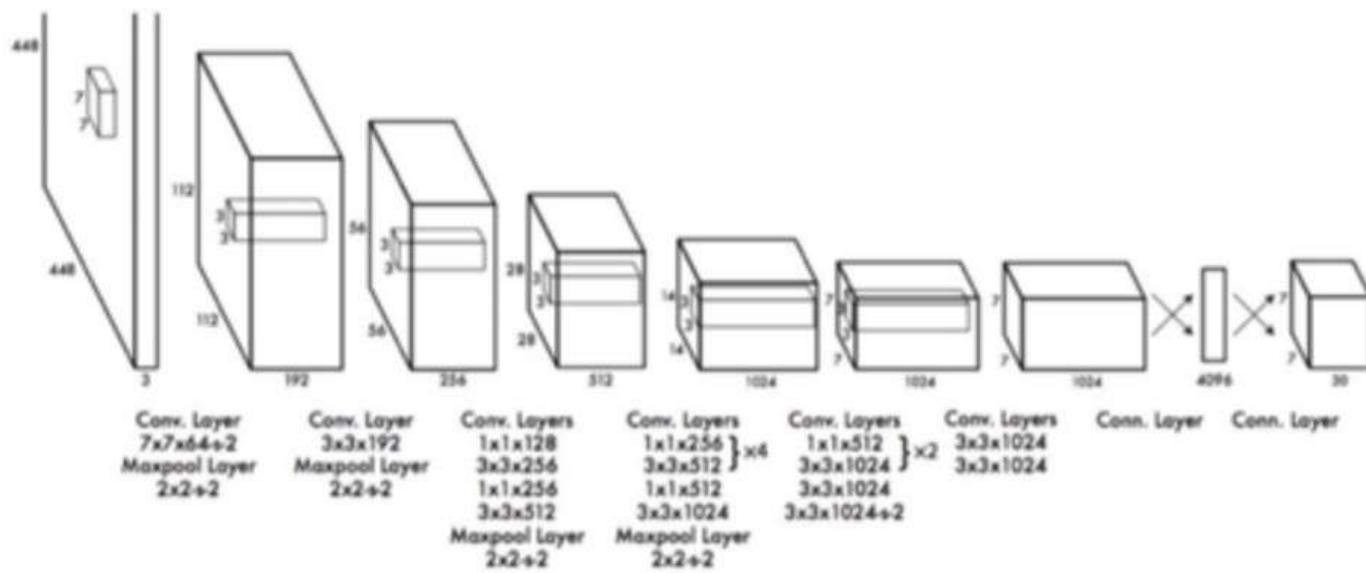


Image4

# YOLO – ARCHITECTURE [9]

- 24 couches convolutives suivies de 2 couches entièrement connectées
  - Les 20 premières couches convolutives sont pré-entraînées sur l'ensemble de données de classification ImageNet 1000-class
  - Les 4 dernières couches convolutives suivies de 2 couches entièrement connectées sont ajoutées pour entraîner le réseau à la détection d'objets avec une base de données spécifique (avec images annotées)



# BIBLIOGRAPHIE

1. Corner Detector, Donovan Parks et Jean-Philippe Gravel
2. Analyse d'images, Détection de contours, Jean-Hugh Thomas, 2007
3. Features, Digital Visual Effects, Yung-Yu Chuang, Université de Hong-kong, 2010.
4. Stanford University - CS231A Computer Vision: from 3D reconstruction to recognition
5. Stanford University - CS 131 Computer Vision: Foundations and Applications
6. Cours Vision par ordinateur, Elise Arnaud - Imag,
7. Cours Image Numérique, Aurélie Bugeau – Enseirb
8. Cours Image, Alain Boucher – IFI
9. You Only Look Once - un réseau de neurones pour la détection d'objets - <https://blog.octo.com/>