IMN259 - Analyse d'images

Chapitre 5

Extraction de caractéristiques

notes de cours de
Marie-Flavie Auclair-Fortier & Pierre-Marc Jodoin

Plan

1. Calcul des dérivées d'une image
2. Détection des contours
3. Coins
4. Segmentation

©Marie-Flavie Auclair-Fortier

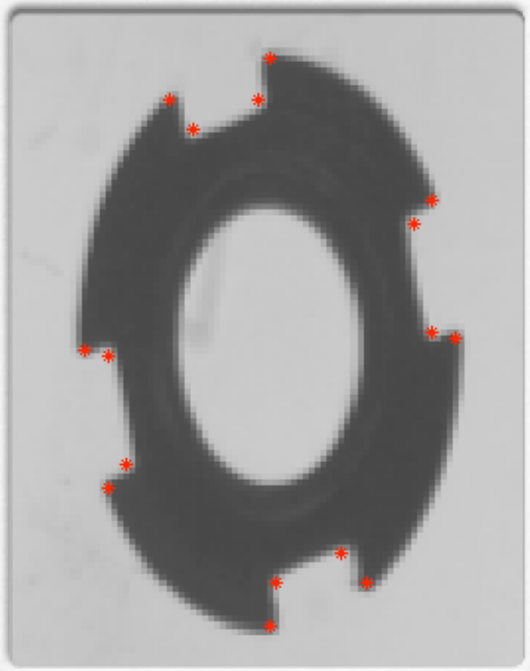
Préambule

- Extraction de différentes caractéristiques

Image originale

Contours

Caractéristiques ponctuelles



Régions

Source :

IMN259 Analyse d'images
©Marie-Flavie Auclair-Fortier

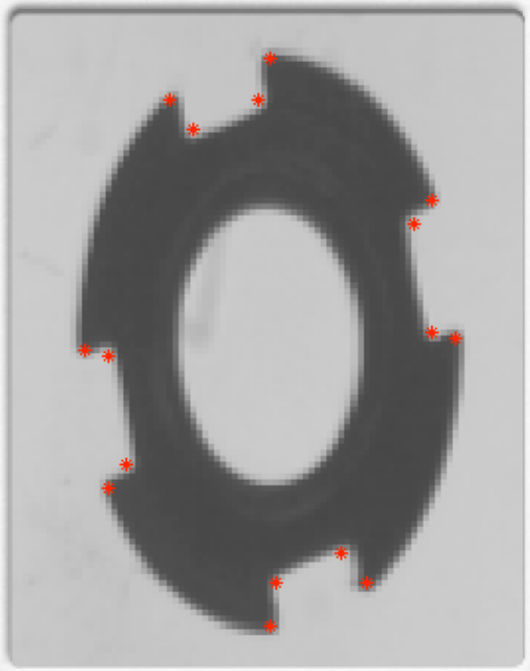
Préambule

- Extraction de différentes caractéristiques

Image originale

Contours

Caractéristiques ponctuelles



Dual

Régions

Source :

IMN259 Analyse d'images
©Marie-Flavie Auclair-Fortier

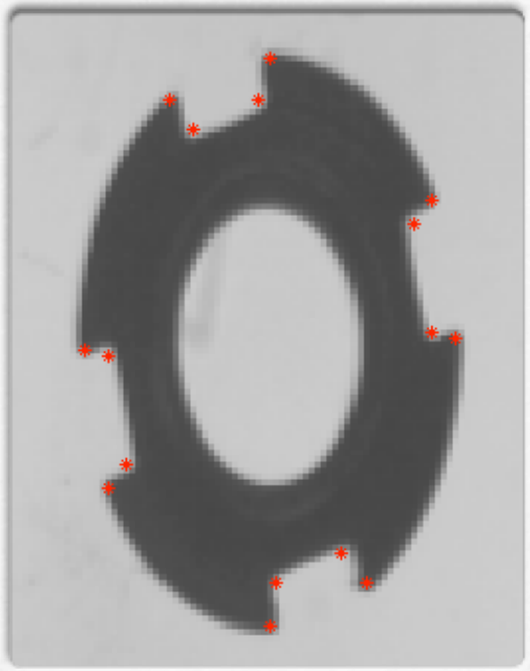
Préambule

- Extraction de différentes caractéristiques

Image originale

Contours

Caractéristiques ponctuelles



Intersections

Dual

Régions

Source :

IMN259 Analyse d'images
©Marie-Flavie Auclair-Fortier

Préambule

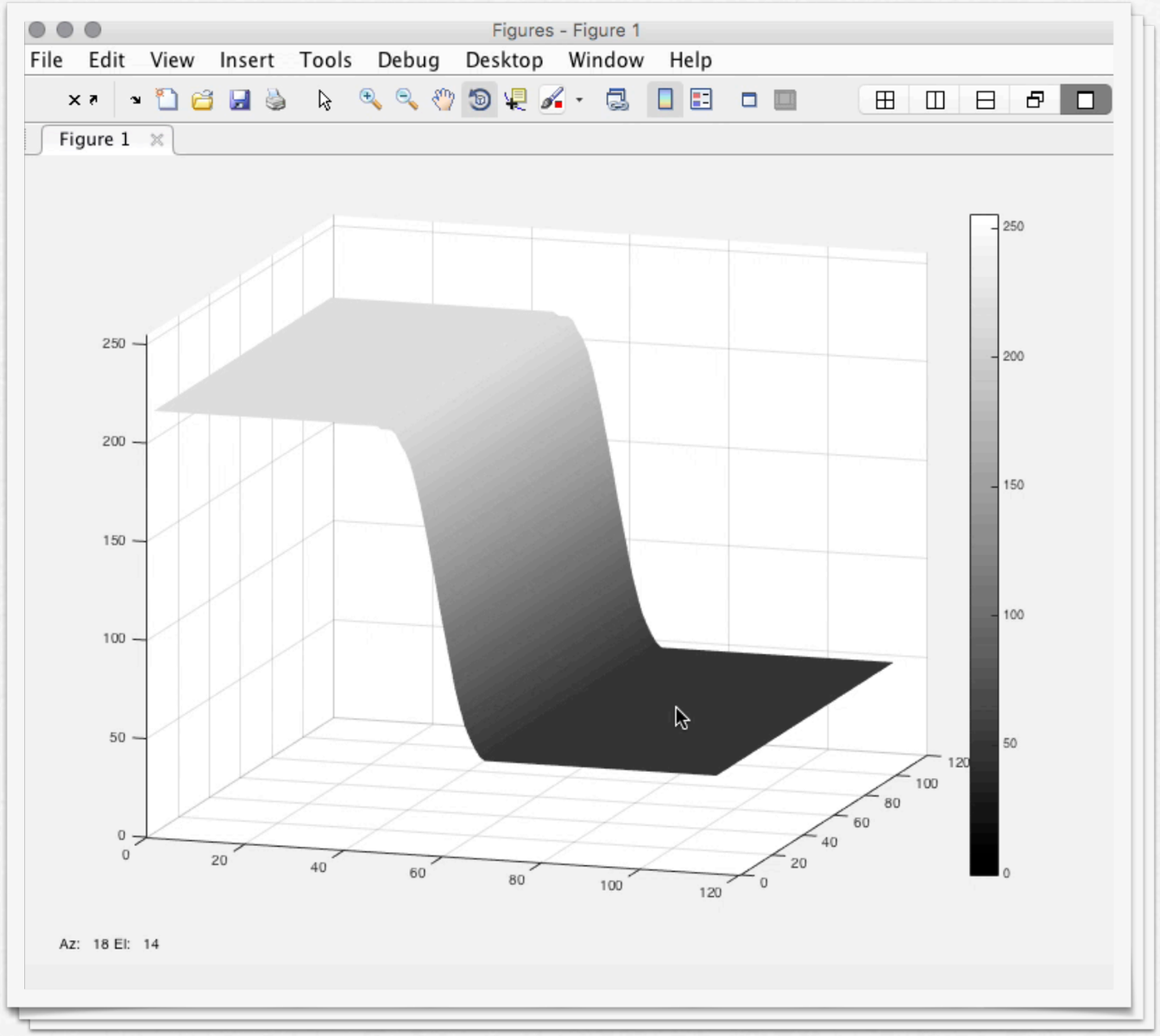
- Une image

Hypothèse : Le contour représente la frontière d'un objet

Source :

PRÉAMBULE

- Une image



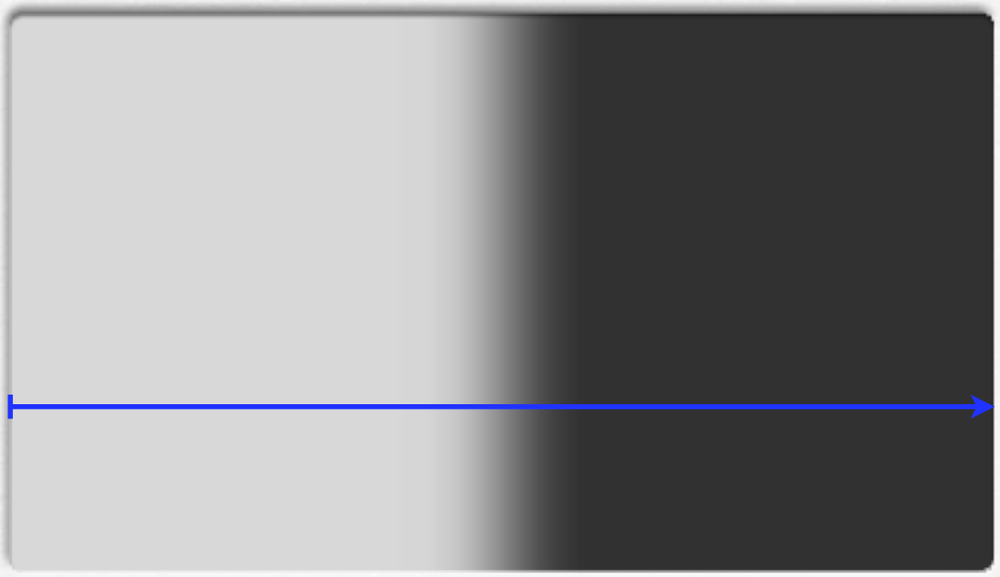
Hypothèse : Le contour représente la frontière d'un objet

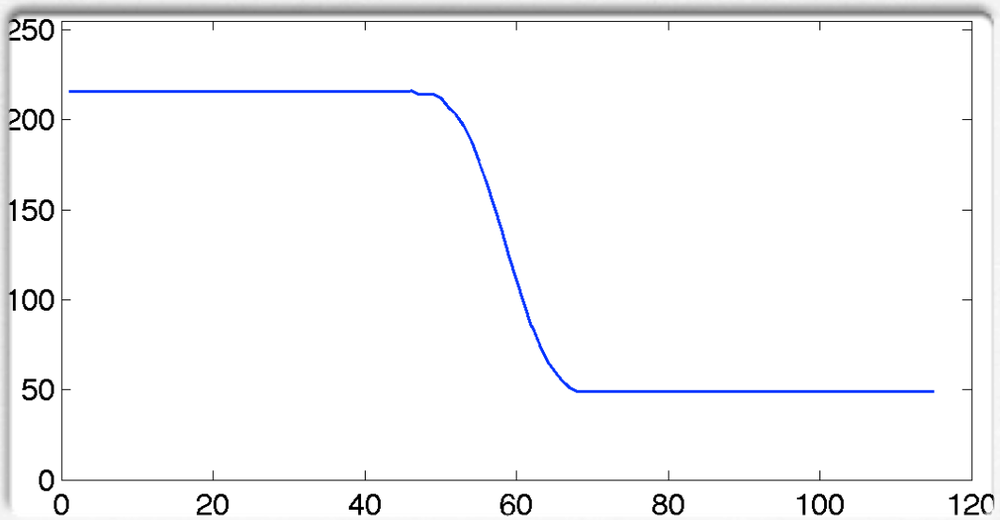
Source :

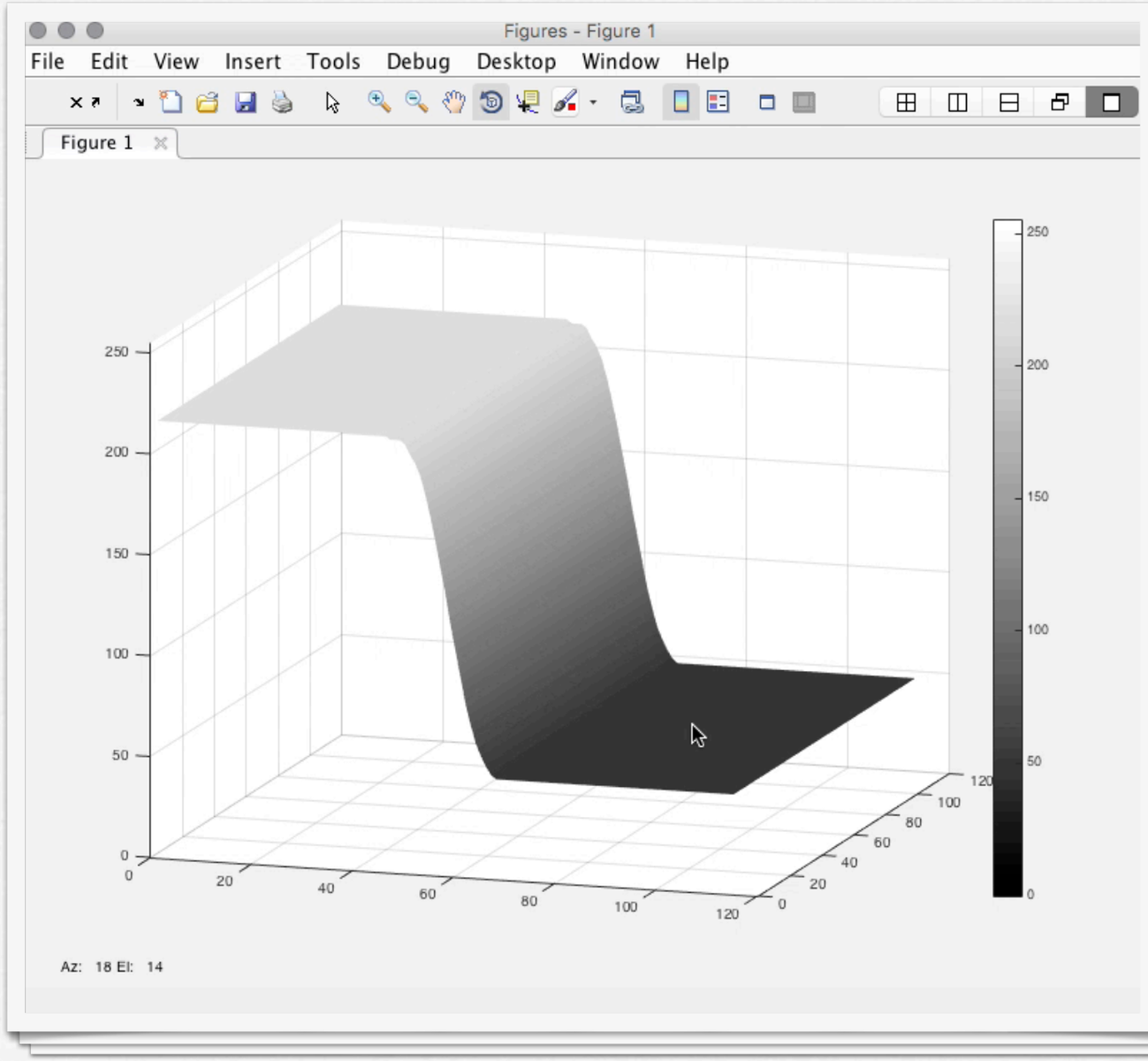
©Marie-Flavie Auclair-Fortier

PRÉAMBULE

- Une image





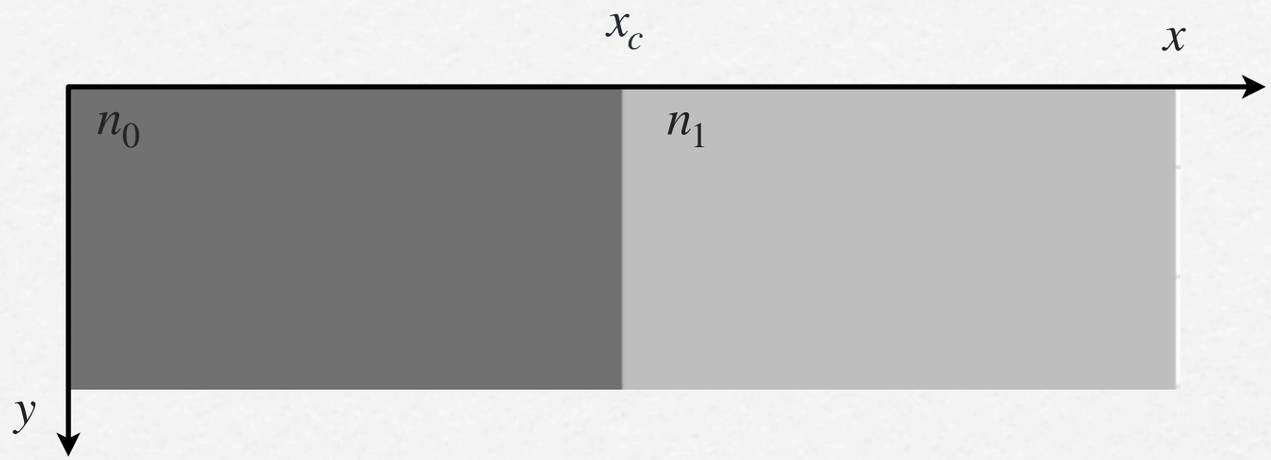


Hypothèse : Le contour représente la frontière d'un objet

Source :

PRÉAMBULE

a) Contour idéal



n_1

n_0

x_c

Source :

©Marie-Flavie Auclair-Fortier

PRÉAMBULE

a) Contour idéal

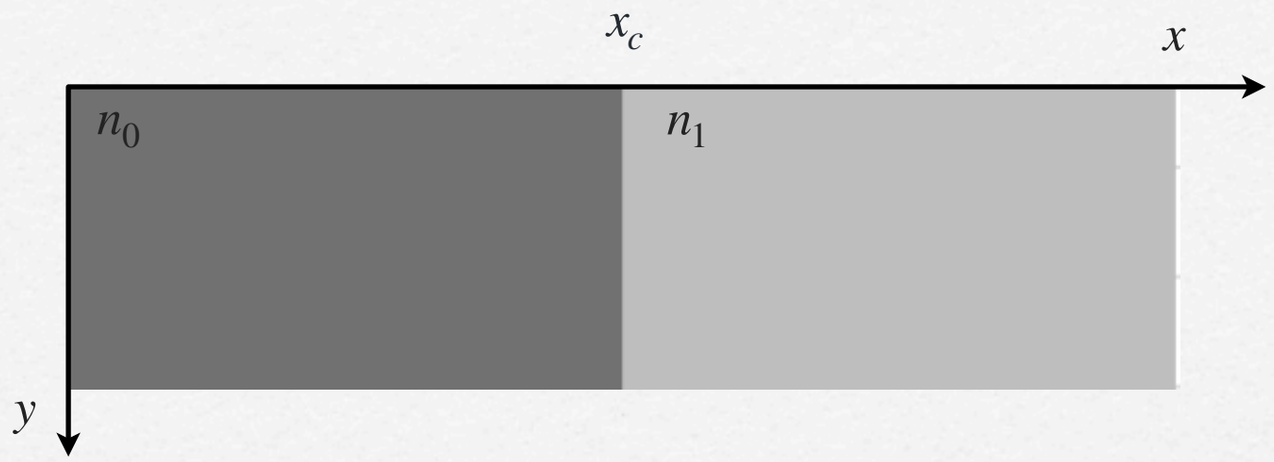


Image d'un contour idéal situé à x_c

n_1

n_0

x_c

Source :

©Marie-Flavie Auclair-Fortier

PRÉAMBULE

a) Contour idéal

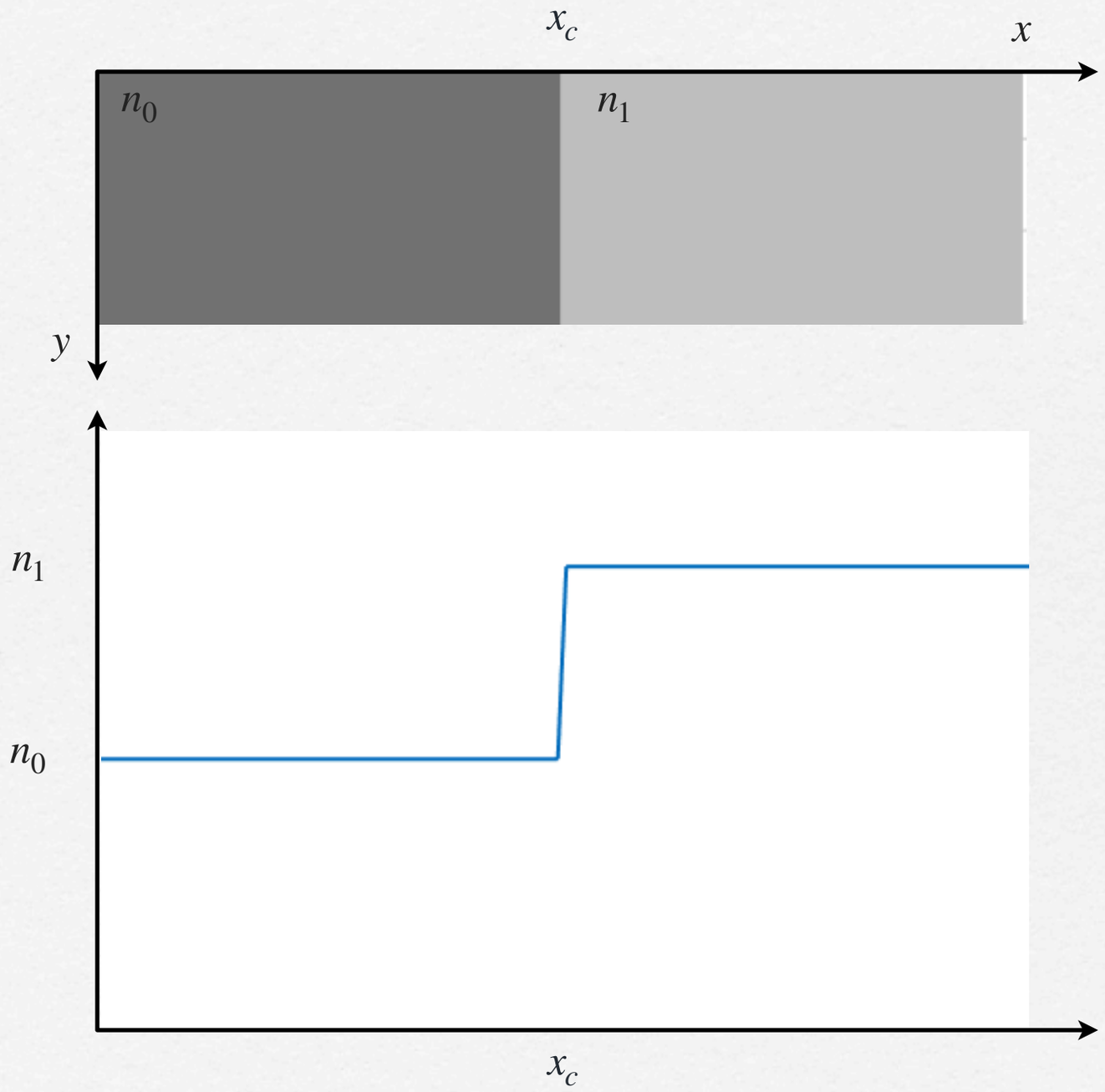


Image d'un contour idéal situé à x_c

profil du contour (une ligne de l'image)

Source :

PRÉAMBULE

a) Contour idéal

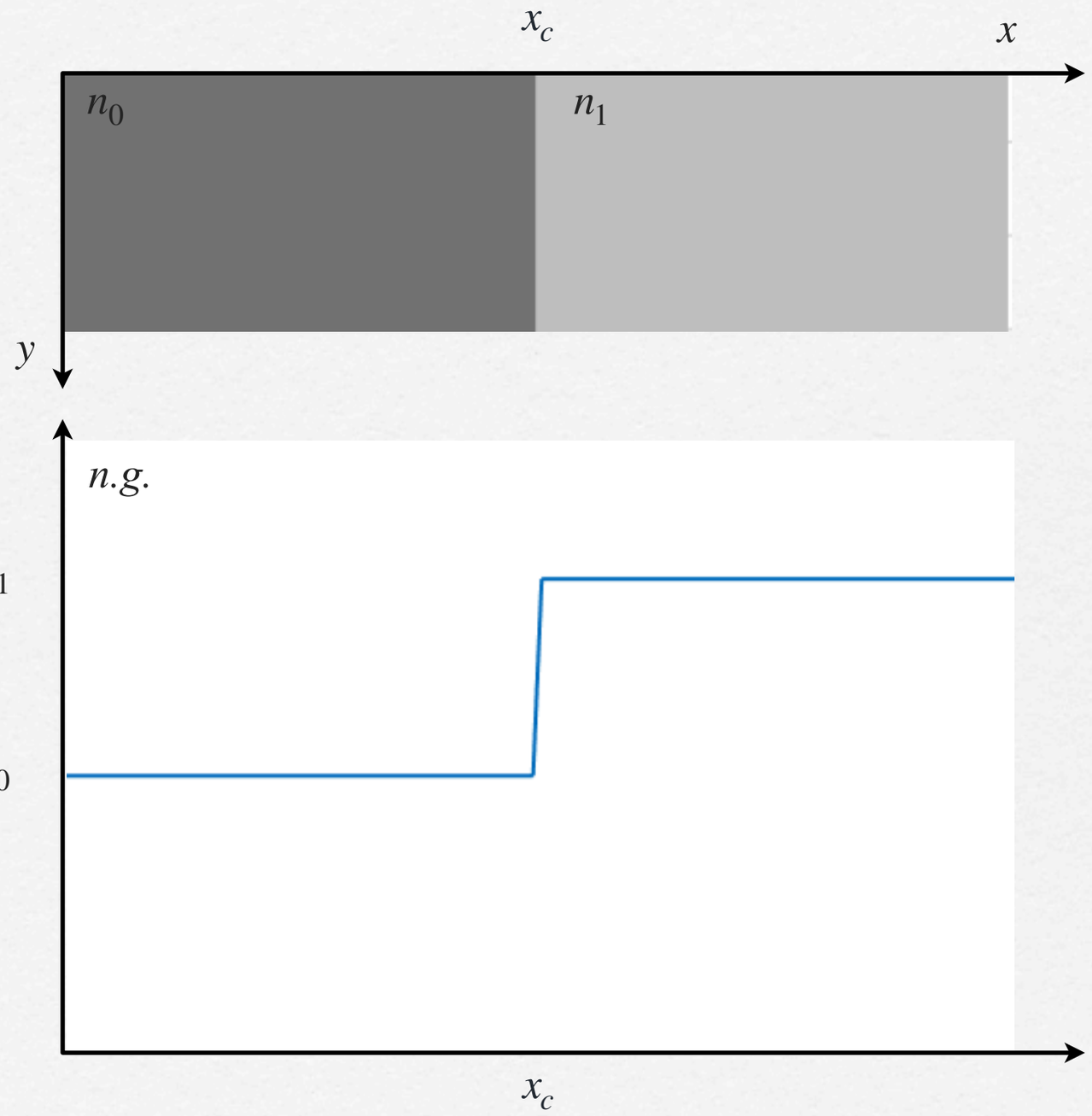


Image d'un contour idéal situé à x_c

profil du contour (une ligne de l'image)

Source :

©Marie-Flavie Auclair-Fortier

PRÉAMBULE

a) Contour idéal

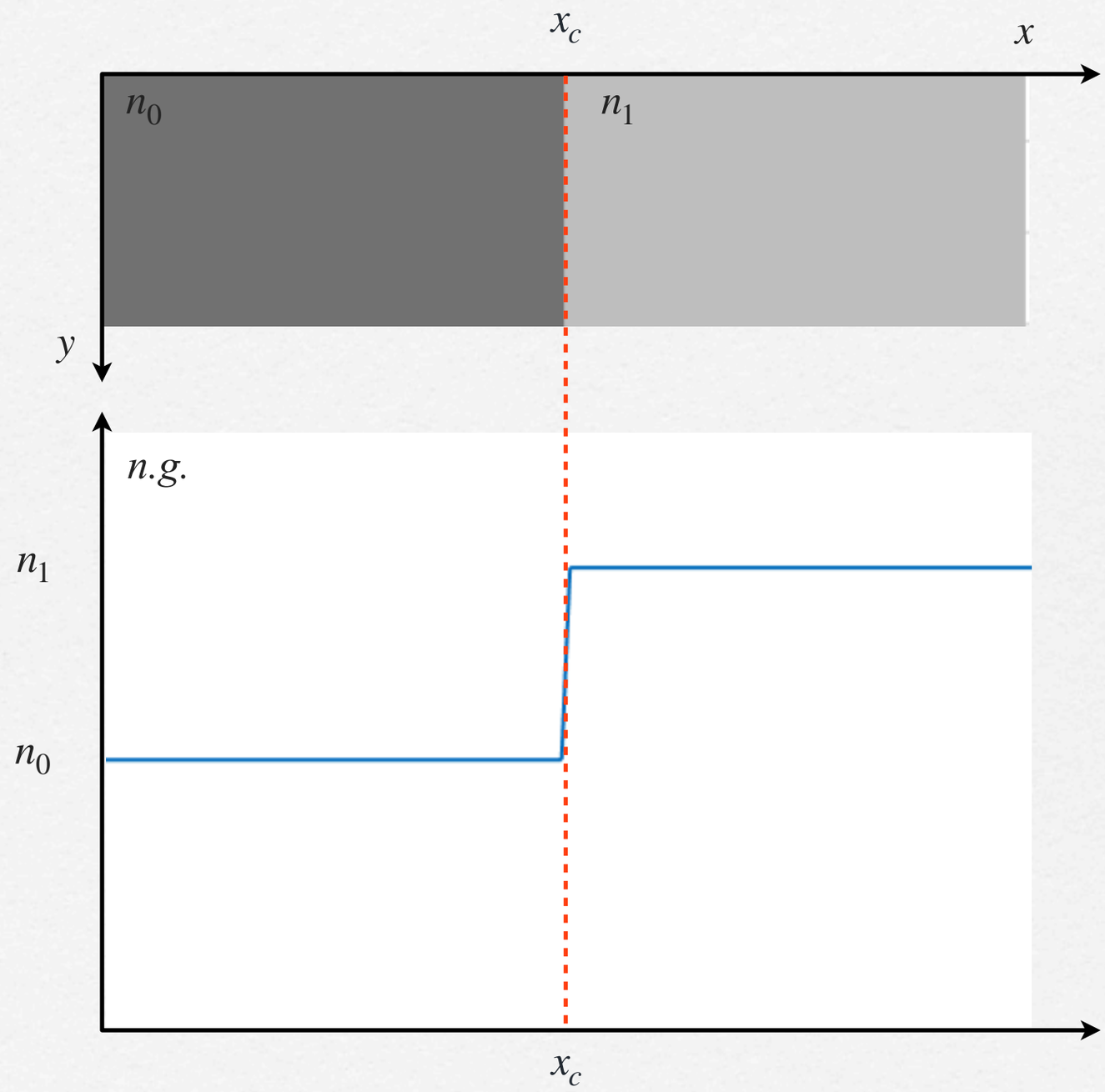


Image d'un contour idéal situé à x_c

profil du contour (une ligne de l'image)

Source :

PRÉAMBULE

b) Contour flou sans bruit

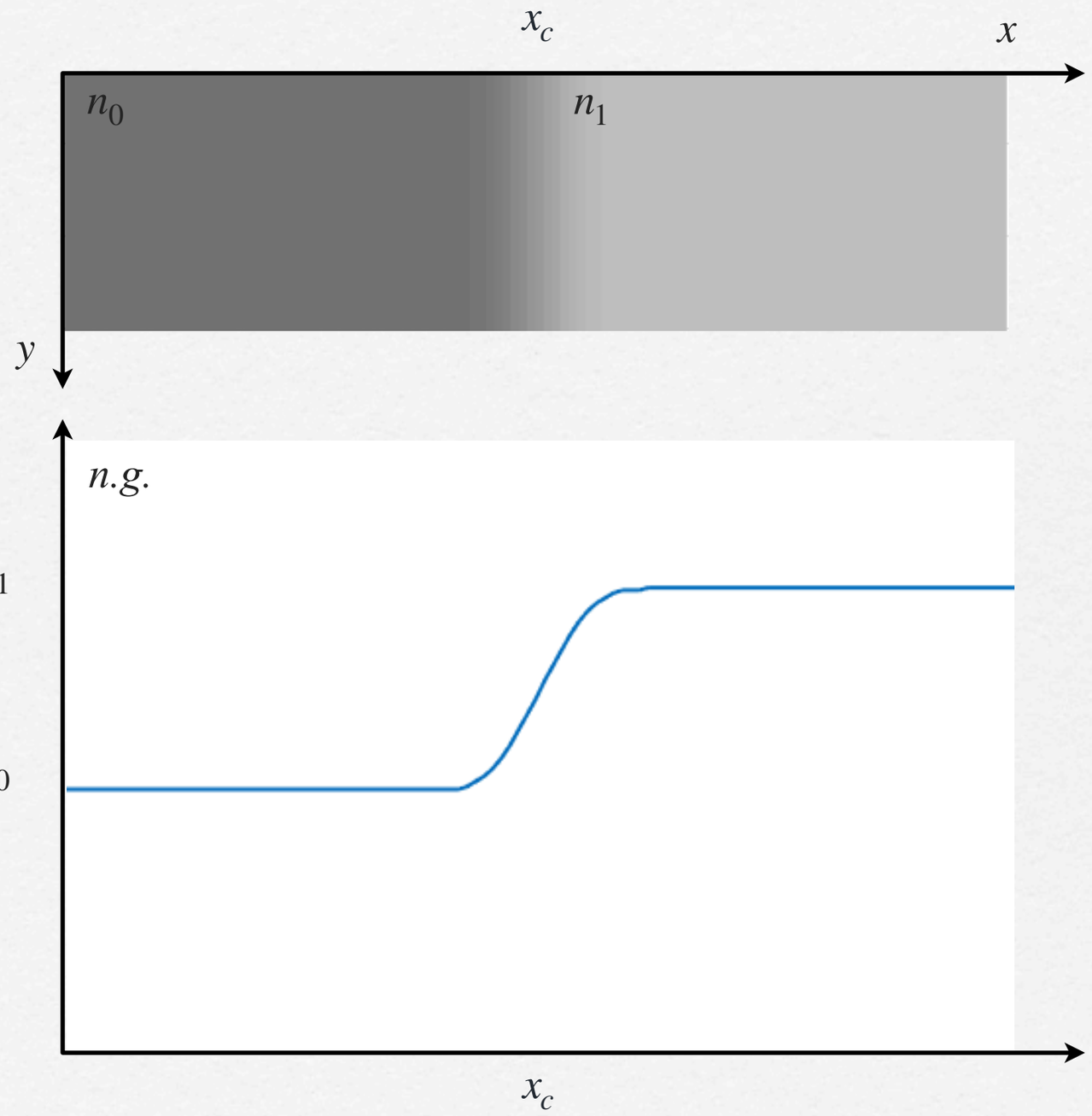


Image d'un contour idéal flou situé à x_c

Une ligne de l'image

Source :

©Marie-Flavie Auclair-Fortier

PRÉAMBULE

b) Contour flou sans bruit

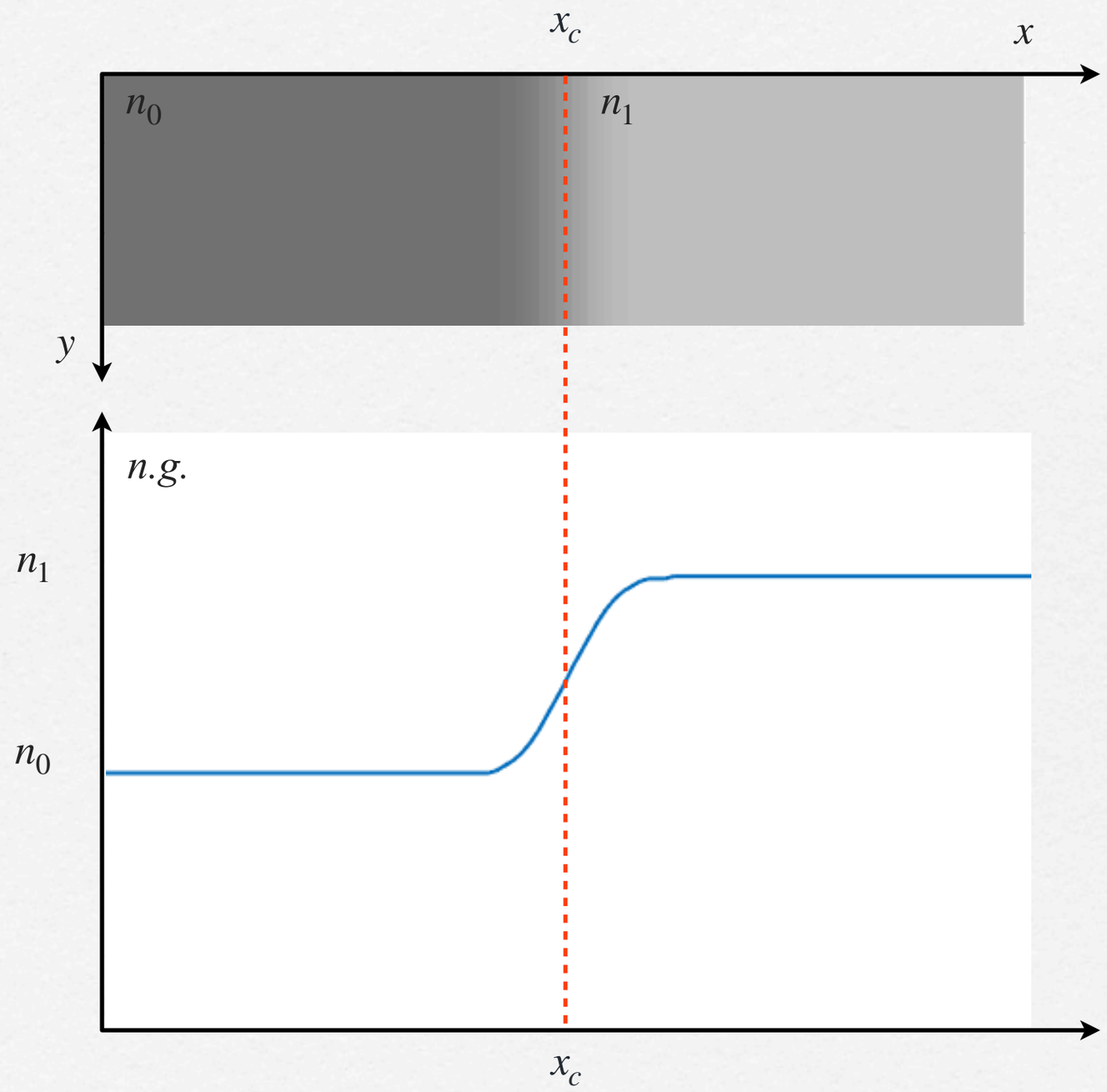


Image d'un contour idéal flou situé à x_c

Une ligne de l'image

Source :

PRÉAMBULE

c) Contour dans la pratique (flou et bruité)

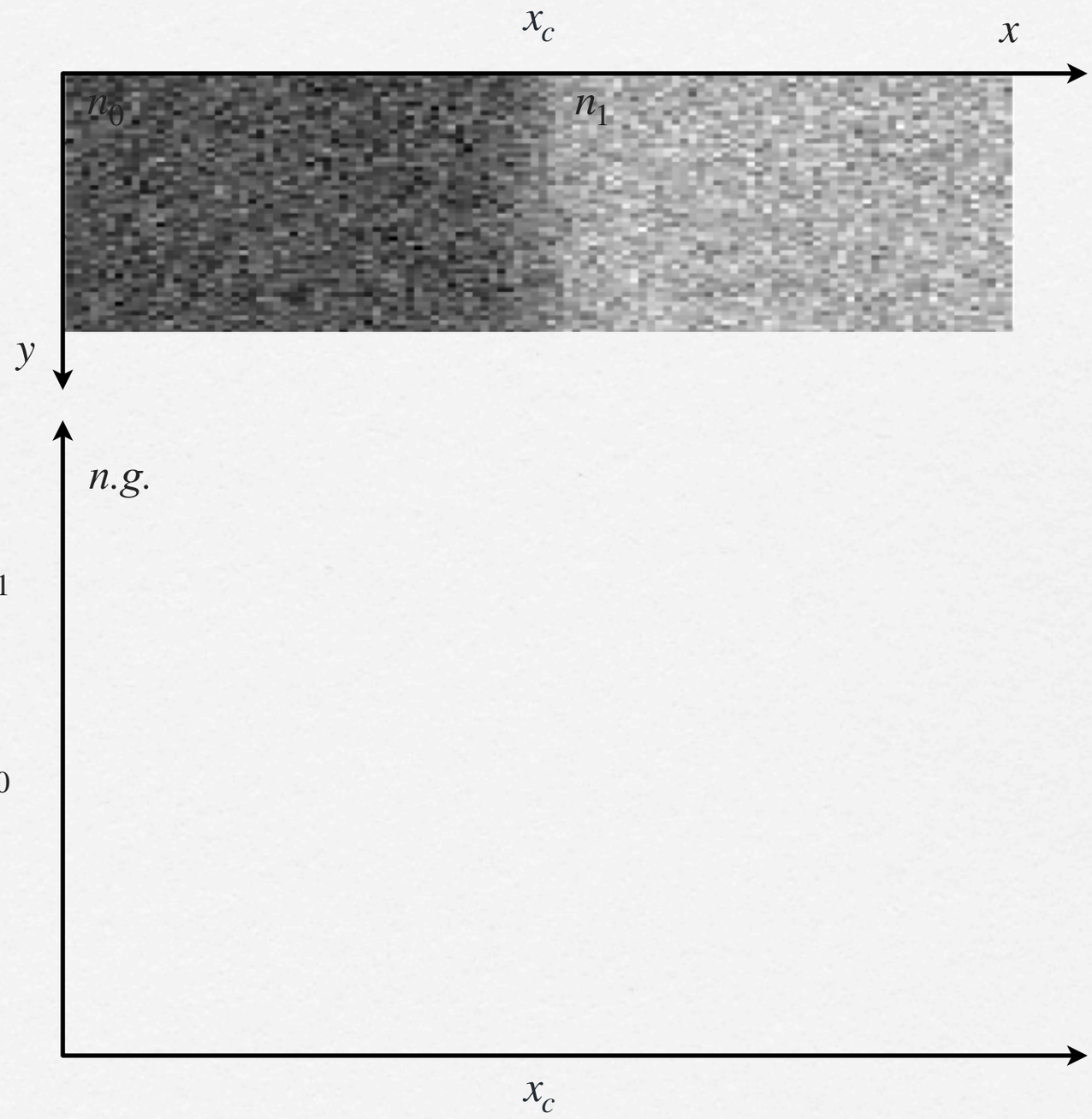


Image d'un contour réaliste situé à x_c

Une ligne de l'image

Source :

©Marie-Flavie Auclair-Fortier

PRÉAMBULE

c) Contour dans la pratique (flou et bruité)

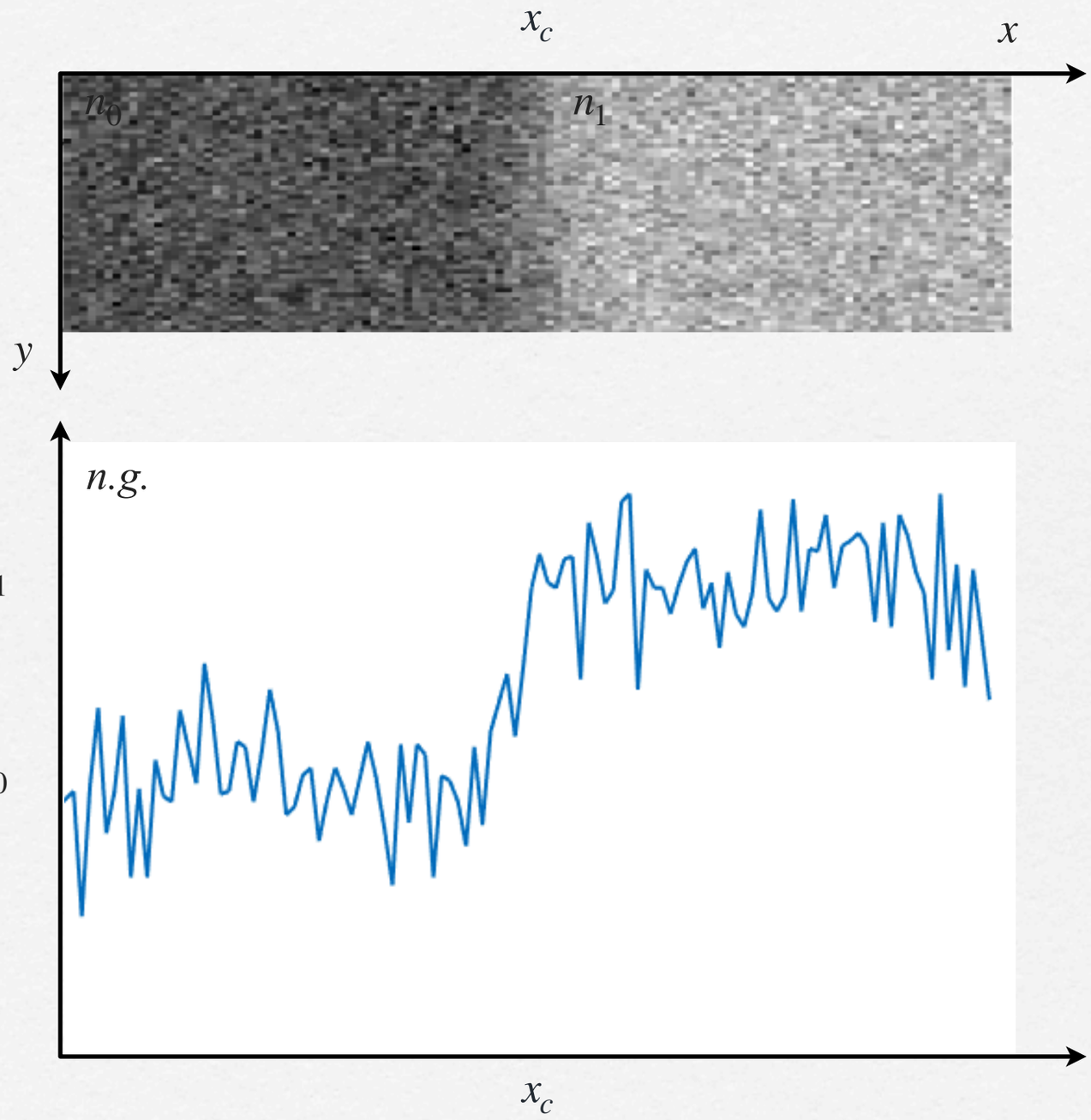


Image d'un contour réaliste situé à x_c

Une ligne de l'image

Source :

PRÉAMBULE

c) Contour dans la pratique (flou et bruité)

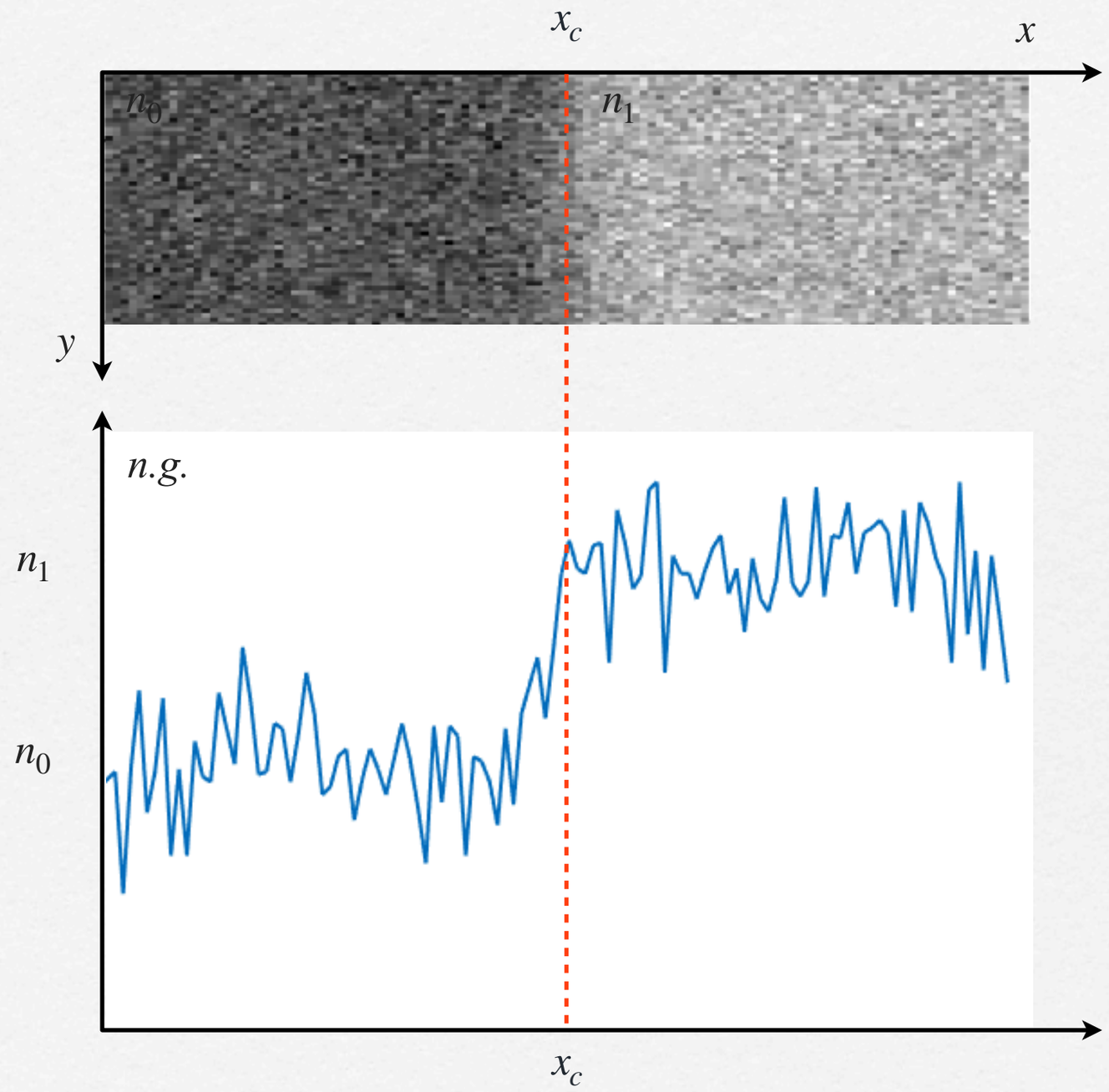


Image d'un contour réaliste situé à x_c

Une ligne de l'image

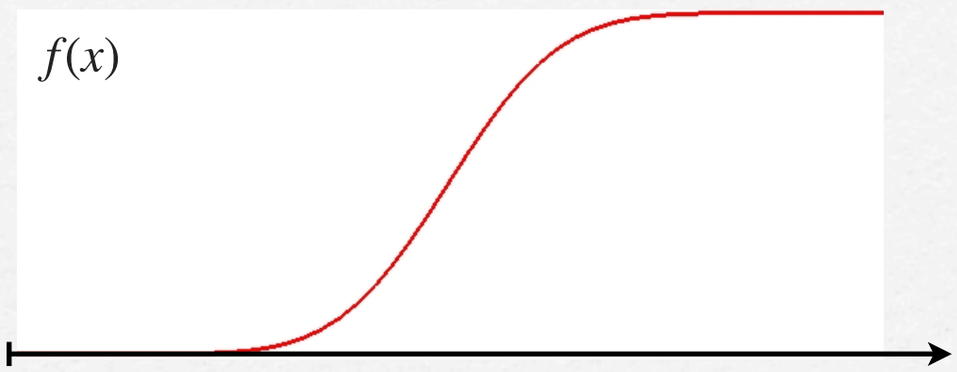
Source :

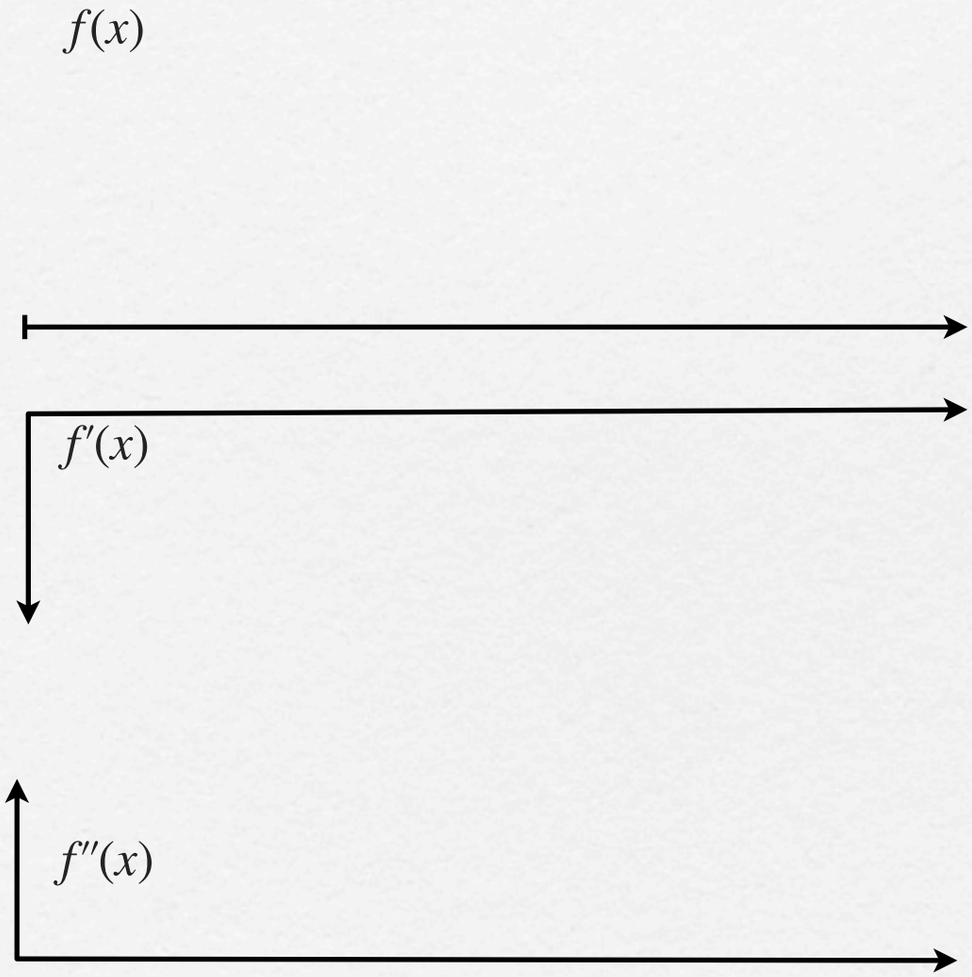
©Marie-Flavie Auclair-Fortier

PRÉAMBULE

d) Détection de contours en 1D

Marche montante





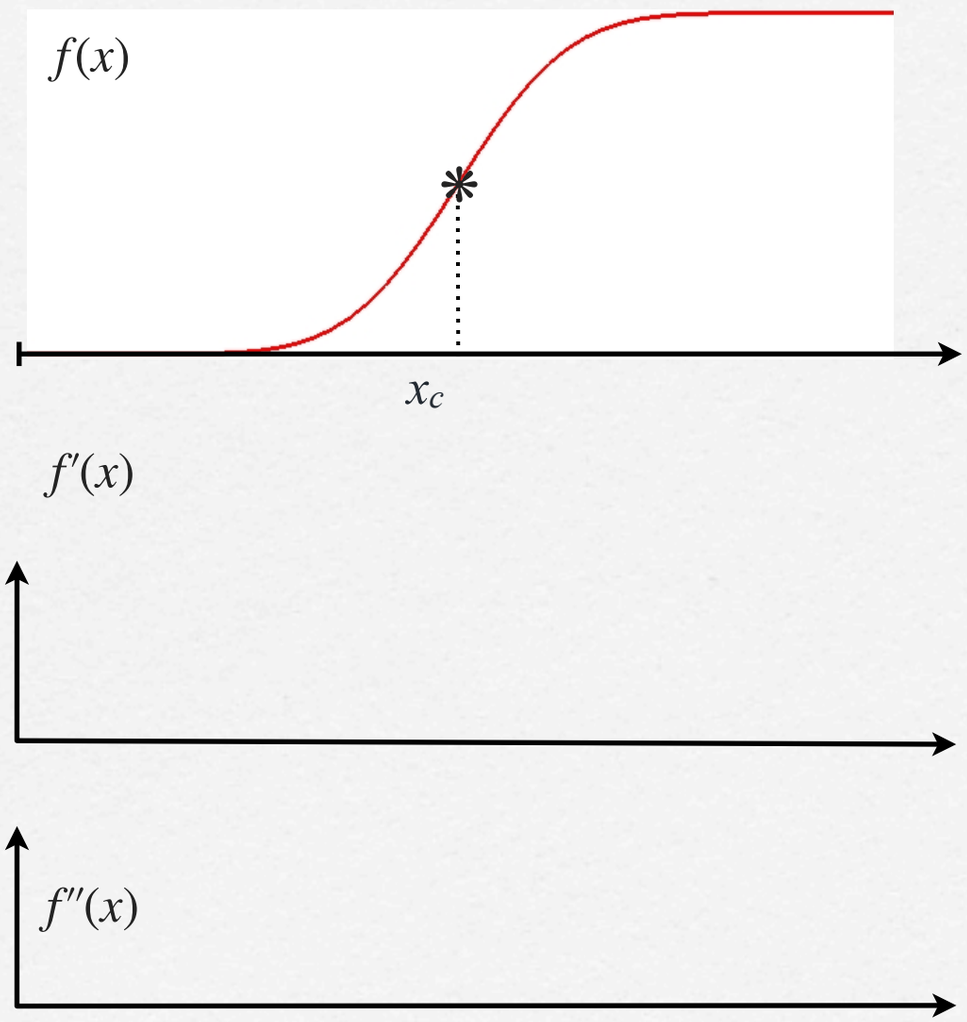
Source :

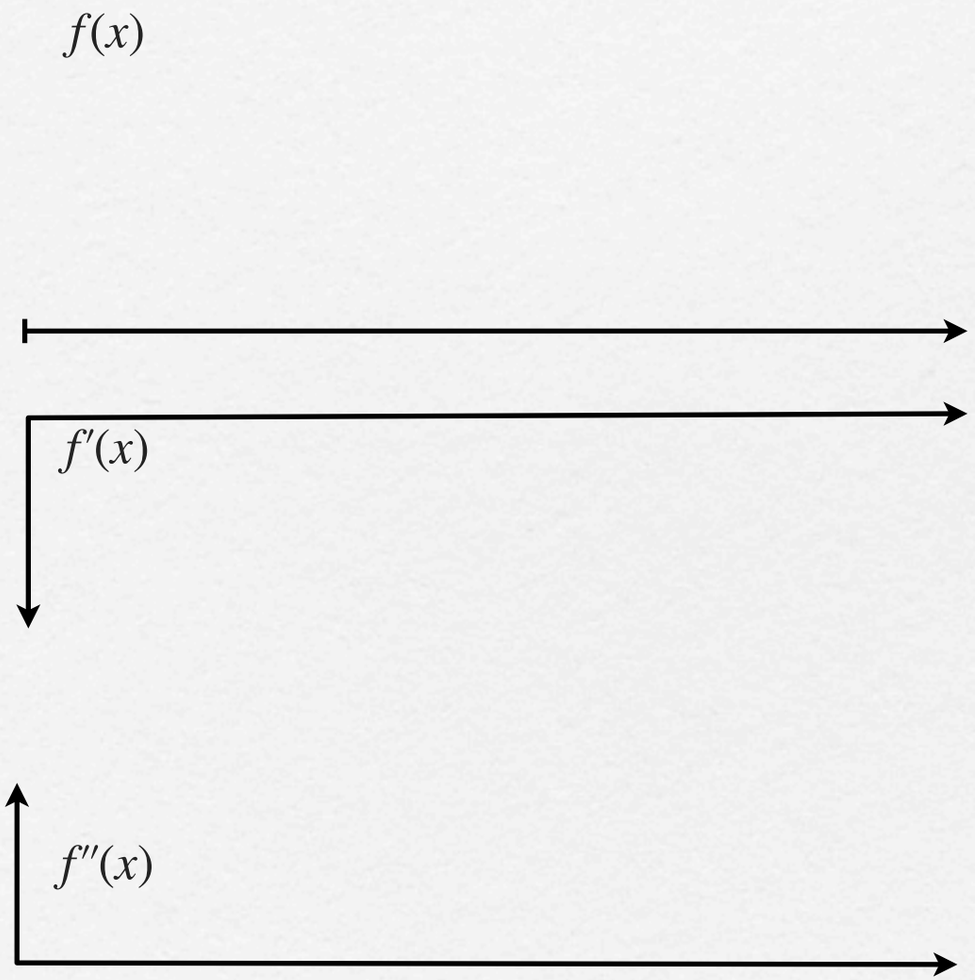
©Marie-Flavie Auclair-Fortier

PRÉAMBULE

d) Détection de contours en 1D

Marche montante





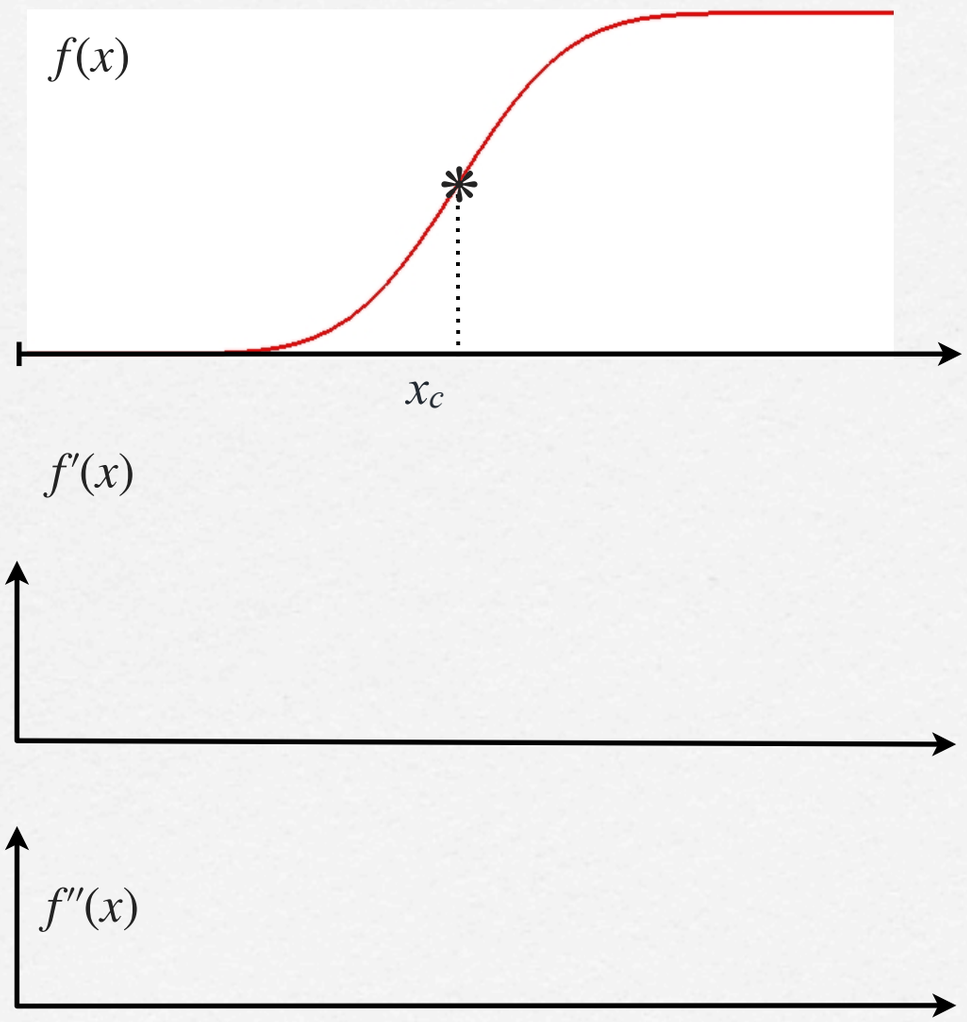
Source :

©Marie-Flavie Auclair-Fortier

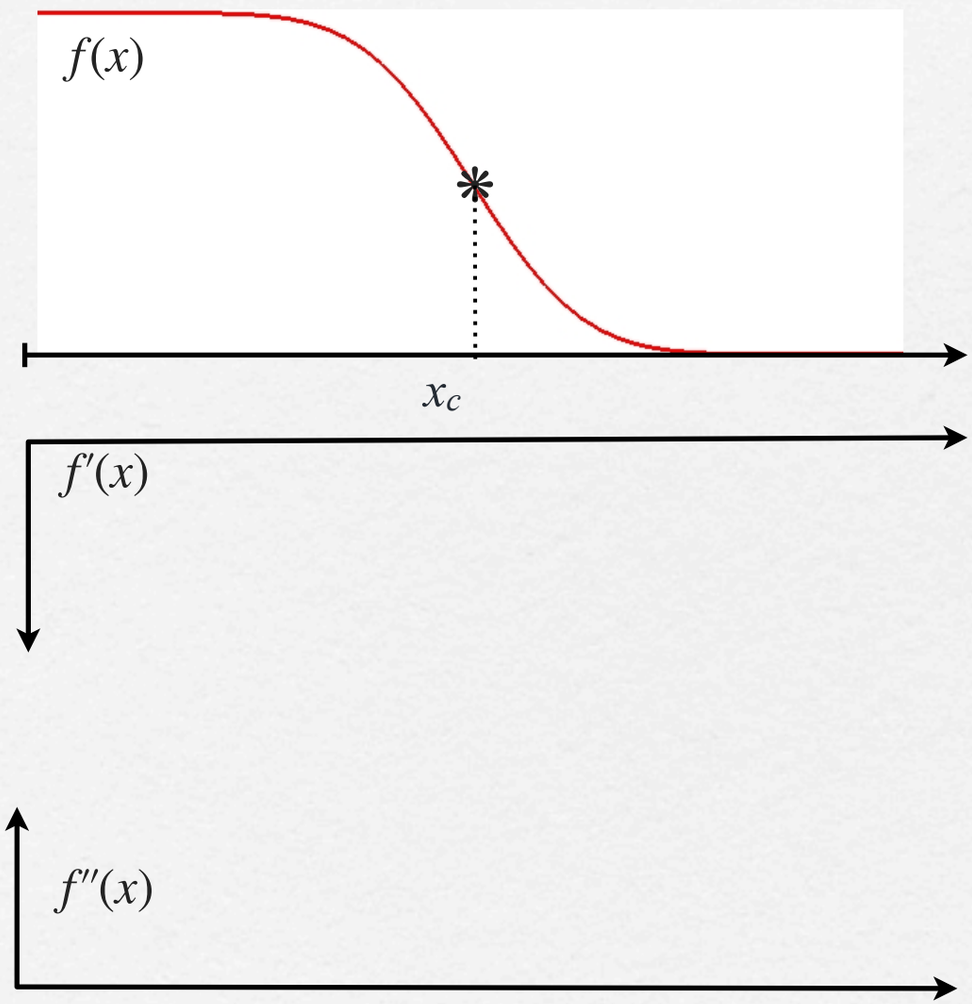
PRÉAMBULE

d) Détection de contours en 1D

Marche montante



Marche descendante



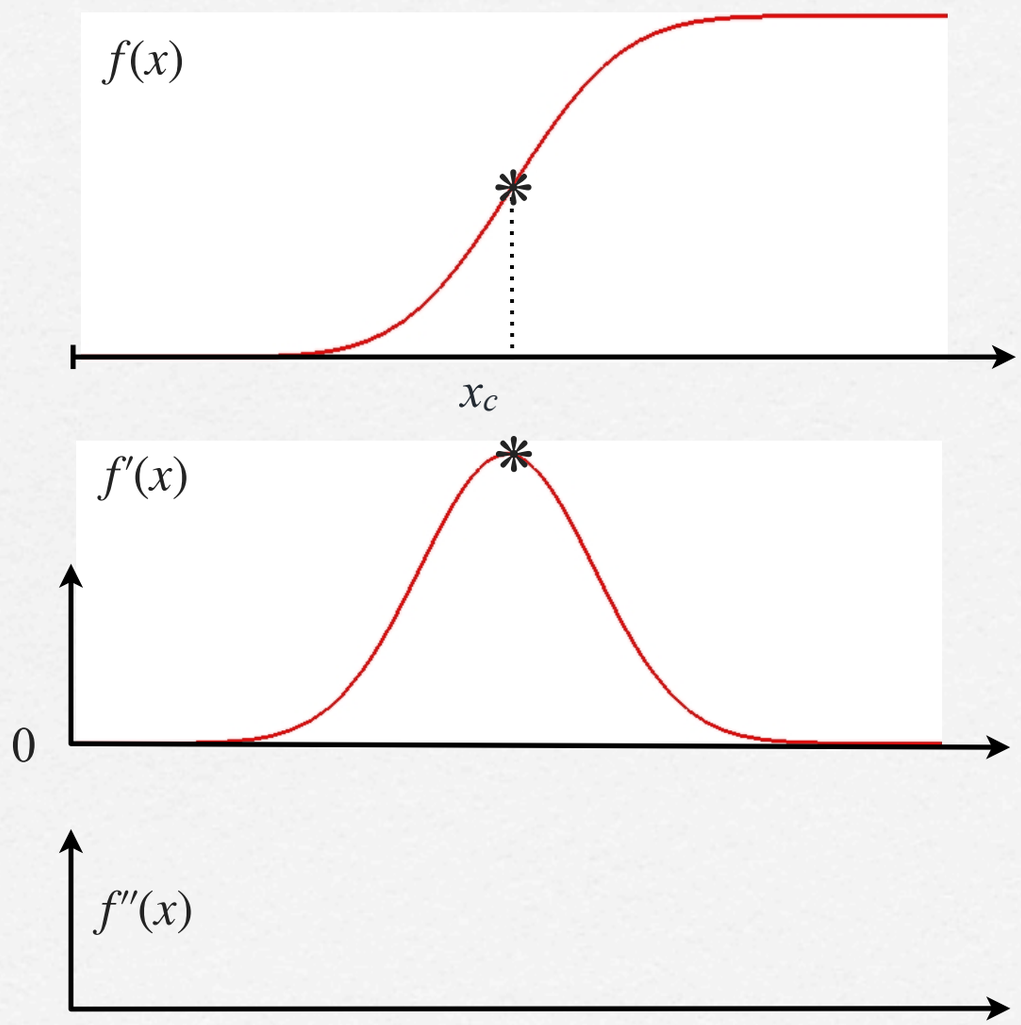
Source :

©Marie-Flavie Auclair-Fortier

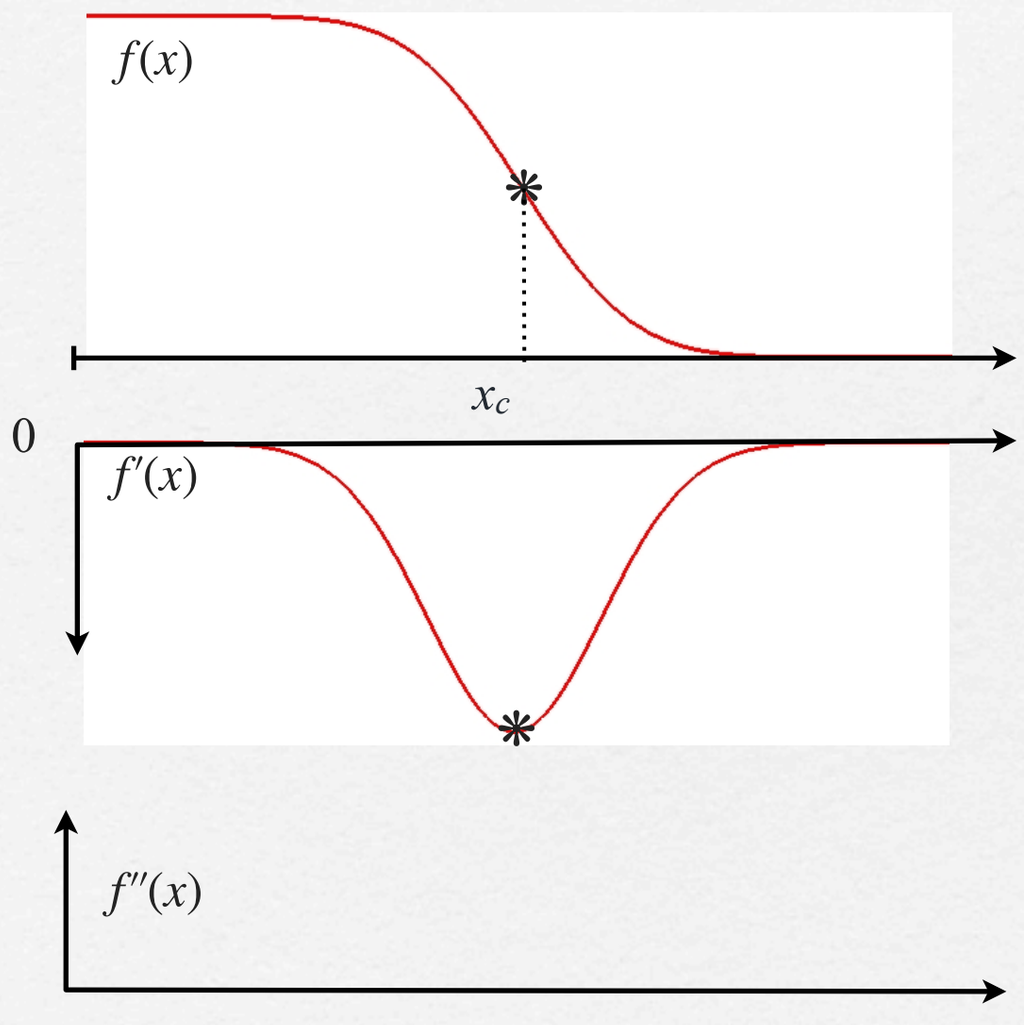
PRÉAMBULE

d) Détection de contours en 1D

Marche montante



Marche descendante



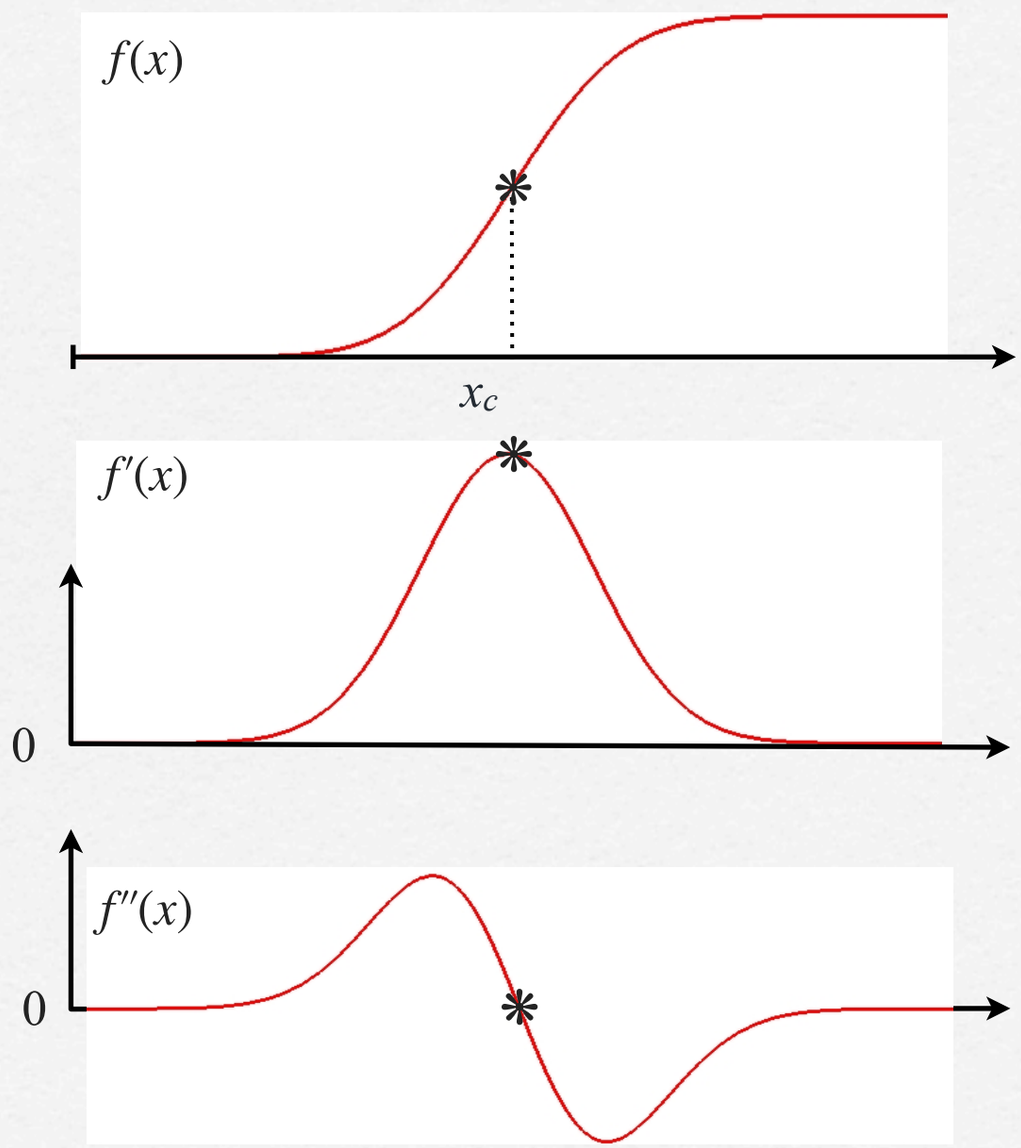
Source :

©Marie-Flavie Auclair-Fortier

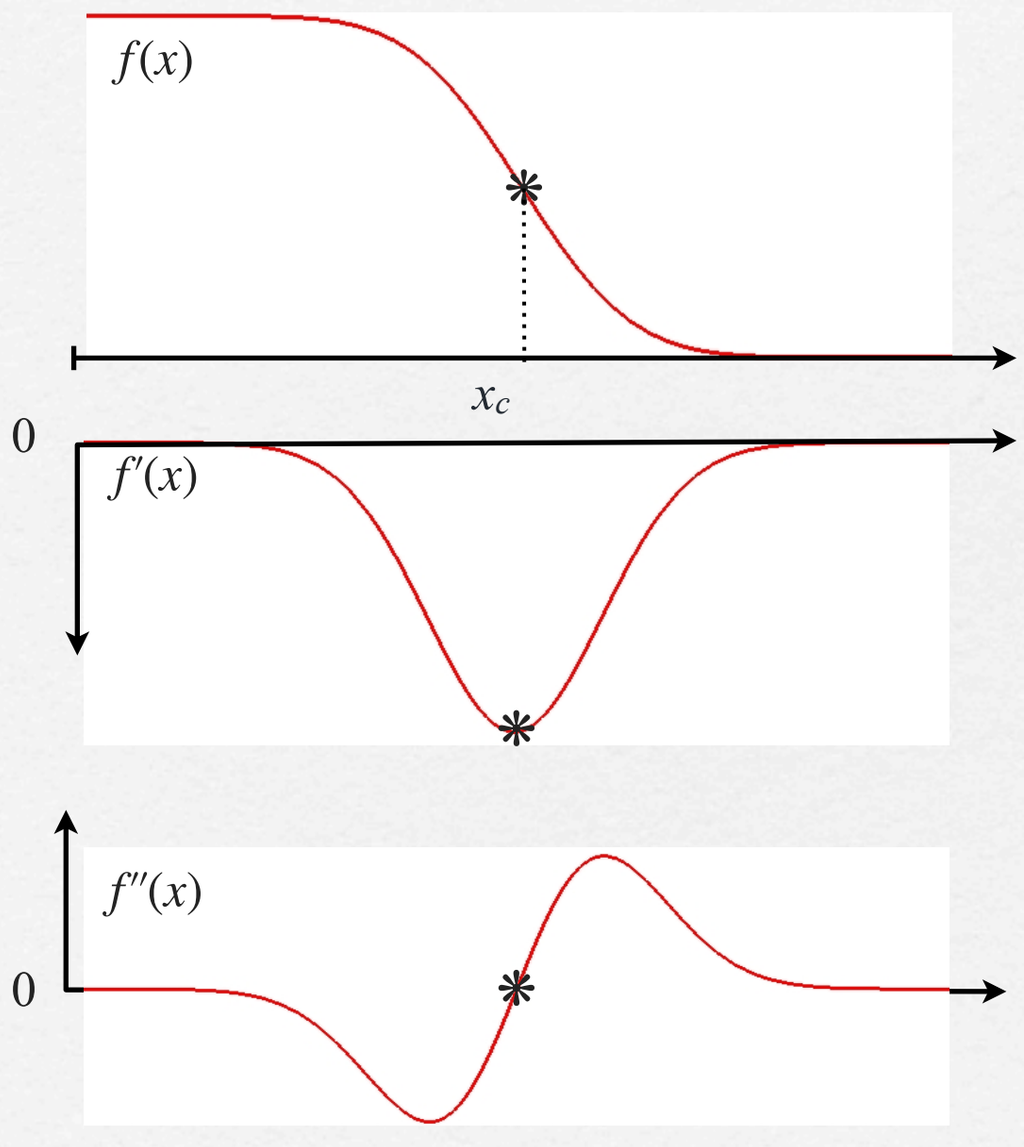
PRÉAMBULE

d) Détection de contours en 1D

Marche montante



Marche descendante

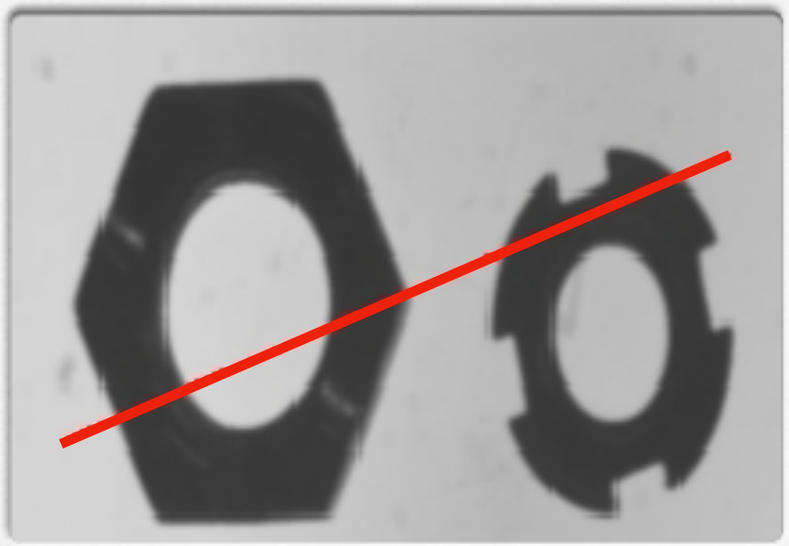


Source :

©Marie-Flavie Auclair-Fortier

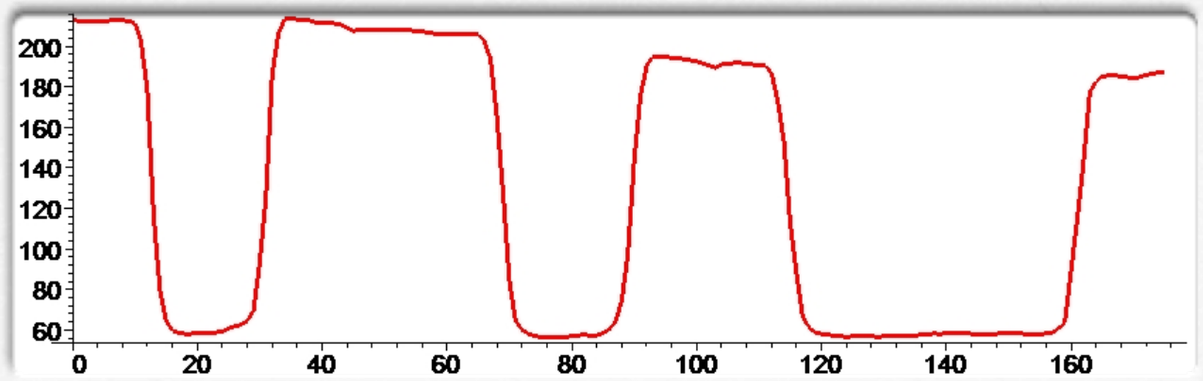
PRÉAMBULE

d) Détection de contours en 1D



Une coupe des niveaux gris dans une image à deux dimensions

Niveaux de gris le long de la ligne rouge

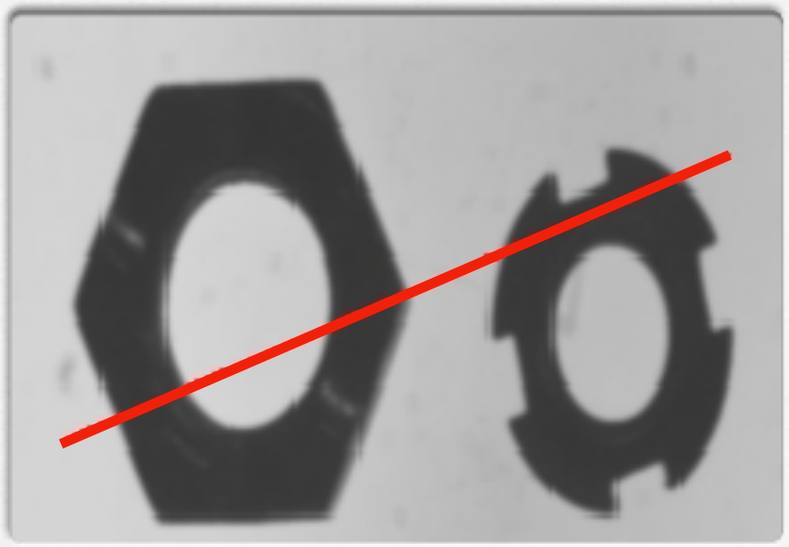


Source :

©Marie-Flavie Auclair-Fortier

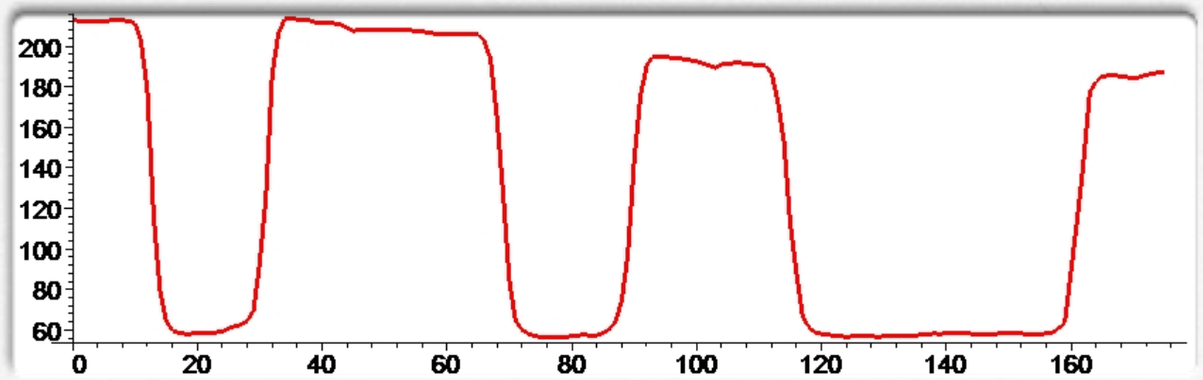
PRÉAMBULE

d) Détection de contours en 1D

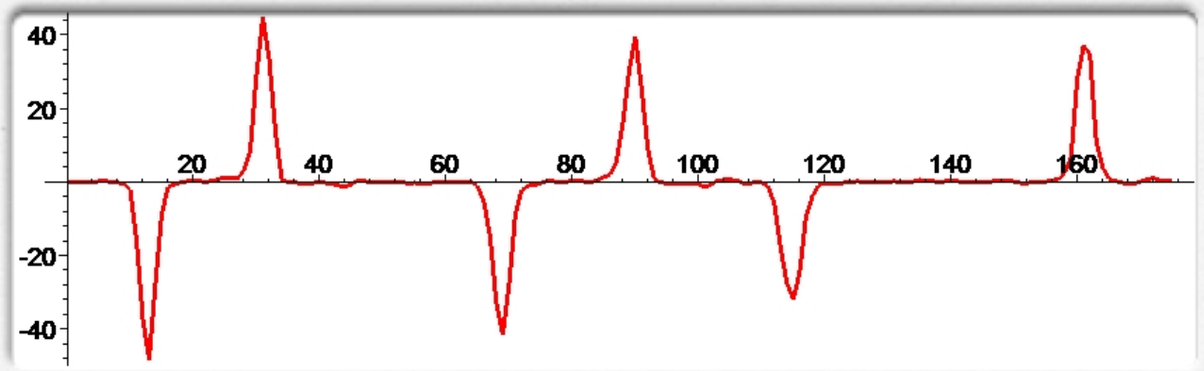


Une coupe des niveaux gris dans une image à deux dimensions

Niveaux de gris le long de la ligne rouge



Dérivée d'ordre 1 des niveaux de gris: les contours sont les ***maxima*** positifs et les ***minima*** négatifs

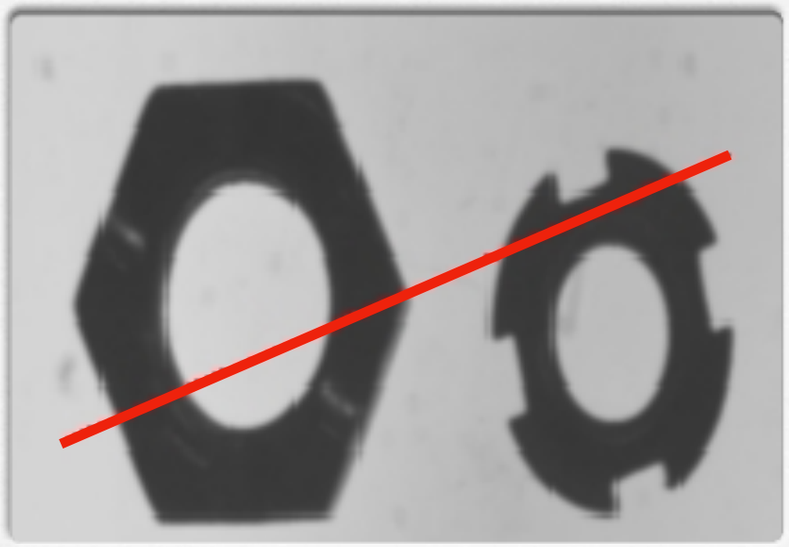


Source :

©Marie-Flavie Auclair-Fortier

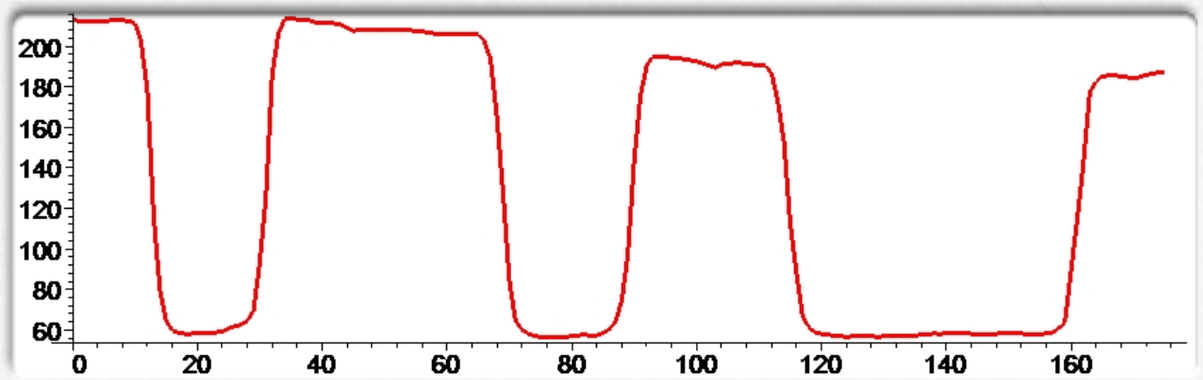
PRÉAMBULE

d) Détection de contours en 1D

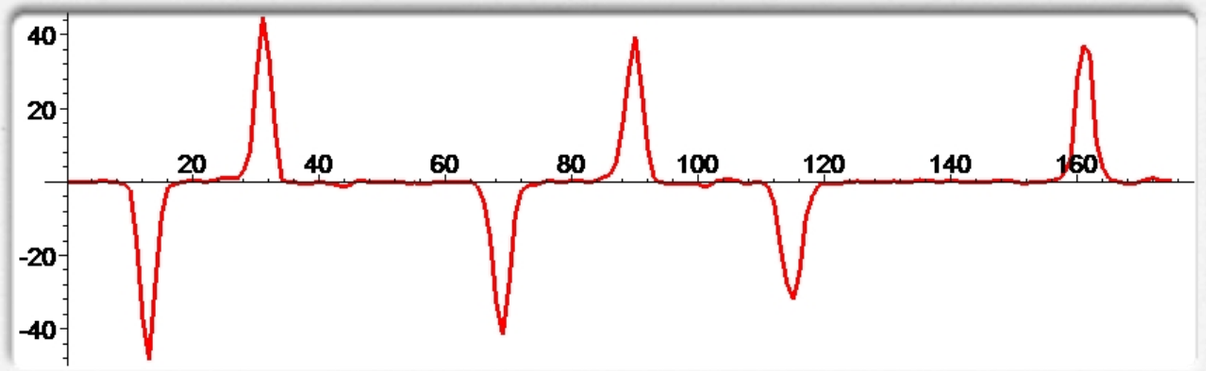


Une coupe des niveaux gris dans une image à deux dimensions

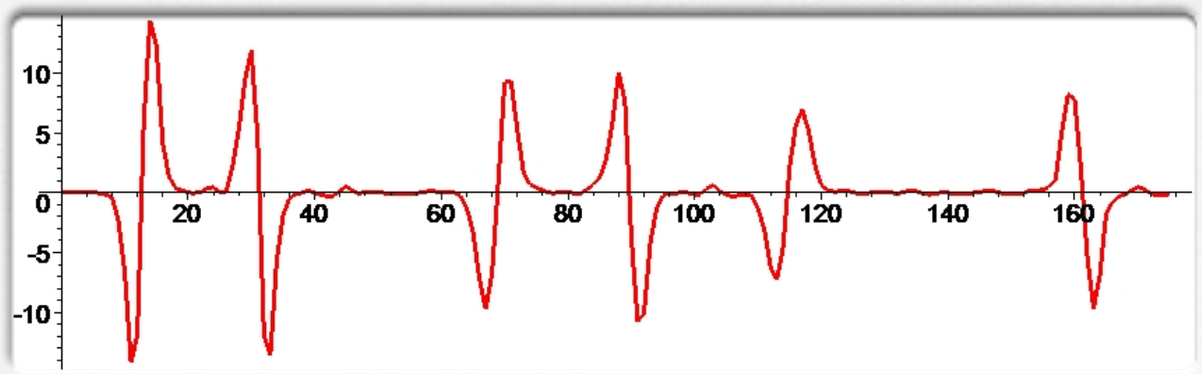
Niveaux de gris le long de la ligne rouge



Dérivée d'ordre 1 des niveaux de gris: les contours sont les ***maxima*** positifs et les ***minima*** négatifs



Dérivée d'ordre 2 des niveaux de gris : Les contours sont les ***passages*** par 0

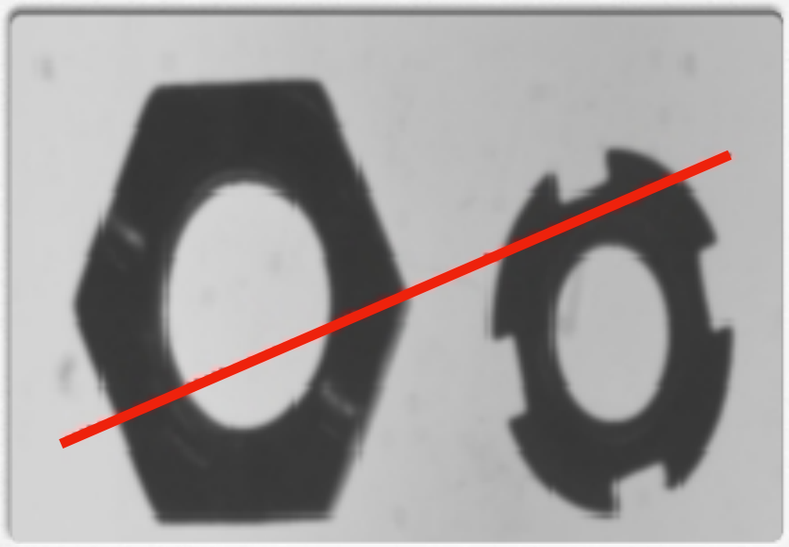


Source :

©Marie-Flavie Auclair-Fortier

PRÉAMBULE

d) Détection de contours en 1D

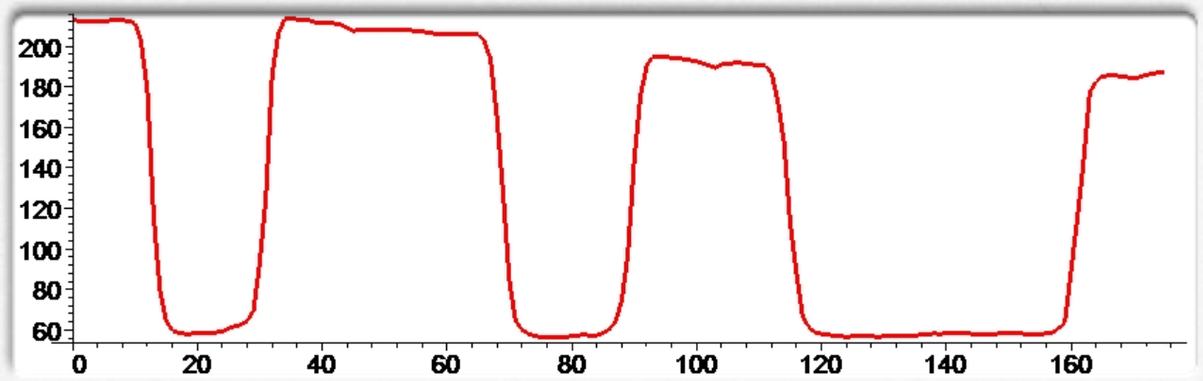


Une coupe des niveaux gris dans une image à deux dimensions

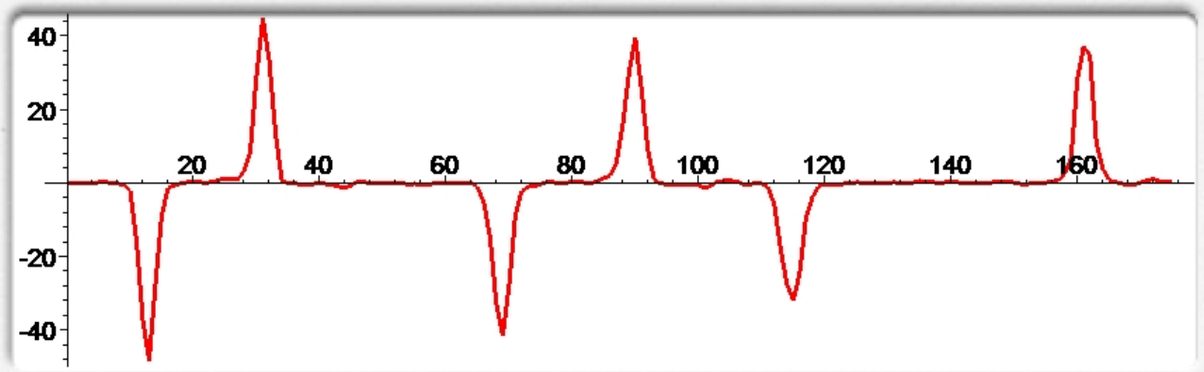
Remarque :

- la dérivée seconde peut être égale à 0 sans que le point soit un point d'inflexion

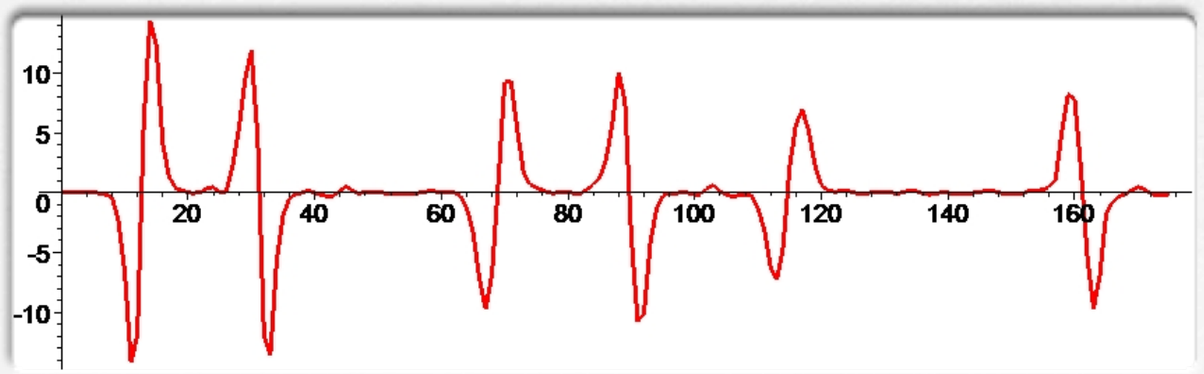
Niveaux de gris le long de la ligne rouge



Dérivée d'ordre 1 des niveaux de gris: les contours sont les ***maxima*** positifs et les ***minima*** négatifs



Dérivée d'ordre 2 des niveaux de gris : Les contours sont les ***passages*** par 0



Source :

IMN259 Analyse d'images
©Marie-Flavie Auclair-Fortier

Plan

1. Calcul des dérivées d'une image
   1.1. Dérivées partielles
   1.2. Gradient
   1.3. Laplacien
2. Contours
3. Coins
4. Segmentation

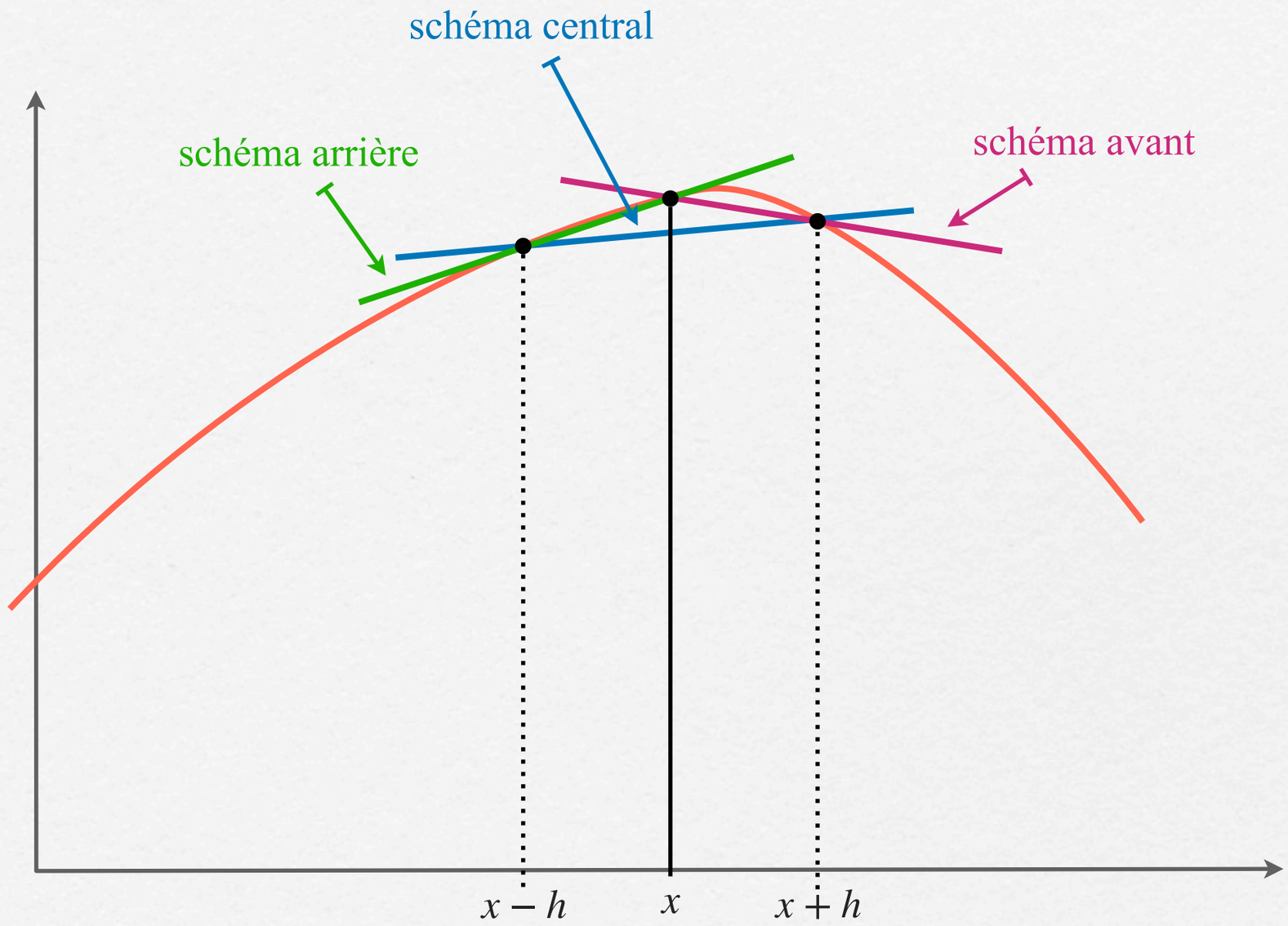
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 1



Source : https://commons.wikimedia.org/wiki/File:Finite_difference_method.svg

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 1
  - ✓ 2D :

		masque
$\frac{\partial f}{\partial m}[m,n] \approx$ avant :	$f[m+1,n]-f[m,n]$	$\begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$
arrière :	$f[m,n]-f[m-1,n]$	$\begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$
centrale :	$\frac{f[m+1,n]-f[m-1,n]}{2}$	$\begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix}$

Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 1
  - ✓ 2D :

$$\frac{\partial f}{\partial m}[m,n] \approx$$

		masque
avant :	$f[m+1,n]-f[m,n]$	$\begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$
arrière :	$f[m,n]-f[m-1,n]$	$\begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$
centrale :	$\frac{f[m+1,n]-f[m-1,n]}{2}$	$\begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix}$

$$\frac{\partial f}{\partial n}[m,n] \approx$$

		masque
avant :	$f[m,n+1]-f[m,n]$	$\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$
arrière :	$f[m,n]-f[m,n-1]$	$\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$
centrale :	$\frac{f[m,n+1]-f[m,n-1]}{2}$	$\begin{bmatrix} -1/2 \\ 0 \\ 1/2 \end{bmatrix}$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 1
  - ✓ 2D :

$$\frac{\partial f}{\partial m}[m,n] \approx$$

		masque
avant :	$f[m+1,n]-f[m,n]$	$\begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$
arrière :	$f[m,n]-f[m-1,n]$	$\begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$
centrale :	$\frac{f[m+1,n]-f[m-1,n]}{2}$	$\begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix}$

$$\frac{\partial f}{\partial n}[m,n] \approx$$

		masque
avant :	$f[m,n+1]-f[m,n]$	$\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$
arrière :	$f[m,n]-f[m,n-1]$	$\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$
centrale :	$\frac{f[m,n+1]-f[m,n-1]}{2}$	$\begin{bmatrix} -1/2 \\ 0 \\ 1/2 \end{bmatrix}$

Attention :

- avant, arrière et centrale: filtres non symétriques
  - ✓ $f[m,n] * h[-m,-n] = (f \circ h)[m,n]$
- centrale: antisymétrique
  - ✓ $(f * -h)[m,n] = -(f * h)[m,n]$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 1
  - ✓ 2D :

Important :

- Pour toutes les dérivées, il faut que la somme des éléments du masque = 0

		masque
$\frac{\partial f}{\partial m}[m,n] \approx$ avant :	$f[m+1,n]-f[m,n]$	$\begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$
arrière :	$f[m,n]-f[m-1,n]$	$\begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$
centrale :	$\frac{f[m+1,n]-f[m-1,n]}{2}$	$\begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix}$

		masque
$\frac{\partial f}{\partial n}[m,n] \approx$ avant :	$f[m,n+1]-f[m,n]$	$\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$
arrière :	$f[m,n]-f[m,n-1]$	$\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$
centrale :	$\frac{f[m,n+1]-f[m,n-1]}{2}$	$\begin{bmatrix} -1/2 \\ 0 \\ 1/2 \end{bmatrix}$

Attention :

- avant, arrière et centrale: filtres non symétriques
  - ✓ $f[m,n] * h[-m,-n] = (f \circ h)[m,n]$
- centrale: antisymétrique
  - ✓ $(f * -h)[m,n] = -(f * h)[m,n]$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

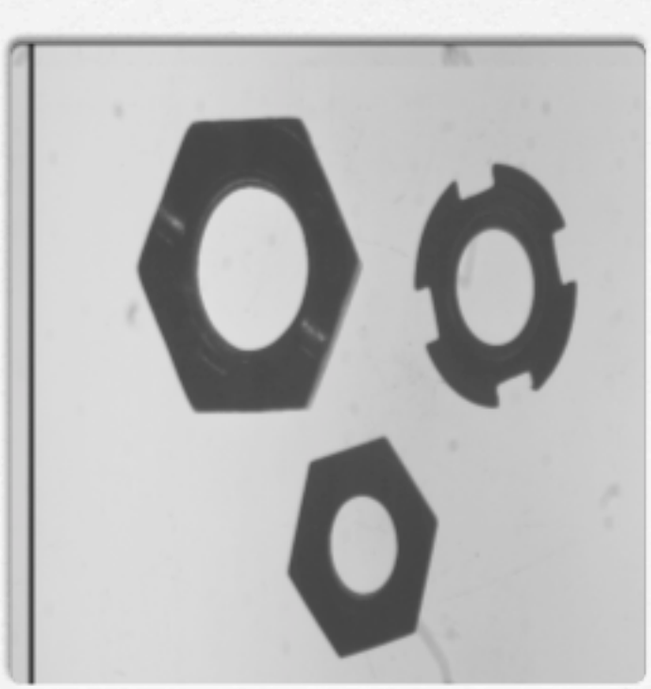
1. Dérivées partielles

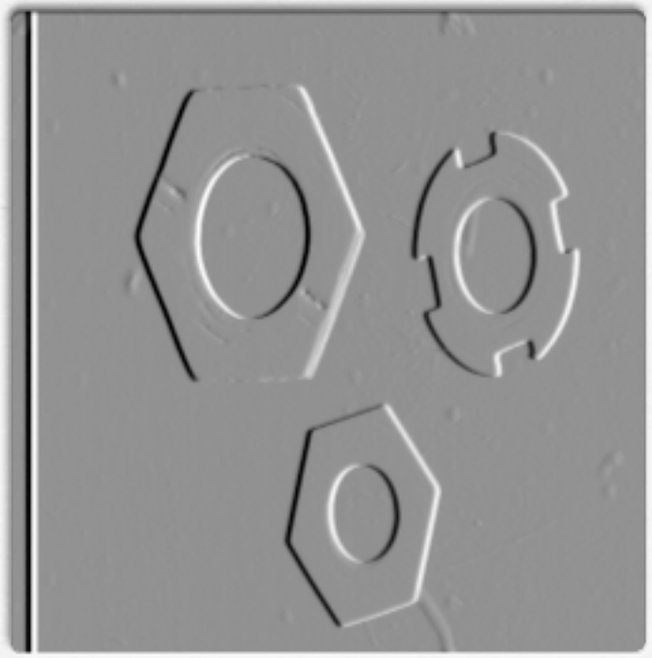
a) Différences finies

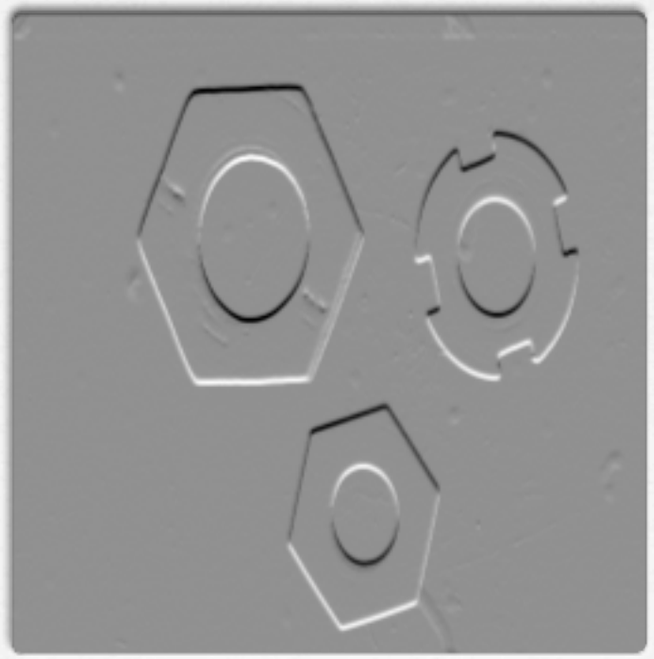
- Dérivées d'ordre 1
  - ✓ 2D :

$$\frac{\partial f}{\partial m}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$\frac{\partial f}{\partial n}[m,n] \approx f[m,n] * \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$







Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

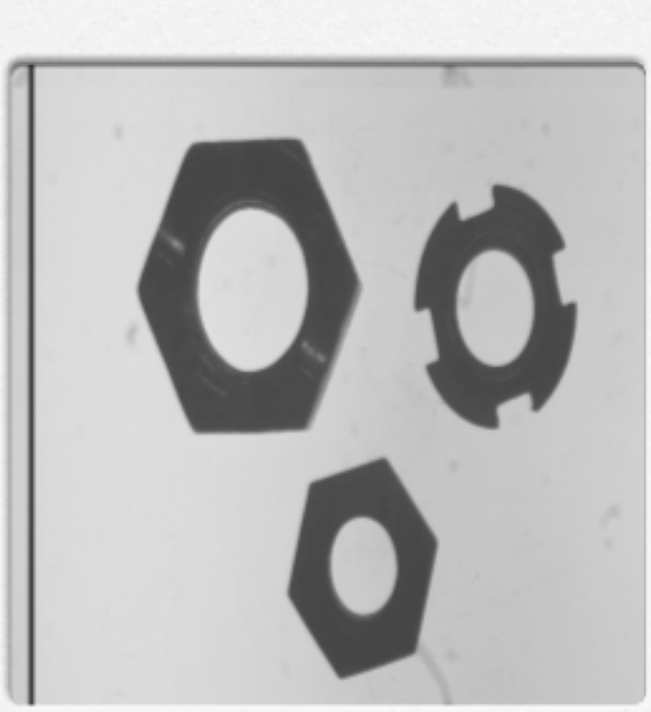
- Dérivées d'ordre 1
  - ✓ 2D :

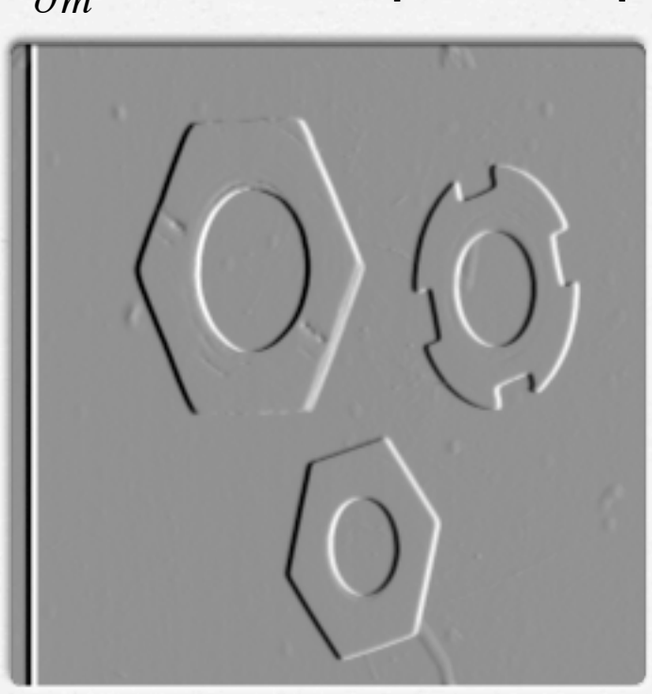
Attention :

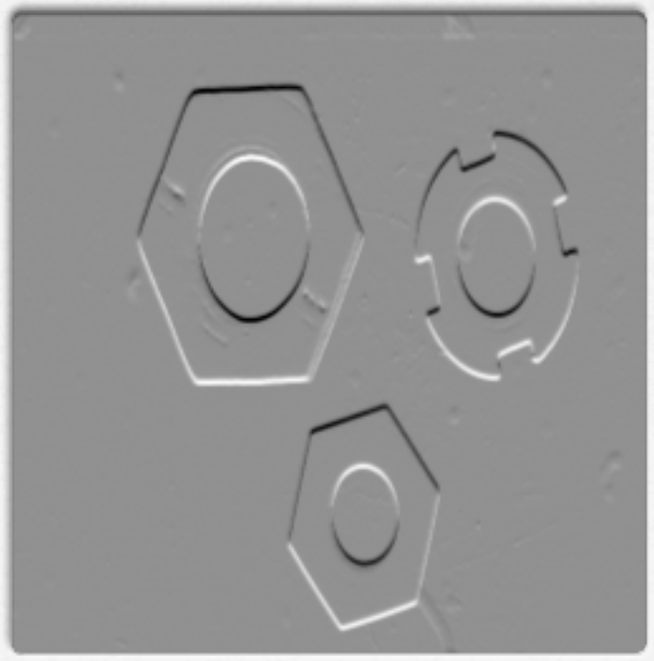
- ici on a appliqué un changement de dynamique linéaire, de façon à mettre les négatifs en foncé et les positifs en clair. Le 0 original est environ à 128

$$\frac{\partial f}{\partial m}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$\frac{\partial f}{\partial n}[m,n] \approx f[m,n] * \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$







Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n]$$

$$\approx \frac{f[m+2,n]-2f[m,n]+f[m-2,n]}{4}$$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx \frac{f[m+2,n]-2f[m,n]+f[m-2,n]}{4}$$

masque

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

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx \frac{f[m+2,n]-2f[m,n]+f[m-2,n]}{4}$$

masque

$$\longrightarrow \begin{bmatrix} 1/4 & 0 & -1/2 & 0 & 1/4 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial n^2}[m,n] \approx \frac{f[m,n+2]-2f[m,n]+f[m,n-2]}{4}$$

$$\longrightarrow \begin{bmatrix} 1/4 \\ 0 \\ -1/2 \\ 0 \\ 1/4 \end{bmatrix}$$

Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx \frac{f[m+2,n]-2f[m,n]+f[m-2,n]}{4}$$

masque

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

$$\frac{\partial^2 f}{\partial n^2}[m,n] \approx \frac{f[m,n+2]-2f[m,n]+f[m,n-2]}{4}$$

$$\begin{bmatrix} 1/4 \\ 0 \\ -1/2 \\ 0 \\ 1/4 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial m \partial n}[m,n] = \frac{\partial^2 f}{\partial n \partial m}[m,n]$$
$$\approx \frac{f[m+1,n+1]-f[m+1,n-1]-f[m-1,n+1]+f[m-1,n-1]}{4}$$

$$\begin{bmatrix} 1/4 & 0 & -1/4 \\ 0 & 0 & 0 \\ -1/4 & 0 & 1/4 \end{bmatrix}$$

Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx$$

$$f[m,n] * \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix}$$

Attention :

- ici on a appliqué un changement de dynamique de façon à mettre les négatifs en foncé et les positifs en clair. Le 0 est environ à 128

Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

Attention :

- ici on a appliqué un changement de dynamique de façon à mettre les négatifs en foncé et les positifs en clair. Le 0 est environ à 128

Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial n^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 \\ 0 \\ -2 \\ 0 \\ 1 \end{bmatrix}$$

Attention :

- ici on a appliqué un changement de dynamique de façon à mettre les négatifs en foncé et les positifs en clair. Le 0 est environ à 128

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- Dérivées d'ordre 2
  - ✓ 2D :

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix}$$

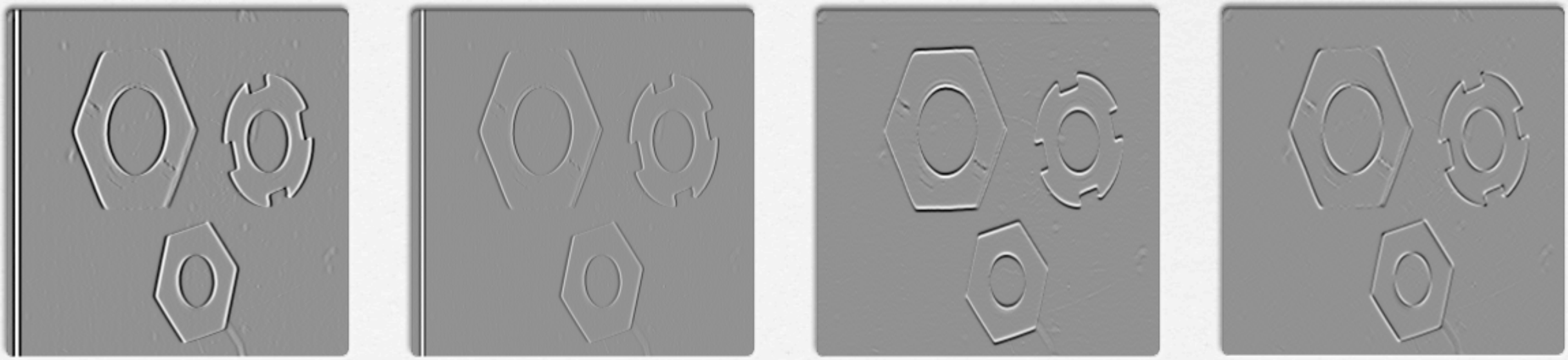
$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial n^2}[m,n] \approx f[m,n] * \begin{bmatrix} 1 \\ 0 \\ -2 \\ 0 \\ 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial m \partial n}[m,n] \approx f[m,n] * \begin{bmatrix} 1/4 & 0 & -1/4 \\ 0 & 0 & 0 \\ -1/4 & 0 & 1/4 \end{bmatrix}$$

Attention :

- ici on a appliqué un changement de dynamique de façon à mettre les négatifs en foncé et les positifs en clair. Le 0 est environ à 128



Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- *Problème* : les différences finies sont sensibles au bruit

Source : S. Narasimhan

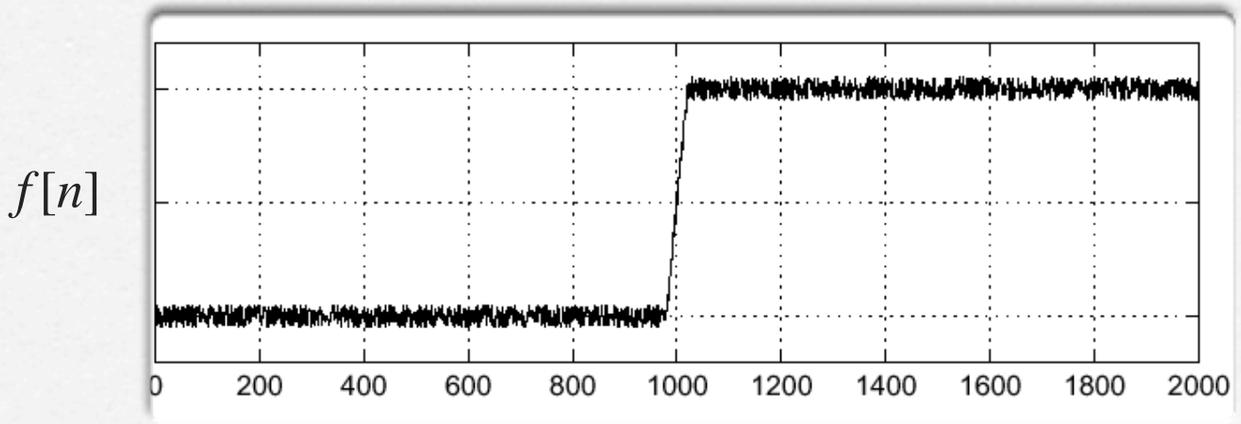
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- *Problème* : les différences finies sont sensibles au bruit
  - ✓ *Exemple* : simple contour 1D bruité



Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

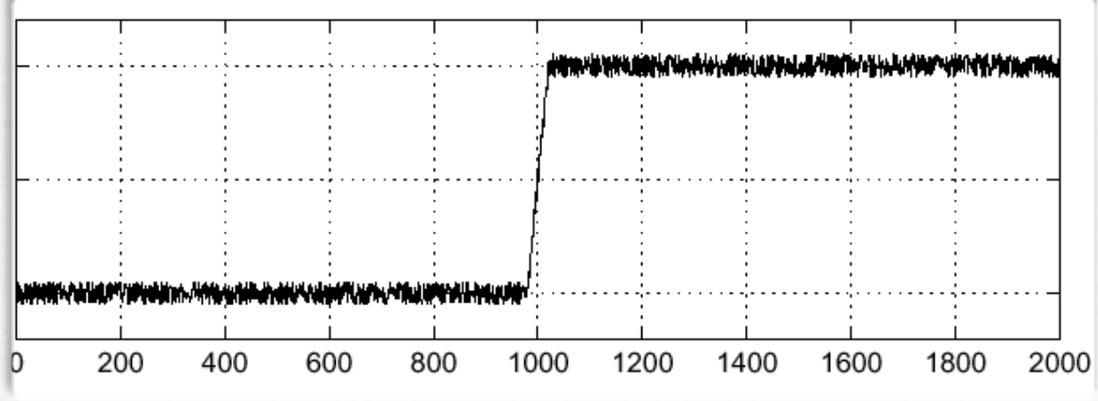
1. Calcul des dérivées d'image

1. Dérivées partielles

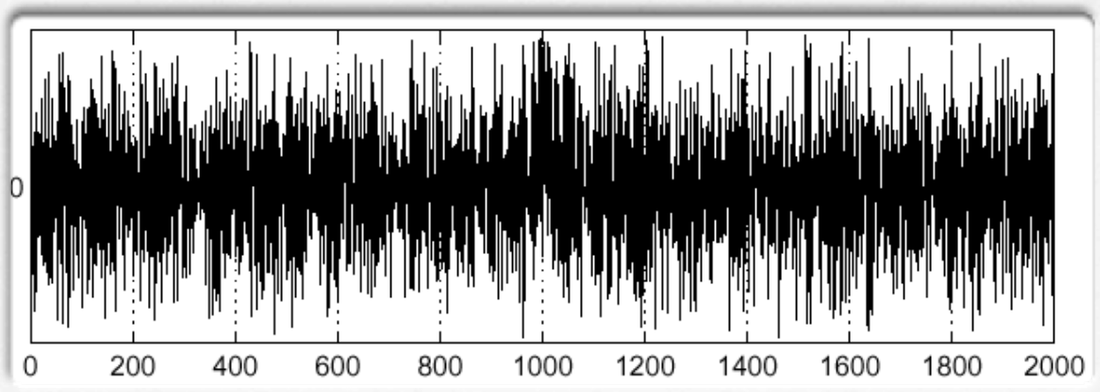
a) Différences finies

- *Problème* : les différences finies sont sensibles au bruit
  - ✓ *Exemple* : simple contour 1D bruité

$f[n]$



$f'[n]$



Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

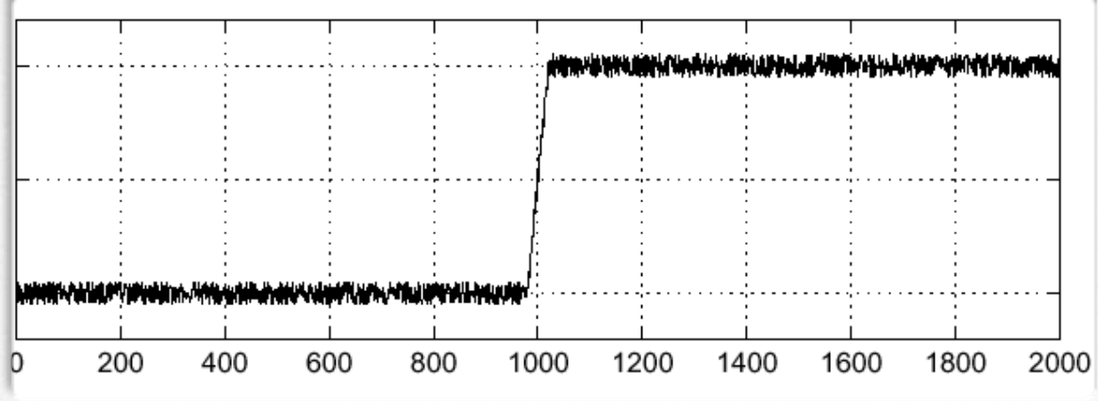
1. Calcul des dérivées d'image

1. Dérivées partielles

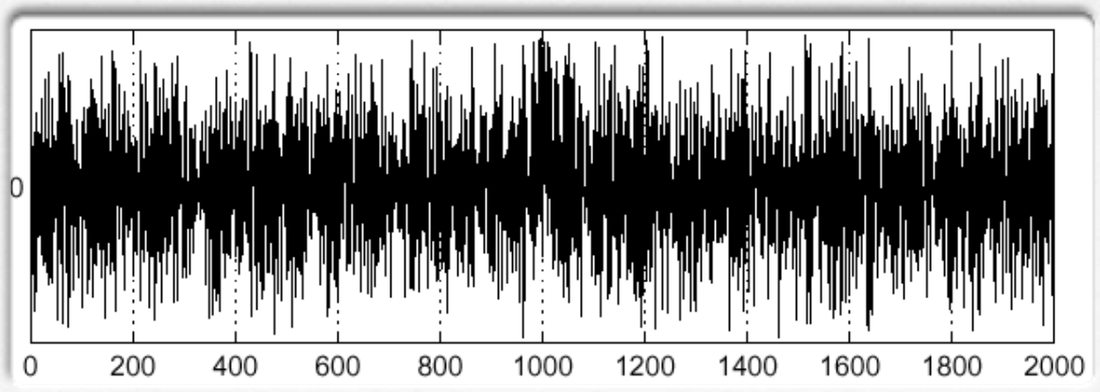
a) Différences finies

- *Problème* : les différences finies sont sensibles au bruit
  - ✓ *Exemple* : simple contour 1D bruité

$f[n]$



$f'[n]$



Où est le contour?

Source : S. Narasimhan

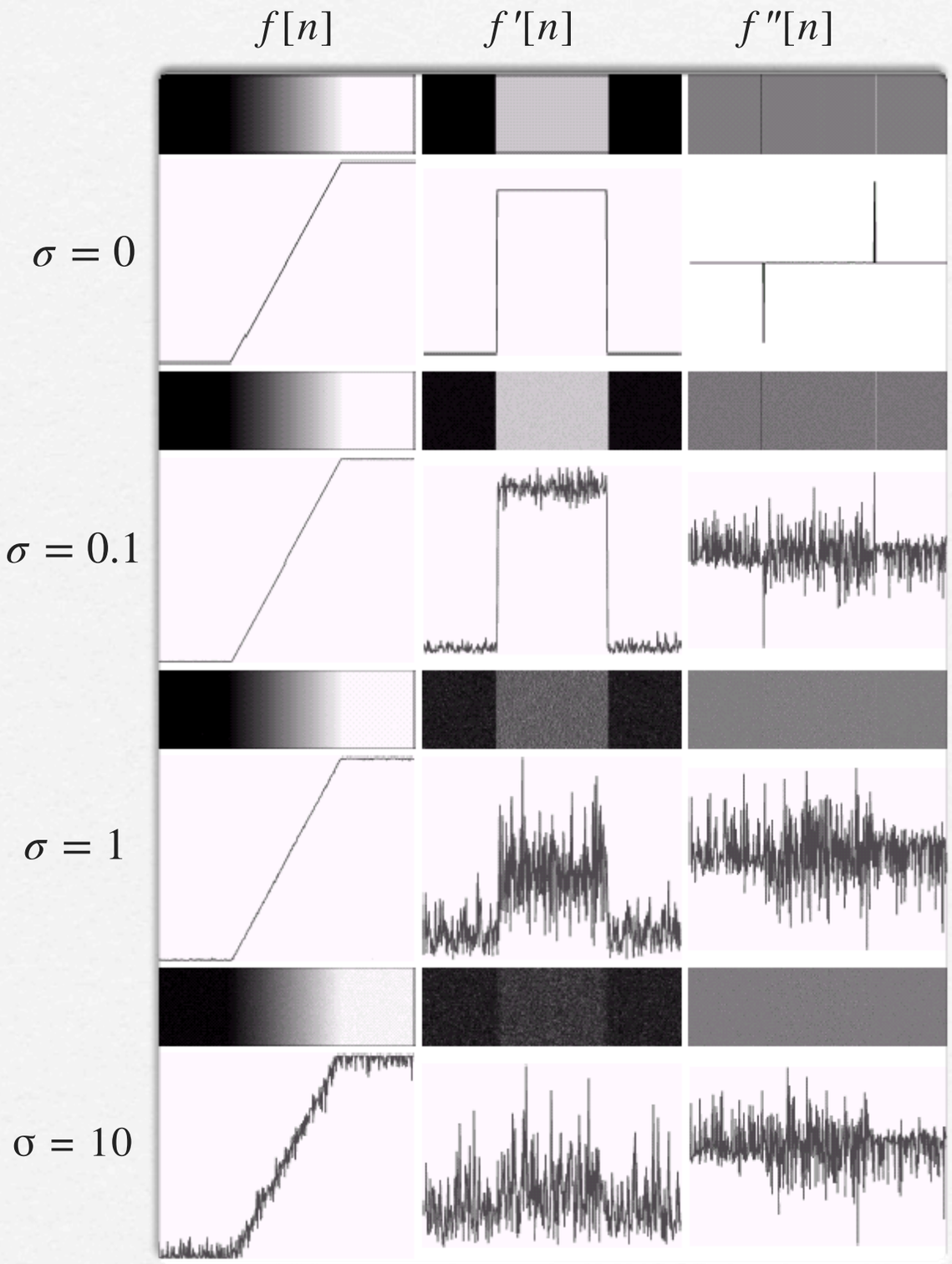
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

- *Problème* : les différences finies sont sensibles au bruit
  - ✓ plus on dérive, plus l'amplitude du bruit augmente
  - ✓ l'extraction des caractéristiques devient difficile



Source : DIP2

1. Calcul des dérivées d'image

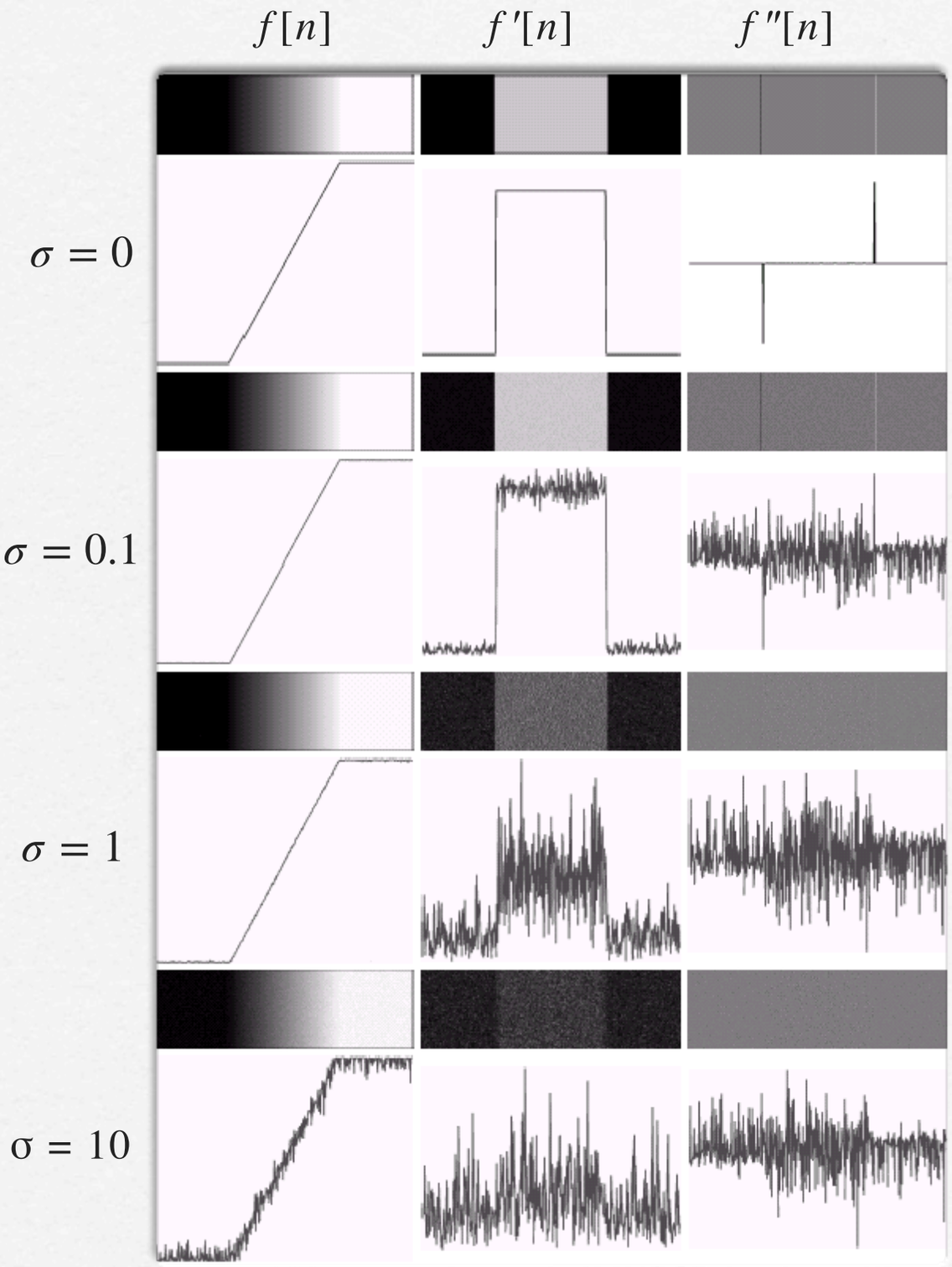
1. Dérivées partielles

a) Différences finies

- *Problème* : les différences finies sont sensibles au bruit
  - ✓ plus on dérive, plus l'amplitude du bruit augmente
  - ✓ l'extraction des caractéristiques devient difficile

➡ *Exemple* :

$$f[n] = u[n] + b[n], \qquad \text{avec } \mathrm{P}(b[n]) = g_\sigma(b)$$



Source : DIP2

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

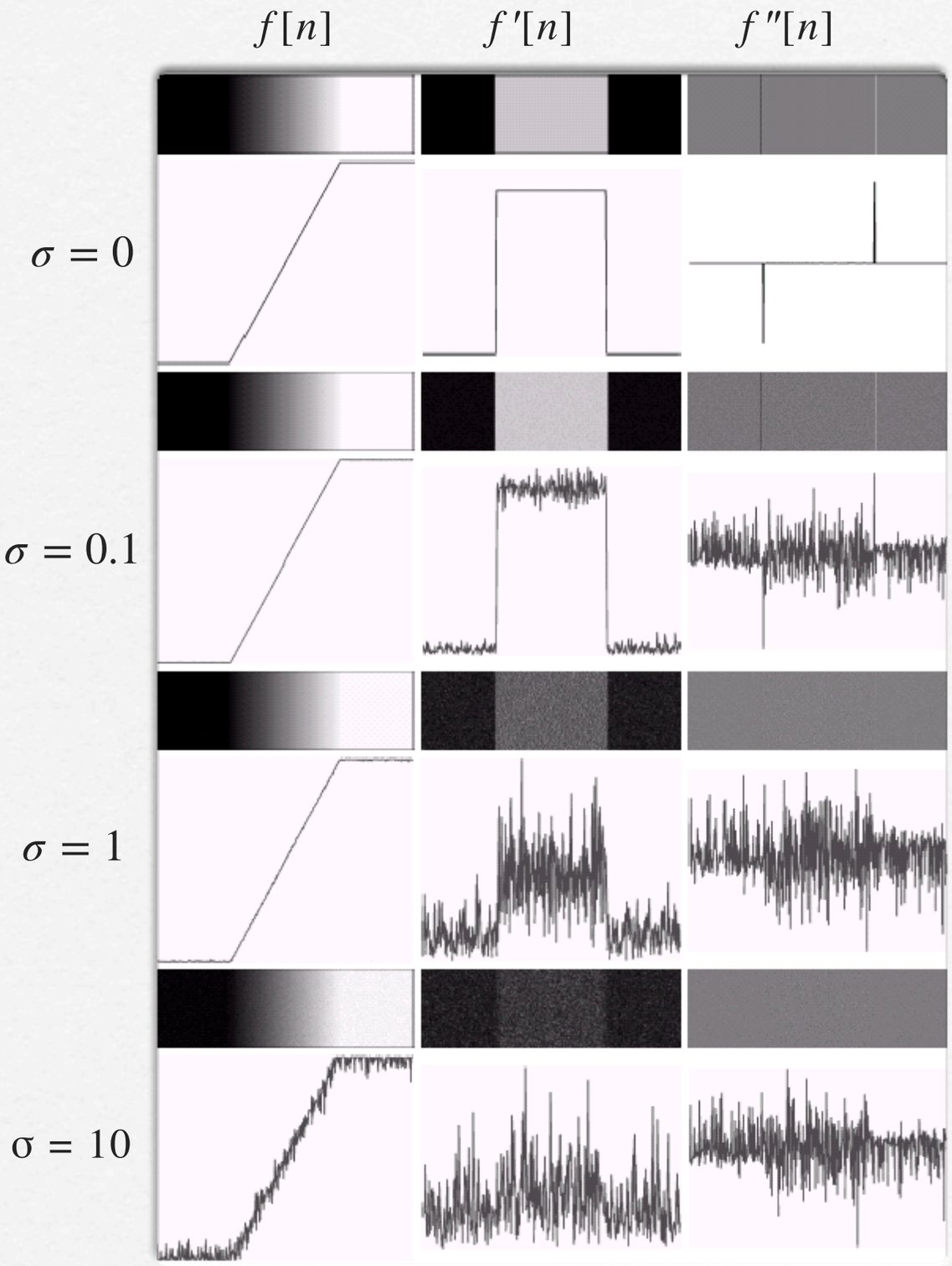
- *Problème* : les différences finies sont sensibles au bruit
  - ✓ plus on dérive, plus l'amplitude du bruit augmente
  - ✓ l'extraction des caractéristiques devient difficile

➡ *Exemple* :

$$f[n] = u[n] + b[n], \qquad \text{avec } \mathrm{P}(b[n]) = g_\sigma(b)$$

Solutions :

- Lisser les dérivées;



Source : DIP2

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

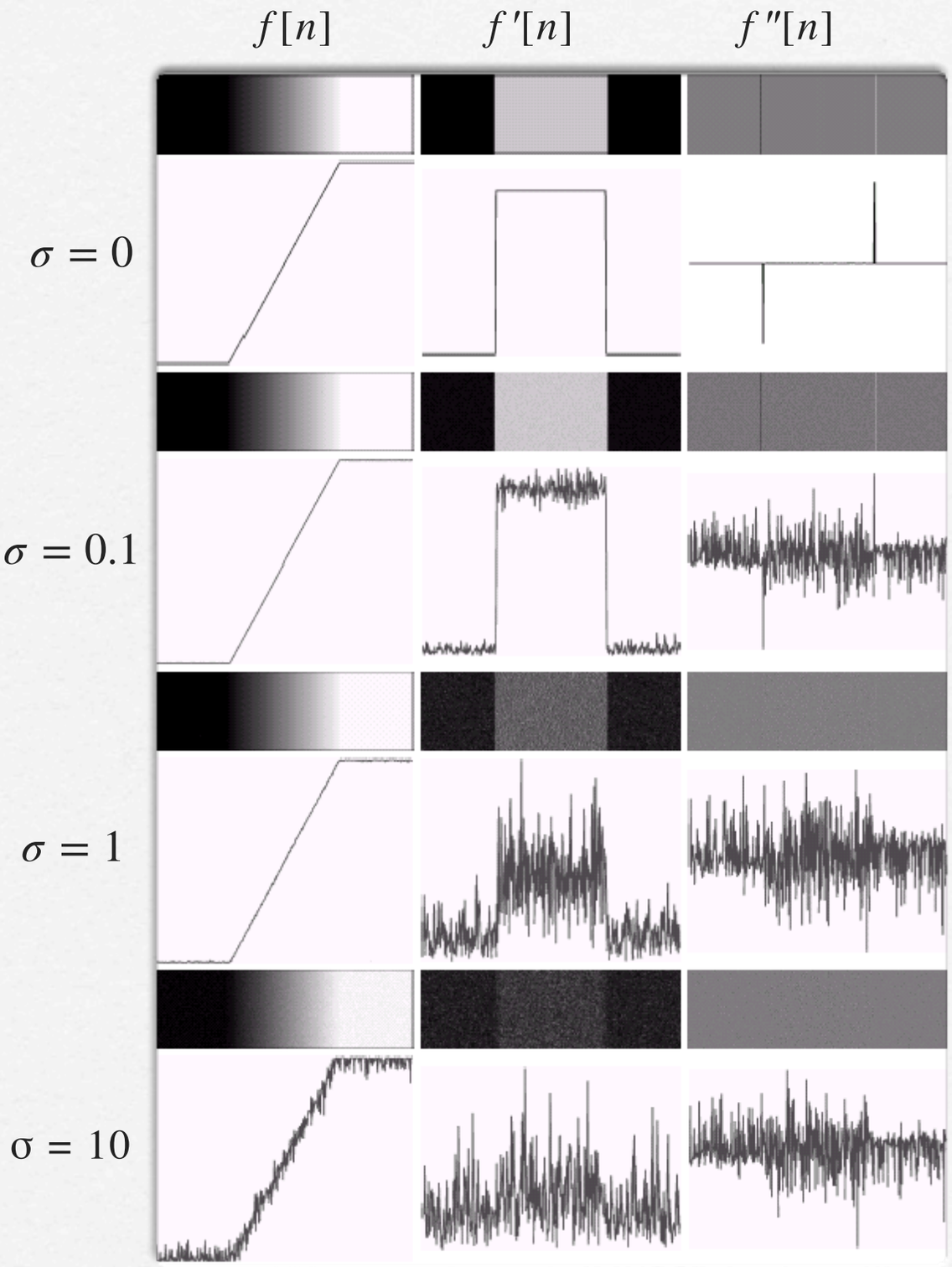
- *Problème* : les différences finies sont sensibles au bruit
  - ✓ plus on dérive, plus l'amplitude du bruit augmente
  - ✓ l'extraction des caractéristiques devient difficile

➡ *Exemple* :

$$f[n] = u[n] + b[n], \qquad \text{avec } \mathrm{P}(b[n]) = g_\sigma(b)$$

Solutions :

- Lisser les dérivées;
- Lisser les images avant;



Source : DIP2

1. Calcul des dérivées d'image

1. Dérivées partielles

a) Différences finies

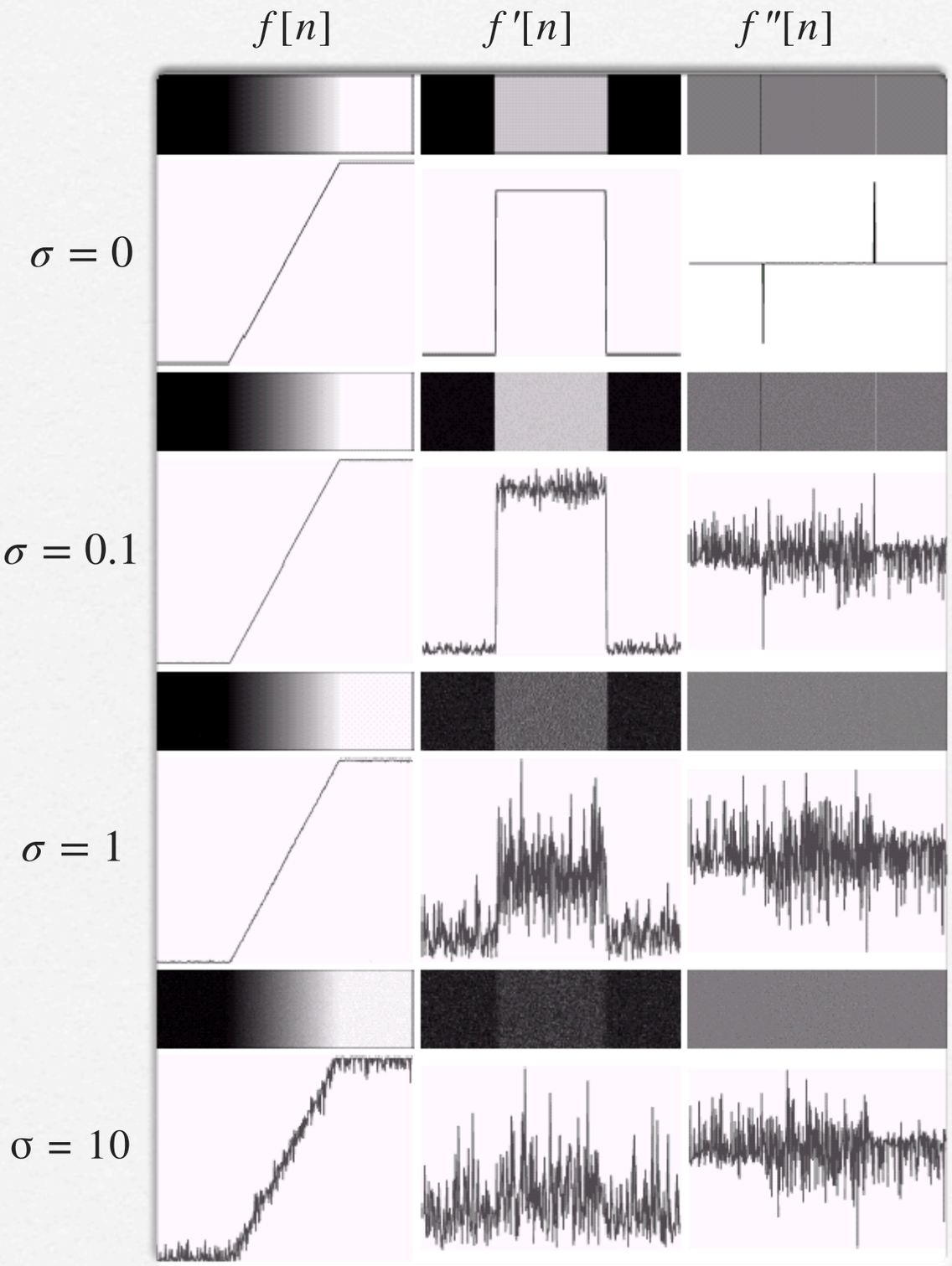
- *Problème* : les différences finies sont sensibles au bruit
  - ✓ plus on dérive, plus l'amplitude du bruit augmente
  - ✓ l'extraction des caractéristiques devient difficile

➡ *Exemple* :

$$f[n] = u[n] + b[n], \qquad \text{avec } \mathrm{P}(b[n]) = g_\sigma(b)$$

Solutions :

- Lisser les dérivées;
- Lisser les images avant;
- ou….



Source : DIP2

©Marie-Flavie Auclair-Fortier

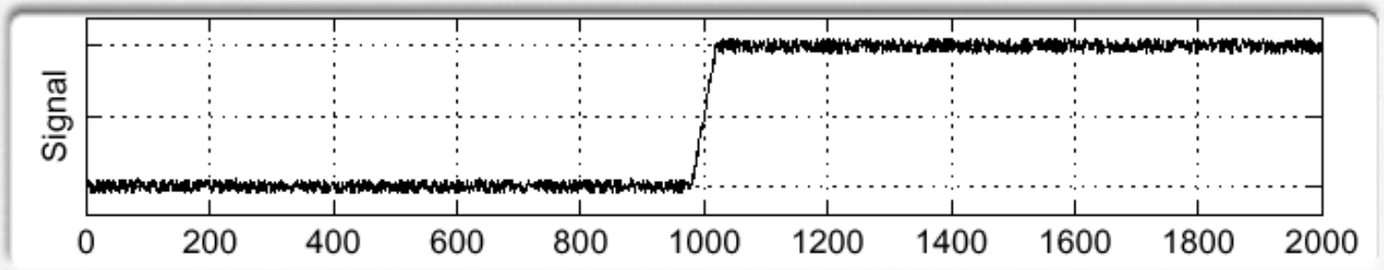
1. Calcul des dérivées d'image

1. Dérivées partielles

Trouver le max dans
$$\frac{\partial}{\partial n}\left(f * h_b\right)[n]$$

b) À base de lissage

$f[n]$



Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

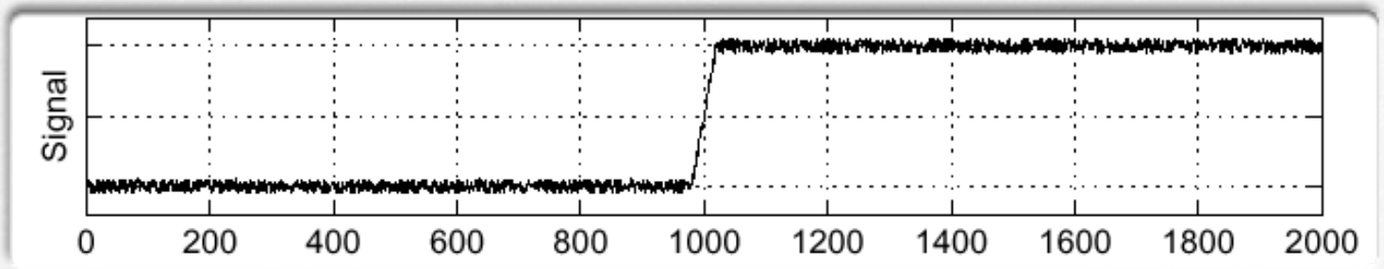
1. Calcul des dérivées d'image

1. Dérivées partielles

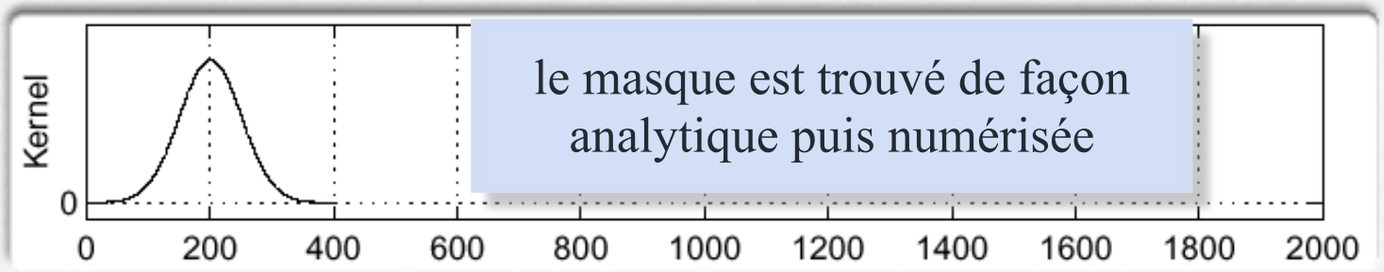
Trouver le max dans $\frac{\partial}{\partial n}\left(f * h_b\right)[n]$

b) À base de lissage

$f[n]$



$h_b[n]$



le masque est trouvé de façon analytique puis numérisée

Source : S. Narasimhan

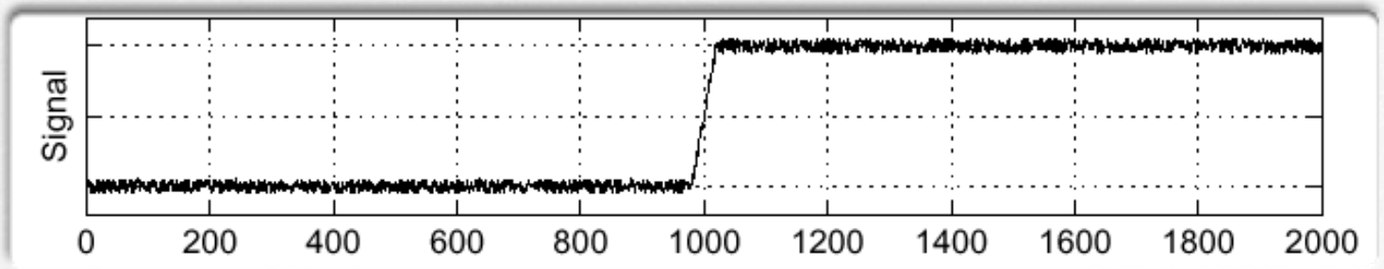
1. Calcul des dérivées d'image

1. Dérivées partielles

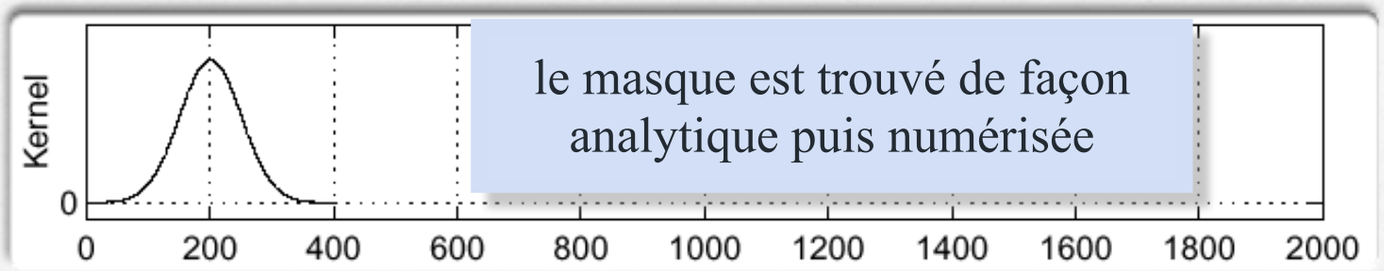
Trouver le max dans
$$\frac{\partial}{\partial n}\left(f * h_b\right)[n]$$

b) À base de lissage

$f[n]$

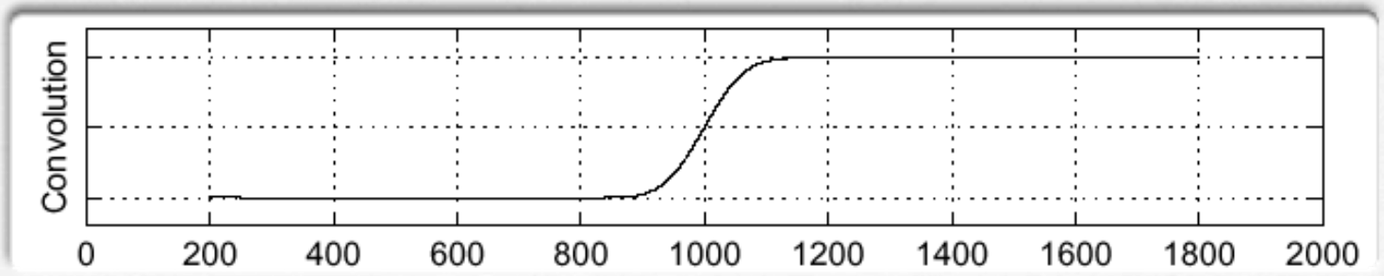


$h_b[n]$



le masque est trouvé de façon analytique puis numérisée

$(f * h_b)[n]$



Source : S. Narasimhan

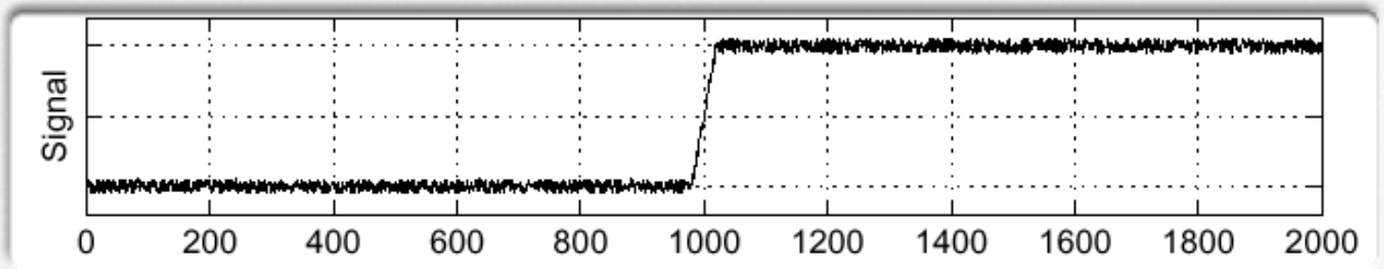
1. Calcul des dérivées d'image

1. Dérivées partielles

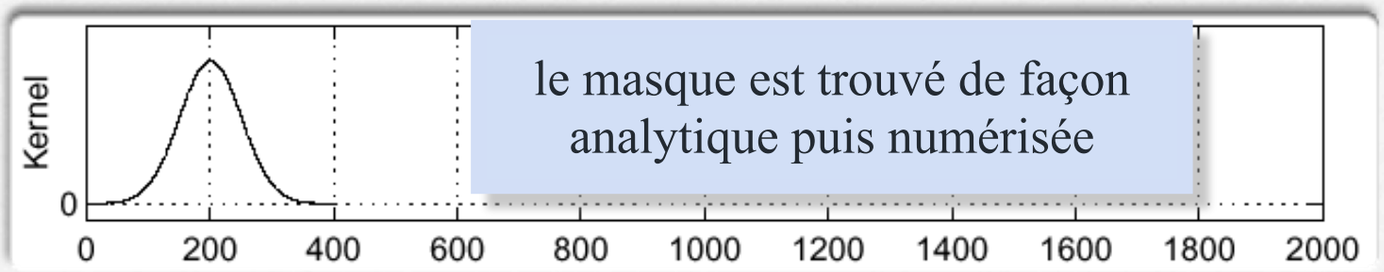
Trouver le max dans $\frac{\partial}{\partial n}\left(f * h_b\right)[n]$

b) À base de lissage

$f[n]$

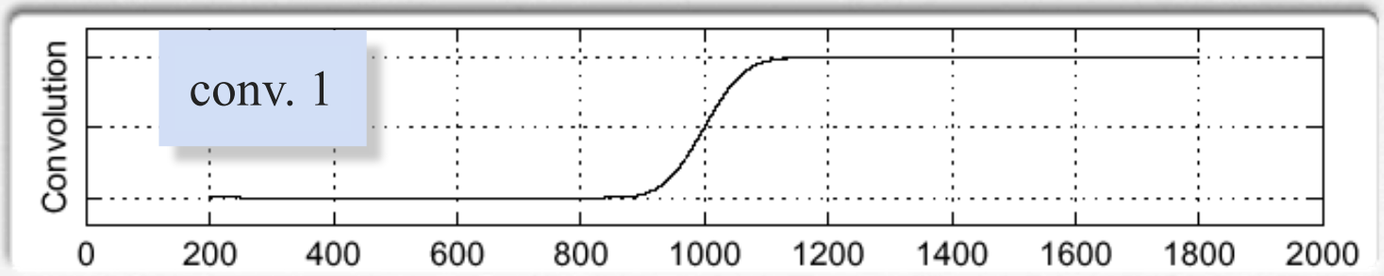


$h_b[n]$



le masque est trouvé de façon analytique puis numérisée

$\left(f * h_b\right)[n]$



conv. 1

Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

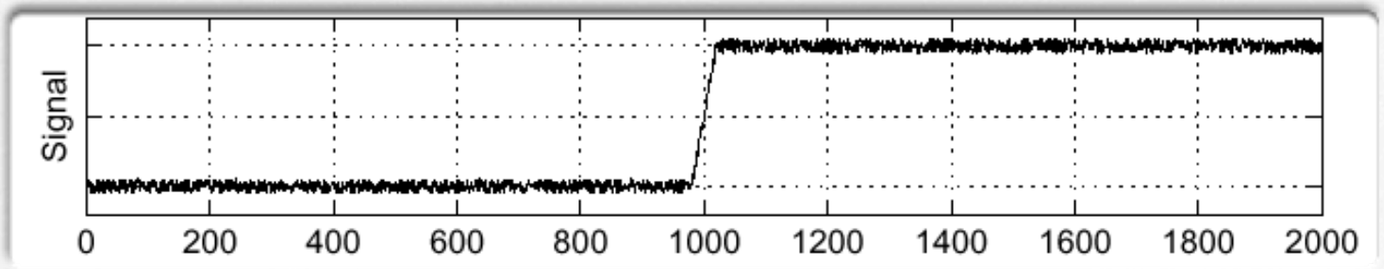
1. Calcul des dérivées d'image

1. Dérivées partielles

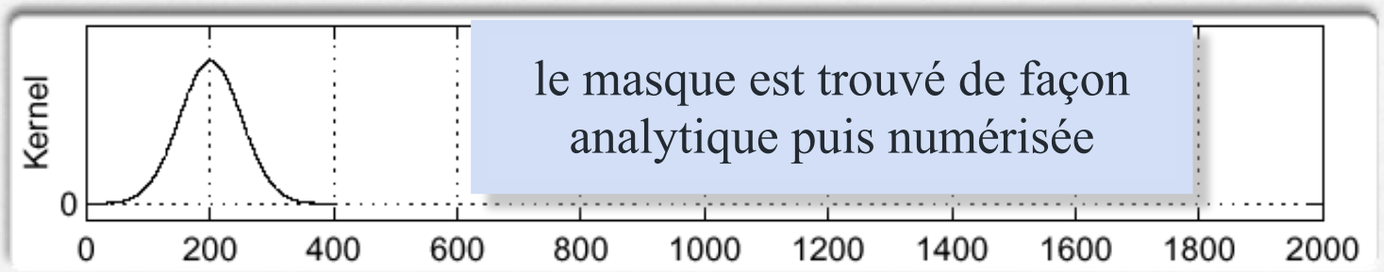
Trouver le max dans $\frac{\partial}{\partial n}\left(f * h_b\right)[n]$

b) À base de lissage

$f[n]$

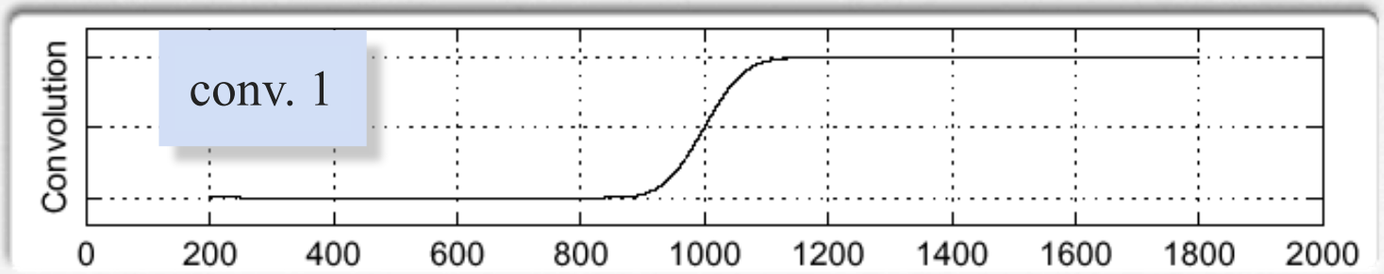


$h_b[n]$



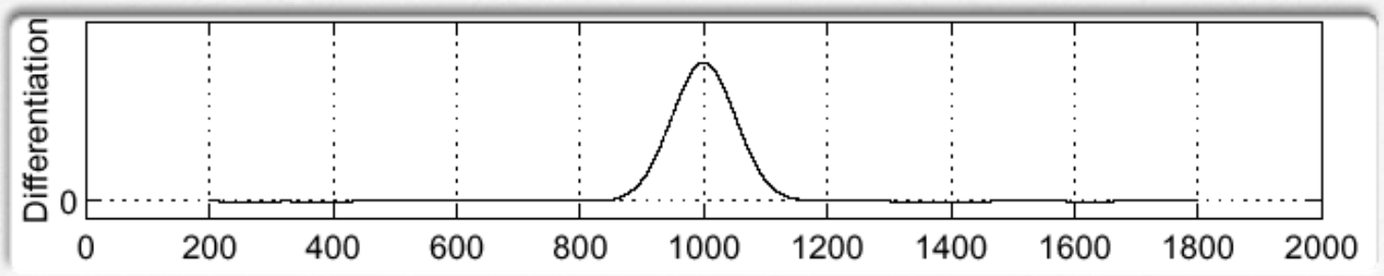
le masque est trouvé de façon analytique puis numérisée

$\left(f * h_b\right)[n]$



conv. 1

$\frac{\partial\left(f * h_b\right)}{\partial n}[n]$



Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

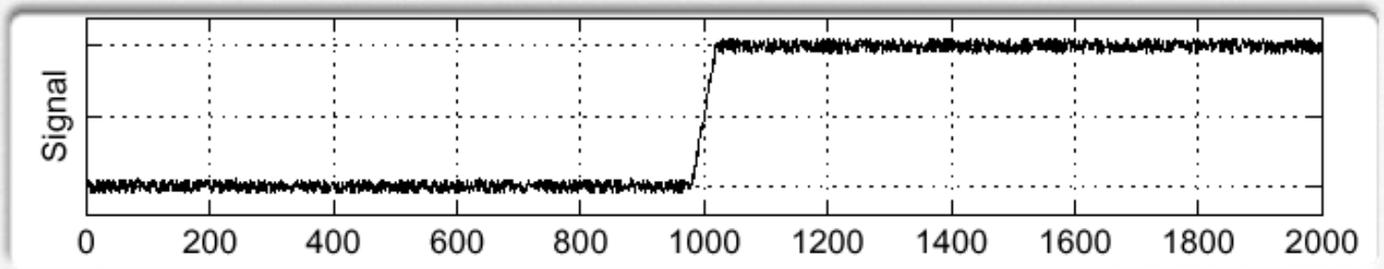
1. Calcul des dérivées d'image

1. Dérivées partielles

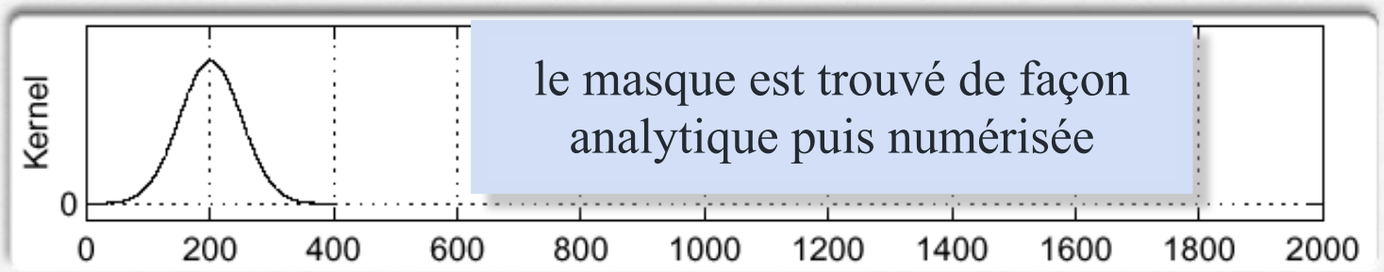
Trouver le max dans $\frac{\partial}{\partial n}\left(f * h_b\right)[n]$

b) À base de lissage

$f[n]$

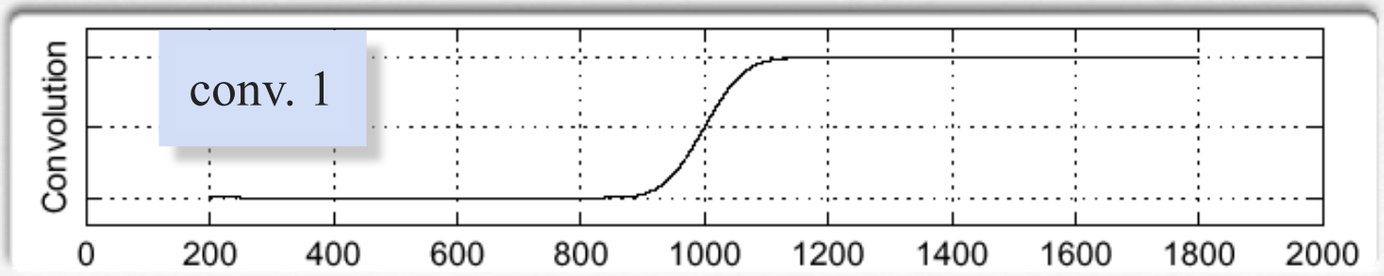


$h_b[n]$

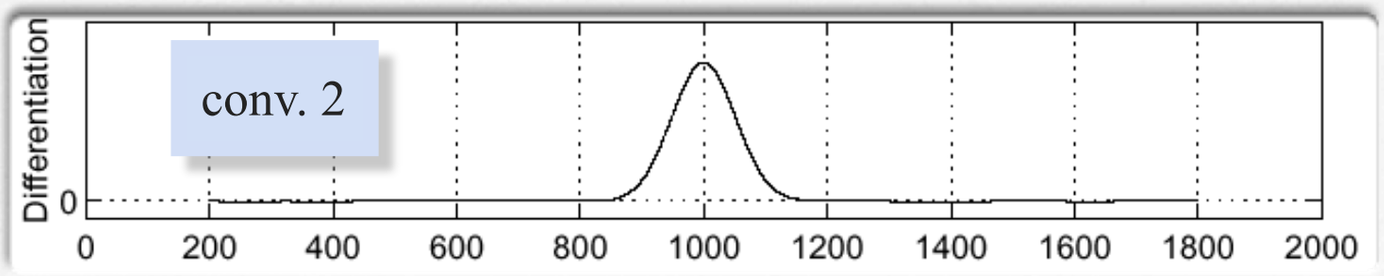


le masque est trouvé de façon analytique puis numérisée

$\left(f * h_b\right)[n]$



$\frac{\partial\left(f * h_b\right)}{\partial n}[n]$



Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

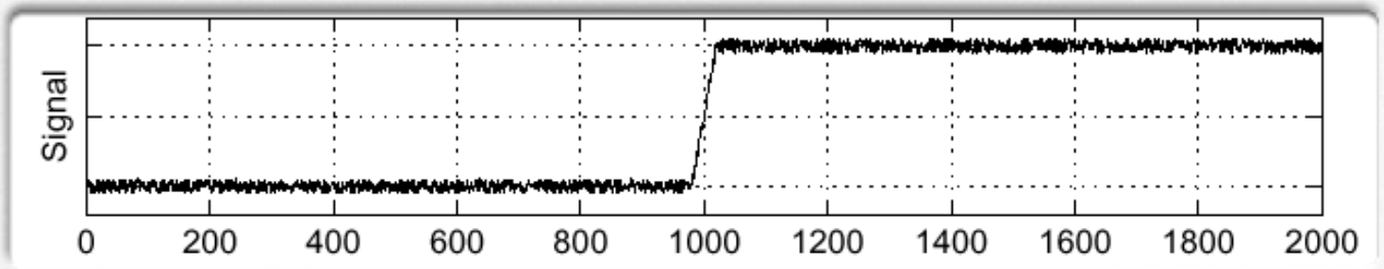
1. Calcul des dérivées d'image

1. Dérivées partielles

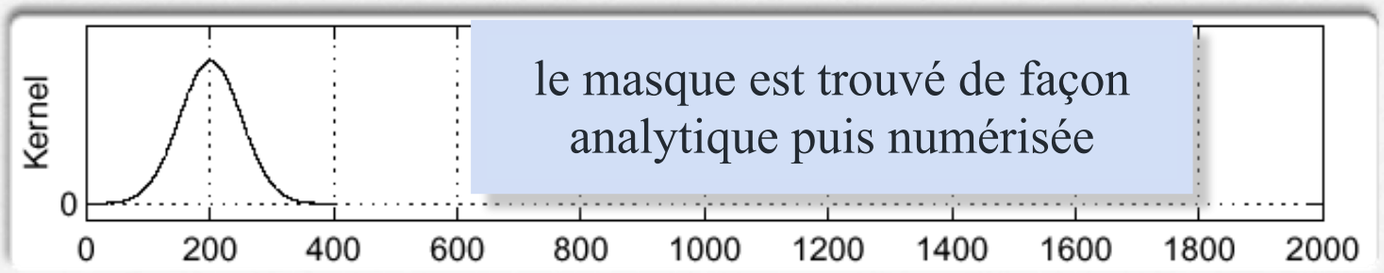
Trouver le max dans $\frac{\partial}{\partial n}\left(f * h_b\right)[n]$

b) À base de lissage

$f[n]$

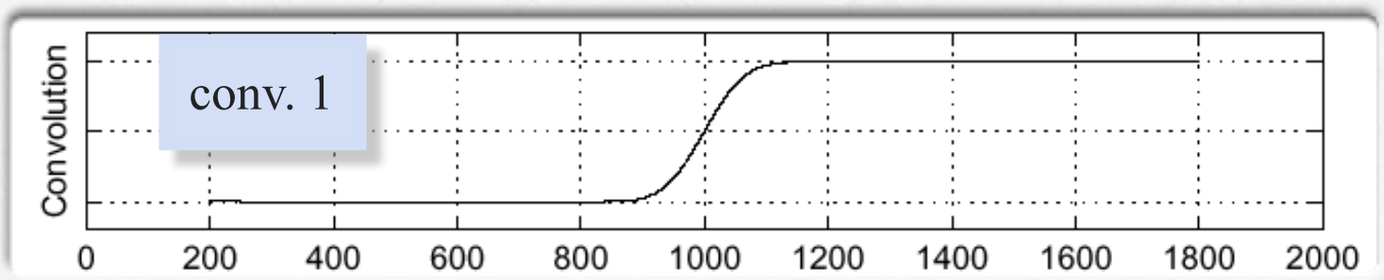


$h_b[n]$

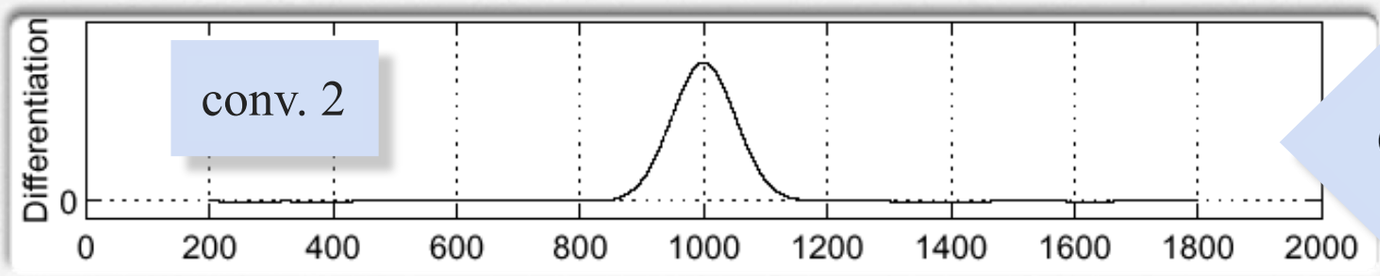


le masque est trouvé de façon analytique puis numérisée

$\left(f * h_b\right)[n]$



$\frac{\partial\left(f * h_b\right)}{\partial n}[n]$



Où est le contour?

Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

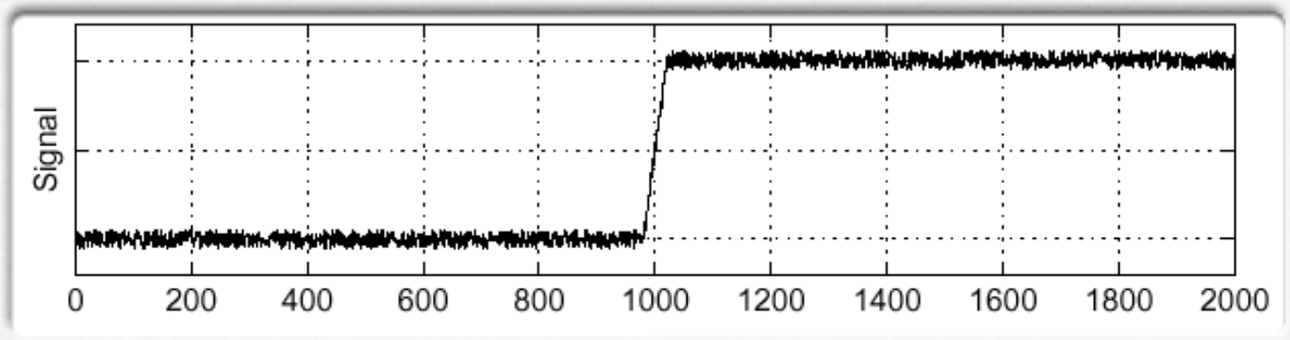
b) À base de lissage

- Sauve une convolution

Rappel des propriétés de la convolution

- $\frac{\partial}{\partial x}(f * h_b)(x) = \left(f * \frac{\partial h_b}{\partial x}\right)(x)$

$f[n]$



Trouver le max dans $\left(f * \frac{\partial h_b}{\partial n}\right)[n]$

Source : S. Narasimhan

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

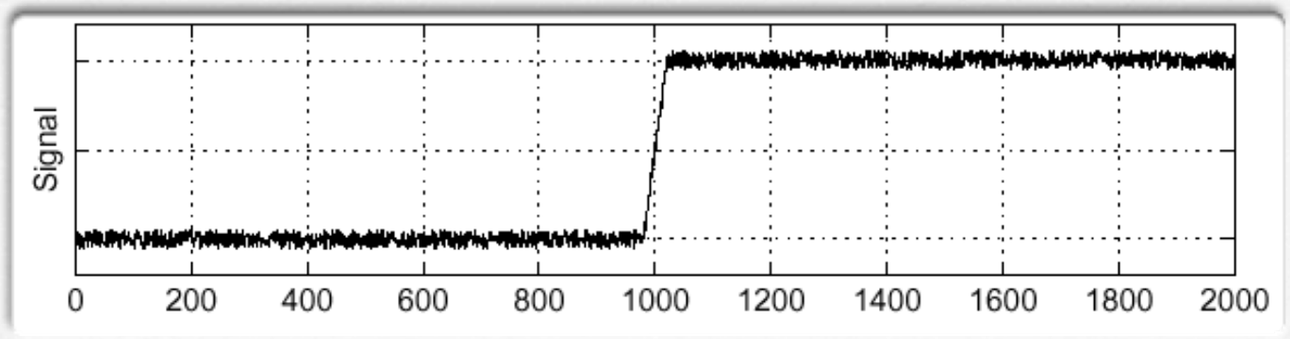
b) À base de lissage

- Sauve une convolution

Rappel des propriétés de la convolution

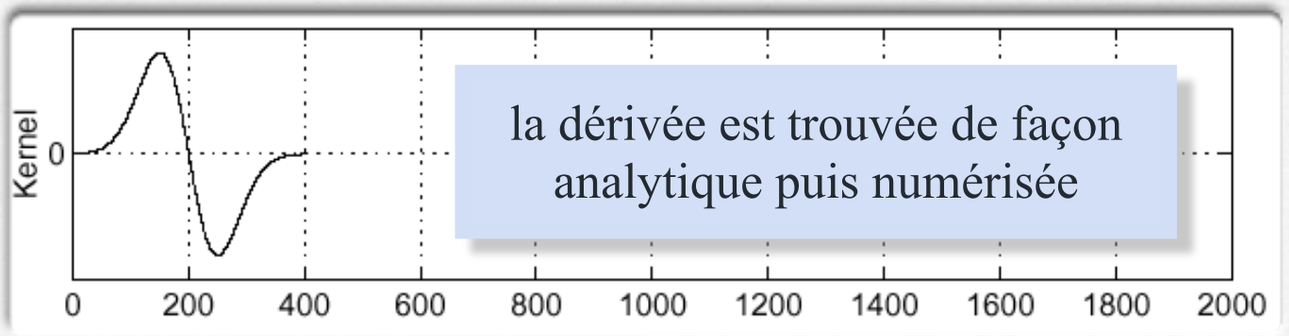
- $\frac{\partial}{\partial x}(f * h_b)(x) = \left(f * \frac{\partial h_b}{\partial x}\right)(x)$

$f[n]$



Trouver le max dans $\left(f * \frac{\partial h_b}{\partial n}\right)[n]$

$\frac{\partial h_b}{\partial n}[n]$



la dérivée est trouvée de façon analytique puis numérisée

Source : S. Narasimhan

1. Calcul des dérivées d'image

1. Dérivées partielles

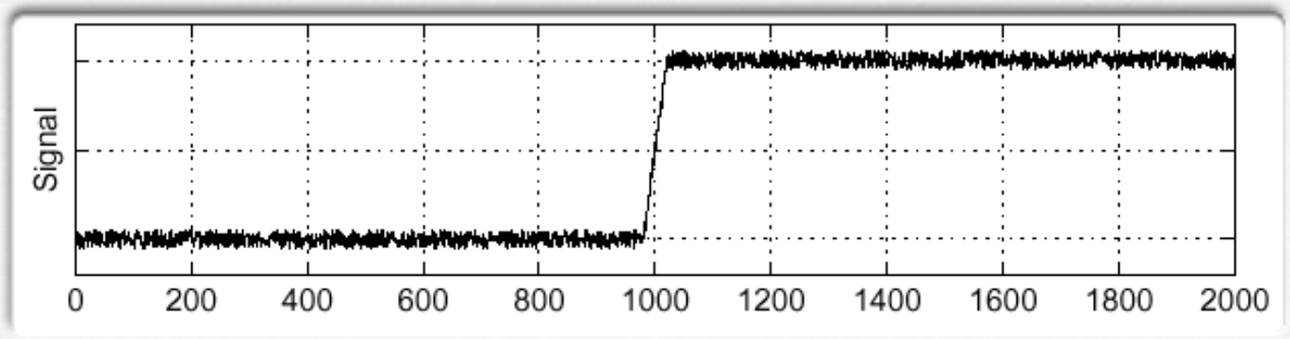
b) À base de lissage

- Sauve une convolution

Rappel des propriétés de la convolution

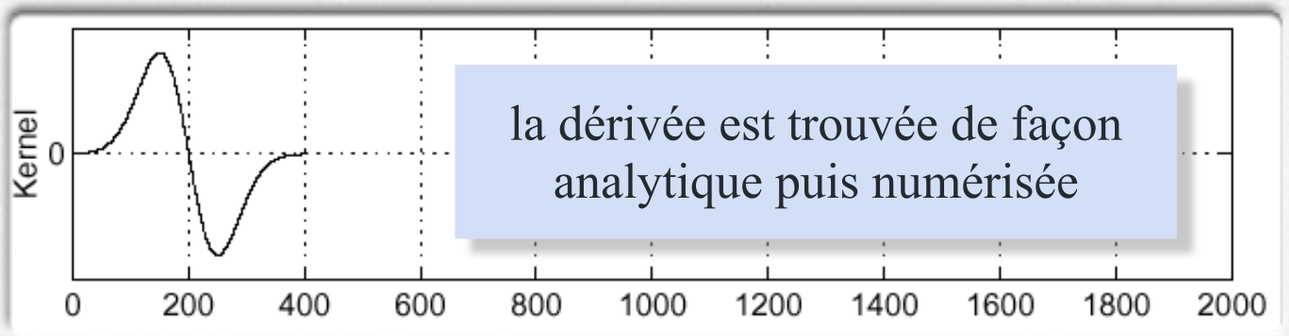
- $\frac{\partial}{\partial x}(f * h_b)(x) = \left(f * \frac{\partial h_b}{\partial x}\right)(x)$

$f[n]$



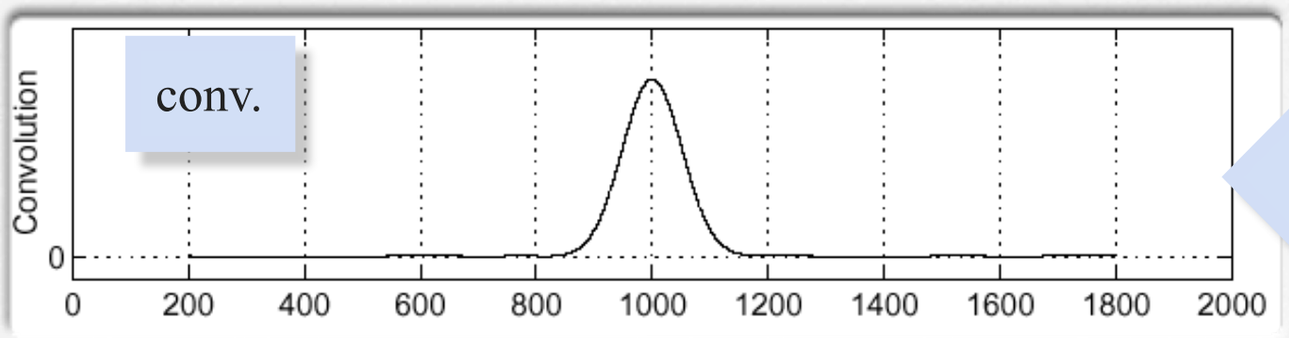
Trouver le max dans $\left(f * \frac{\partial h_b}{\partial n}\right)[n]$

$\frac{\partial h_b}{\partial n}[n]$



la dérivée est trouvée de façon analytique puis numérisée

$\left(f * \frac{\partial h_b}{\partial n}\right)[n]$



Où est le contour?

Source : S. Narasimhan

1. Calcul des dérivées d'image

1. Dérivées partielles

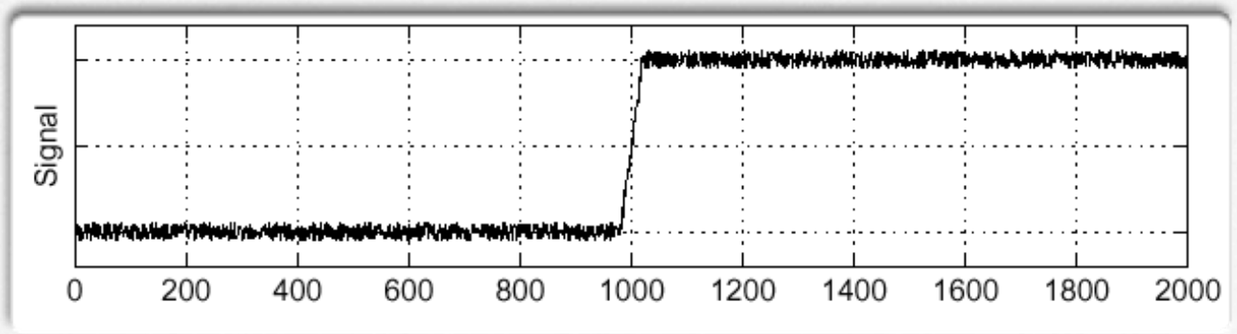
b) À base de lissage

- Ordre 2

Rappel des propriétés de la convolution

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

$f[n]$



Trouver le passage par 0 dans $\left(f * \frac{\partial^2 h_b}{\partial n^2}\right)[n]$

Source : S. Narasimhan

1. Calcul des dérivées d'image

1. Dérivées partielles

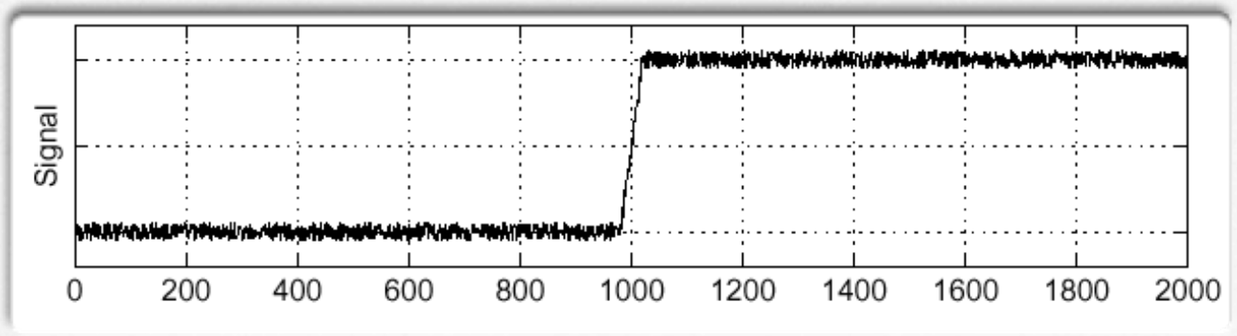
b) À base de lissage

- Ordre 2

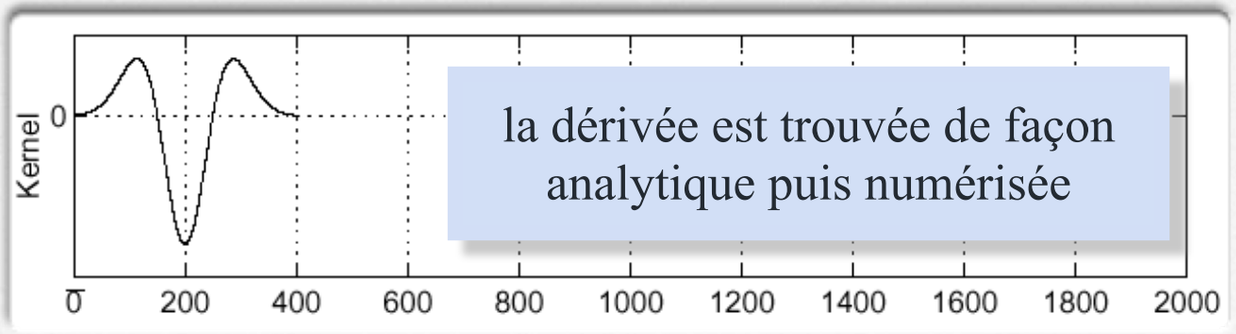
Rappel des propriétés de la convolution

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

$f[n]$



$\frac{\partial^2 h_b}{\partial n^2}[n]$



la dérivée est trouvée de façon analytique puis numérisée

Trouver le passage par 0 dans $\left(f * \frac{\partial^2 h_b}{\partial n^2}\right)[n]$

Source : S. Narasimhan

1. Calcul des dérivées d'image

1. Dérivées partielles

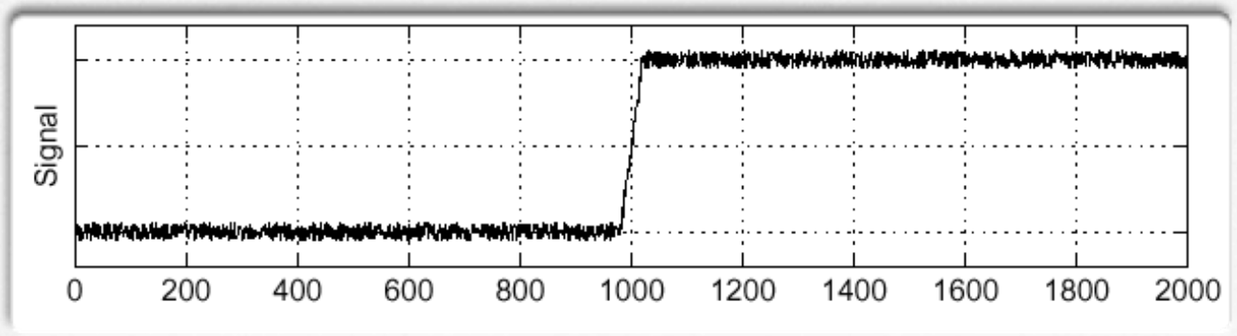
b) À base de lissage

- Ordre 2

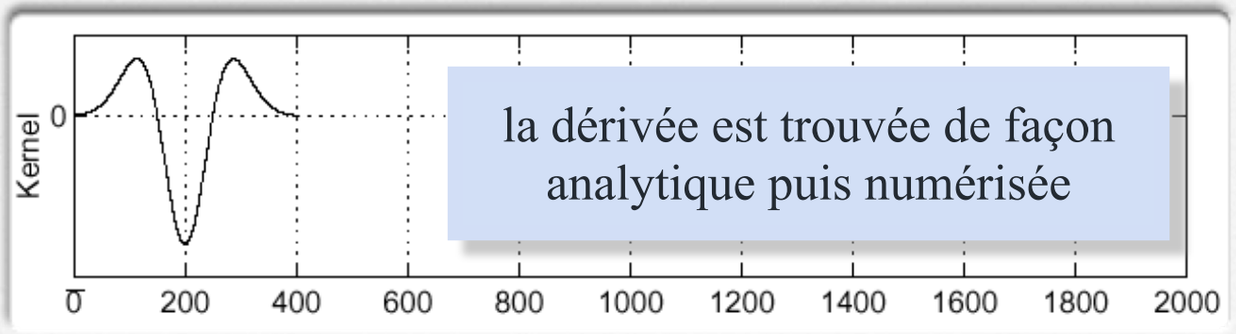
Rappel des propriétés de la convolution

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

$f[n]$

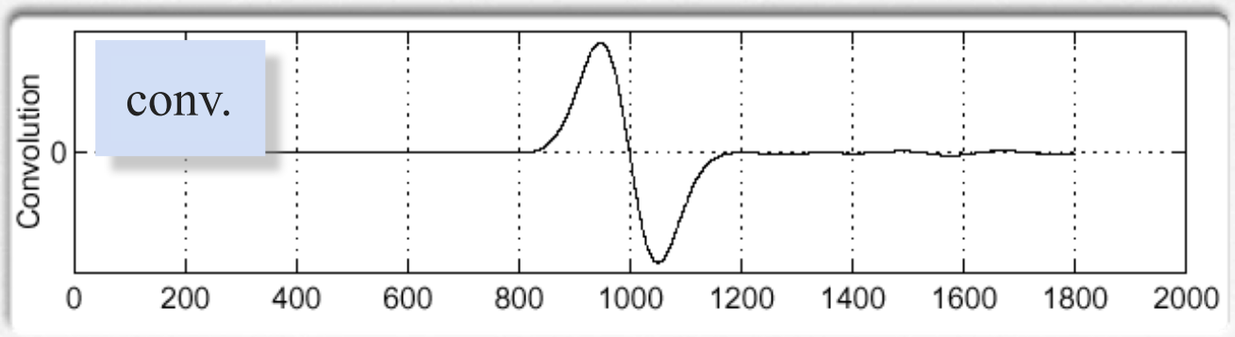


$\frac{\partial^2 h_b}{\partial n^2}[n]$



la dérivée est trouvée de façon analytique puis numérisée

$\left(f * \frac{\partial^2 h_b}{\partial n^2}\right)[n]$



Trouver le passage par 0 dans $\left(f * \frac{\partial^2 h_b}{\partial n^2}\right)[n]$

Source : S. Narasimhan

1. Calcul des dérivées d'image

1. Dérivées partielles

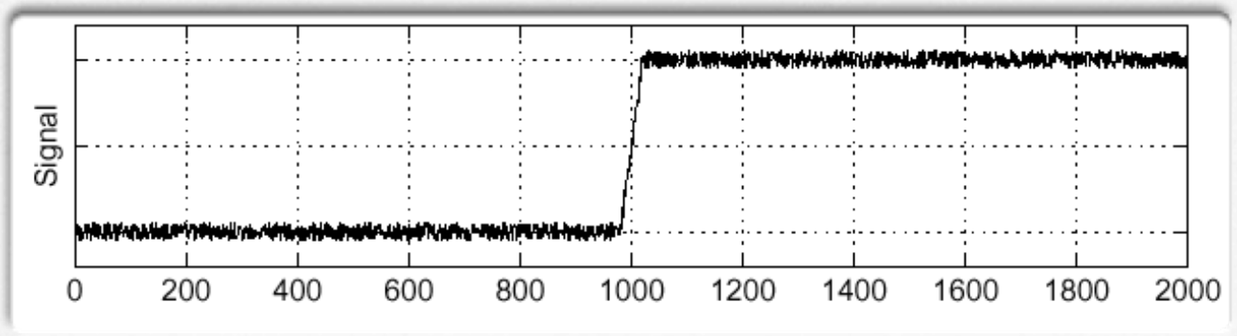
b) À base de lissage

- Ordre 2

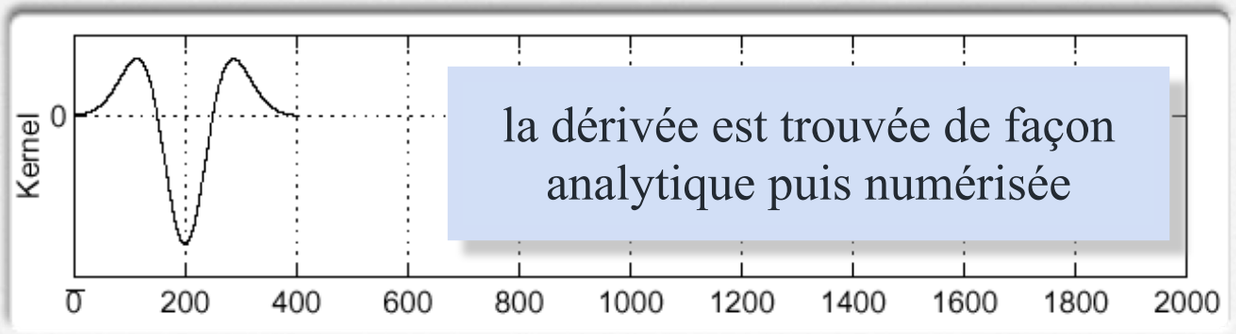
Rappel des propriétés de la convolution

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

$f[n]$



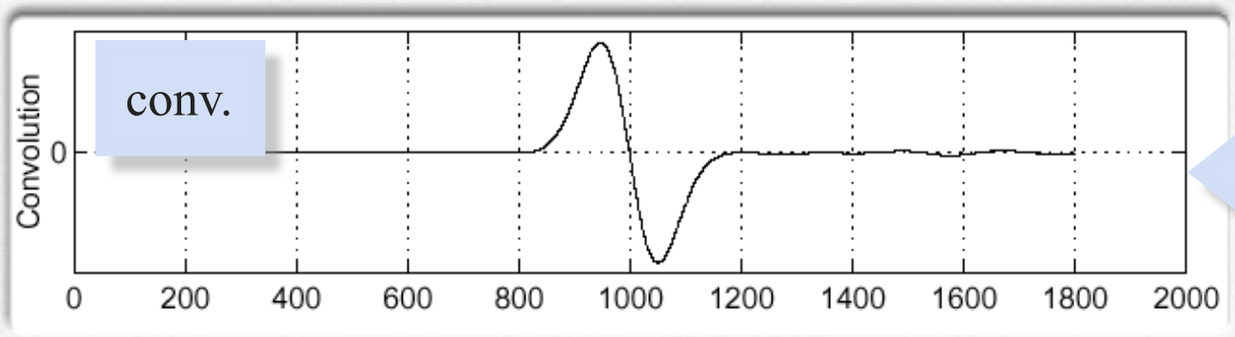
$\frac{\partial^2 h_b}{\partial n^2}[n]$



la dérivée est trouvée de façon analytique puis numérisée

Trouver le passage par 0 dans $\left(f * \frac{\partial^2 h_b}{\partial n^2}\right)[n]$

$\left(f * \frac{\partial^2 h_b}{\partial n^2}\right)[n]$



Où est le contour?

Source : S. Narasimhan

1. Calcul des dérivées d'image

1. Dérivées partielles

b) À base de lissage

- 2D
  - ✓ ordre 1

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

b) À base de lissage

- 2D
  - ✓ ordre 1

$$\frac{\partial f}{\partial m}[m,n] \approx \left(f * \frac{\partial h_b}{\partial m} \right)[m,n] \quad \text{et} \quad \frac{\partial f}{\partial n}[m,n] \approx \left(f * \frac{\partial h_b}{\partial n} \right)[m,n]$$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

b) À base de lissage

- 2D
  - ✓ ordre 1

$$\frac{\partial f}{\partial m}[m,n] \approx \left(f * \frac{\partial h_b}{\partial m} \right)[m,n] \quad \text{et} \quad \frac{\partial f}{\partial n}[m,n] \approx \left(f * \frac{\partial h_b}{\partial n} \right)[m,n]$$

  - ✓ ordre 2

Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

b) À base de lissage

- 2D
  - ✓ ordre 1

$$\frac{\partial f}{\partial m}[m,n] \approx \left(f * \frac{\partial h_b}{\partial m} \right)[m,n] \quad \text{et} \quad \frac{\partial f}{\partial n}[m,n] \approx \left(f * \frac{\partial h_b}{\partial n} \right)[m,n]$$

  - ✓ ordre 2

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx \left(f * \frac{\partial^2 h_b}{\partial m^2} \right)[m,n] \qquad \frac{\partial^2 f}{\partial n^2}[m,n] \approx \left(f * \frac{\partial^2 h_b}{\partial n^2} \right)[m,n] \qquad \frac{\partial^2 f}{\partial m \partial n}[m,n] \approx \left(f * \frac{\partial^2 h_b}{\partial m \partial n} \right)[m,n]$$

Source :

1. Calcul des dérivées d'image

1. Dérivées partielles

b) À base de lissage

- 2D
  - ✓ ordre 1

$$\frac{\partial f}{\partial m}[m,n] \approx \left(f * \frac{\partial h_b}{\partial m} \right)[m,n] \quad \text{et} \quad \frac{\partial f}{\partial n}[m,n] \approx \left(f * \frac{\partial h_b}{\partial n} \right)[m,n]$$

On peut utiliser le masque passe-bas qu'on veut

  - ✓ ordre 2

$$\frac{\partial^2 f}{\partial m^2}[m,n] \approx \left(f * \frac{\partial^2 h_b}{\partial m^2} \right)[m,n] \qquad \frac{\partial^2 f}{\partial n^2}[m,n] \approx \left(f * \frac{\partial^2 h_b}{\partial n^2} \right)[m,n] \qquad \frac{\partial^2 f}{\partial m \partial n}[m,n] \approx \left(f * \frac{\partial^2 h_b}{\partial m \partial n} \right)[m,n]$$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

i) Prewitt d'ordre 2

$$M_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}$$

en x : $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$

en y : $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

i) Prewitt d'ordre 2

$$M_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}$$

en x : $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$ en y : $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$$M_{yy} = \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix}$$

$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

i) Prewitt d'ordre 2

$$M_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix} \qquad \begin{matrix} \text{en } x \\ \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \end{matrix} \qquad \begin{matrix} \text{en } y \\ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \end{matrix}$$

$$M_{yy} = \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$M_{xy} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

ii) Sobel d'ordre 2

$$M_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix}$$

en x

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

en y

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

ii) Sobel d'ordre 2

$$M_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix} \qquad \begin{matrix} \text{en } x \\ \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \end{matrix} \qquad \begin{matrix} \text{en } y \\ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \end{matrix}$$

$$M_{yy} = \begin{bmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

ii) Sobel d'ordre 2

$$M_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix} \qquad \begin{matrix} \text{en } x \\ \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \end{matrix} \qquad \begin{matrix} \text{en } y \\ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \end{matrix}$$

$$M_{yy} = \begin{bmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$M_{xy} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Source :

©Marie-Flavie Auclair-Fortier

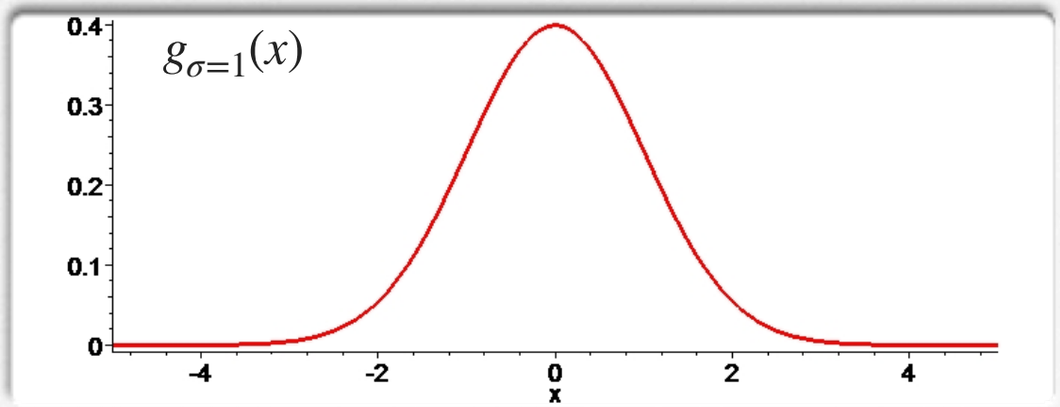
1. Calcul des dérivées d'image

1. Dérivées partielles

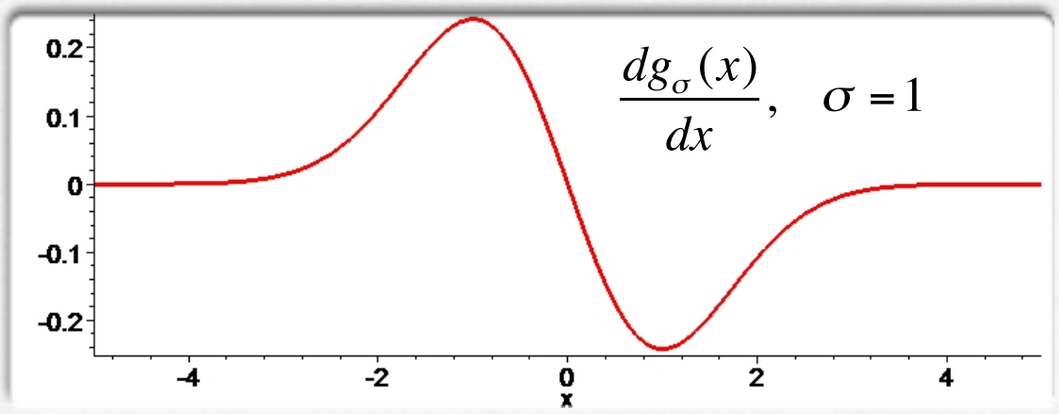
c) Filtres

iii) Canny

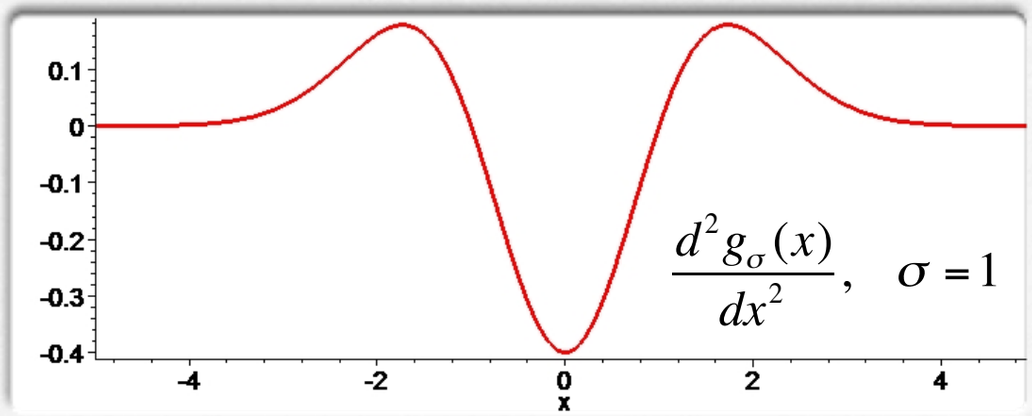
Fonction de Gauss 1D ($\sigma = 1$)



Dérivée d'ordre 1



Dérivée d'ordre 2



Source :

©Marie-Flavie Auclair-Fortier

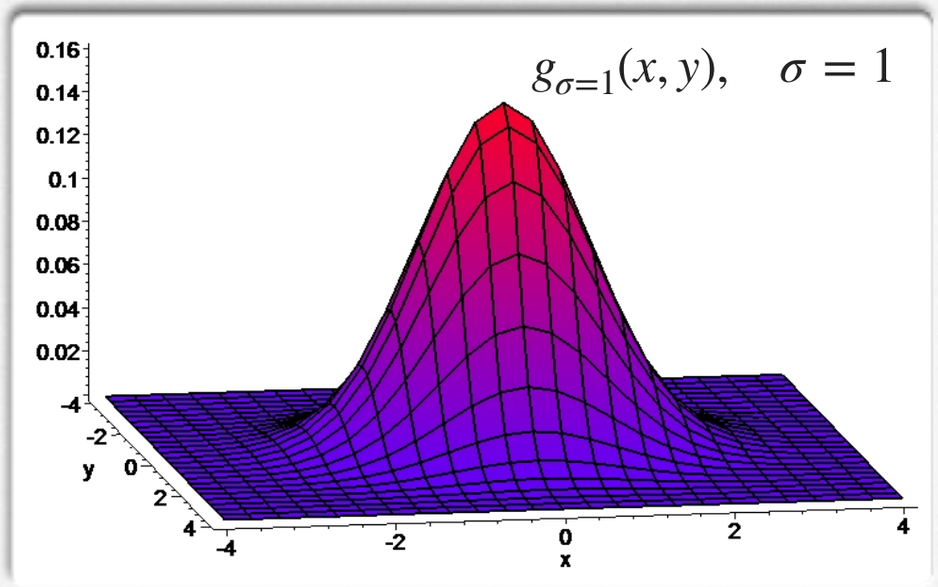
1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

iii) Canny

fonction gaussienne
(lissage 2D)



Source :

©Marie-Flavie Auclair-Fortier

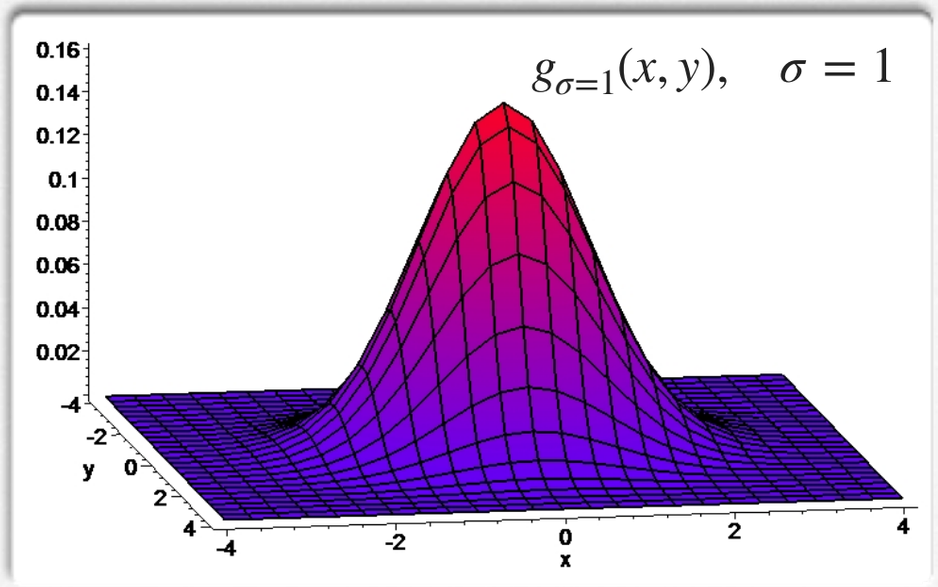
1. Calcul des dérivées d'image

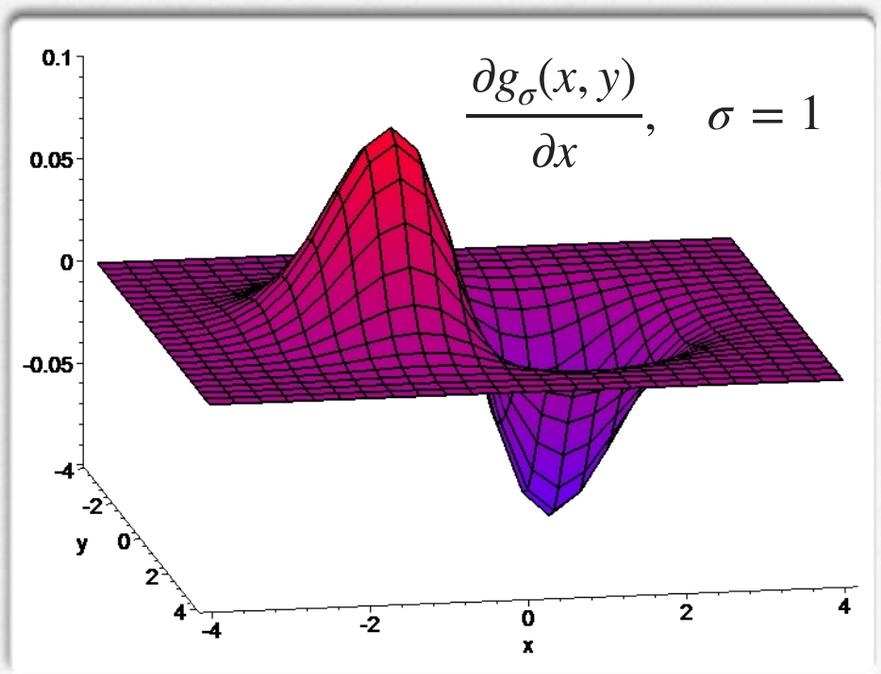
1. Dérivées partielles

c) Filtres

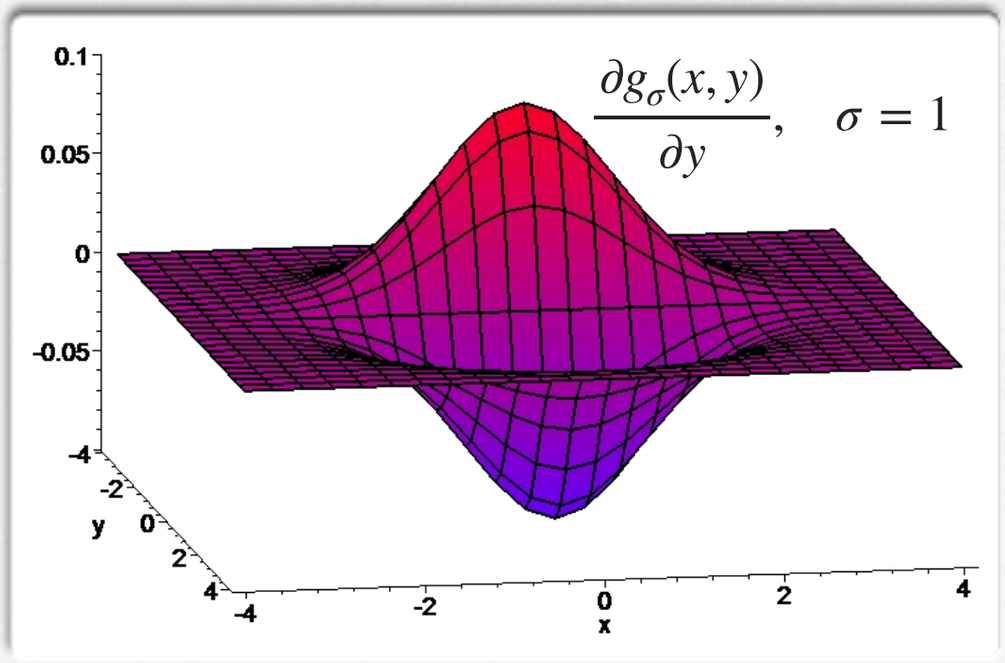
iii) Canny

fonction gaussienne
(lissage 2D)





Dérivées d'ordre 1



Source :

©Marie-Flavie Auclair-Fortier

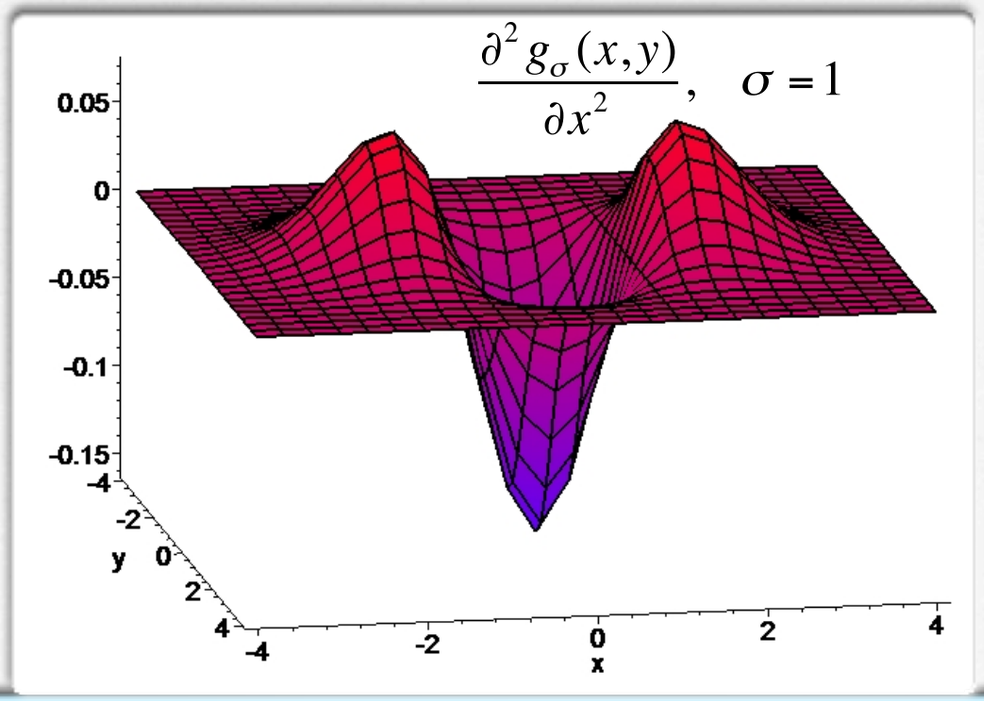
1. Calcul des dérivées d'image

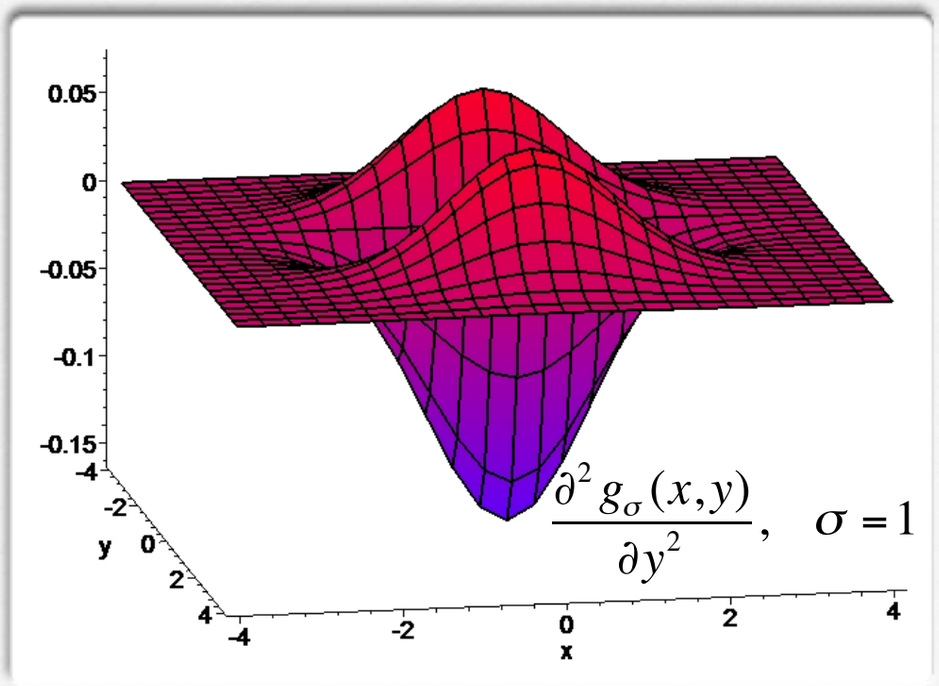
1. Dérivées partielles

c) Filtres

iii) Canny

Dérivées d'ordre 2





Source :

©Marie-Flavie Auclair-Fortier

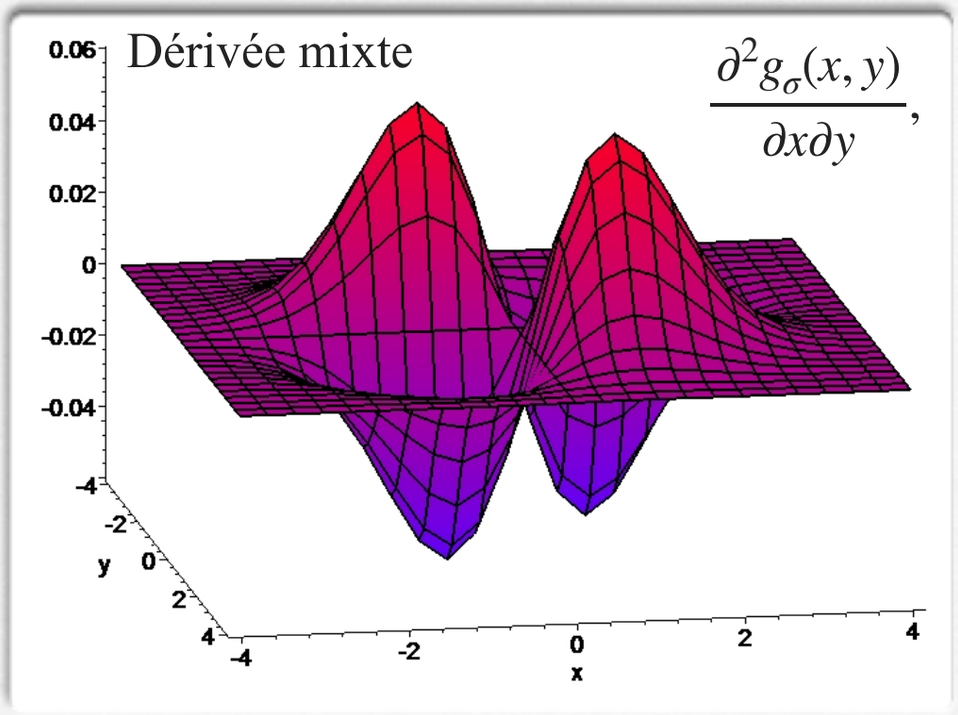
1. Calcul des dérivées d'image

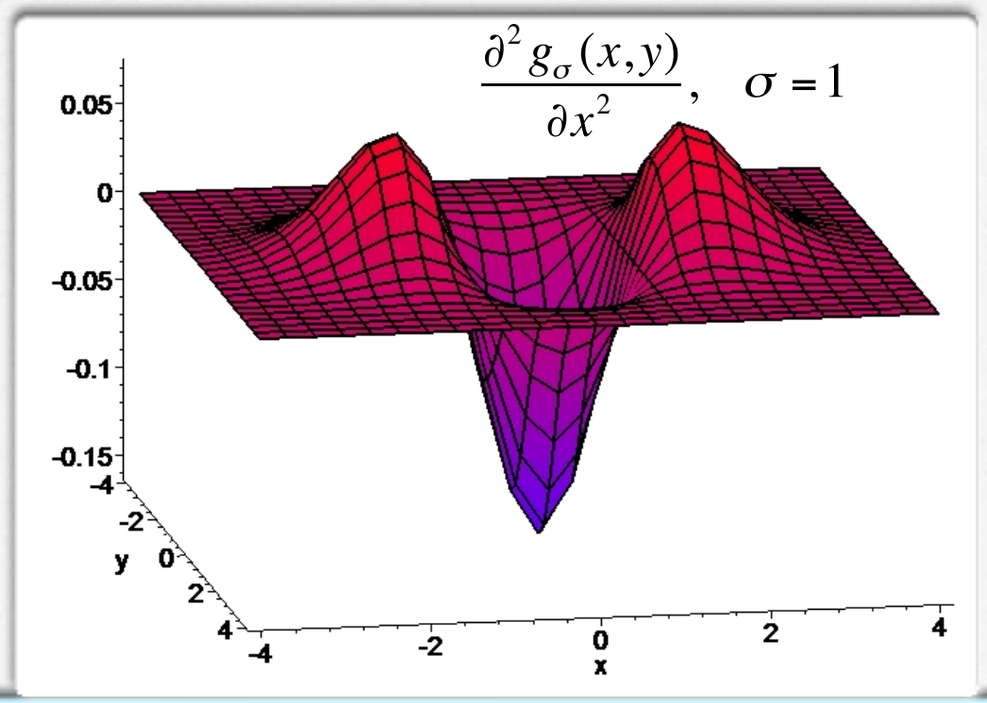
1. Dérivées partielles

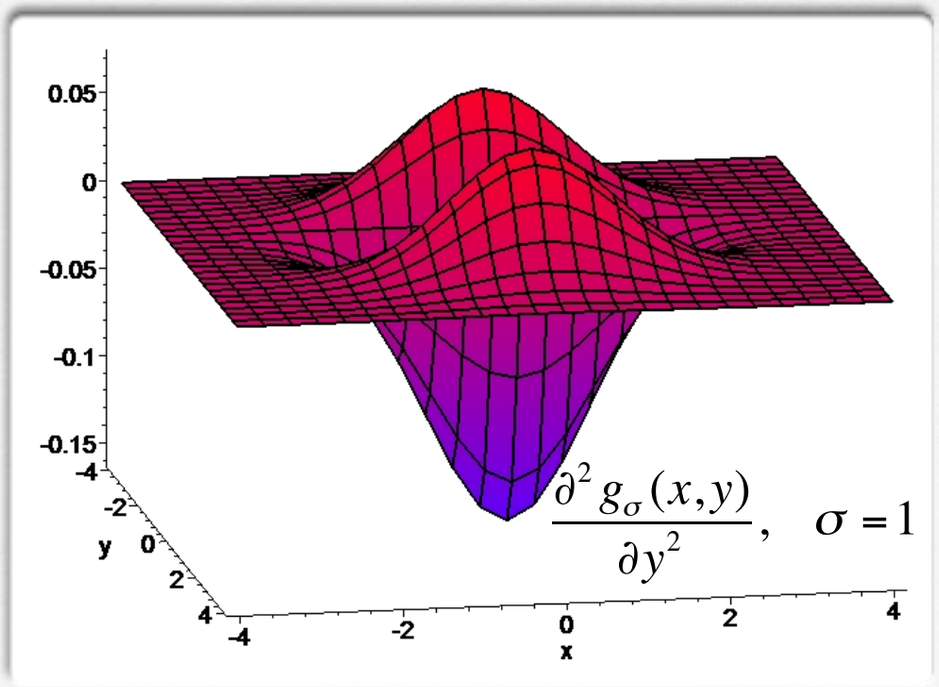
c) Filtres

iii) Canny

Dérivées d'ordre 2







Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

iii) Canny

✓ *Remarques* :

➡ On a un meilleur lissage, mais la convolution est plus longue parce que le masque est plus gros

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

iii) Canny

✓ *Remarques* :

➡ On a un meilleur lissage, mais la convolution est plus longue parce que le masque est plus gros

➡ Pour tous les masques de dérivées, ***il faut*** que la somme des éléments soient = 0

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

iii) Canny

✓ *Remarques* :

➡ On a un meilleur lissage, mais la convolution est plus longue parce que le masque est plus gros

➡ Pour tous les masques de dérivées, ***il faut*** que la somme des éléments soient = 0

➡ On a les mêmes considérations concernant la numérisation que pour le masque de lissage (moyenne pondérée)

★Dimension $N \times N$ avec N impair

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

1. Dérivées partielles

c) Filtres

iii) Canny

✓ *Remarques* :

➡ On a un meilleur lissage, mais la convolution est plus longue parce que le masque est plus gros

➡ Pour tous les masques de dérivées, ***il faut*** que la somme des éléments soient = 0

➡ On a les mêmes considérations concernant la numérisation que pour le masque de lissage (moyenne pondérée)

★Dimension $N \times N$ avec N impair

★Choix de N pour la dimension des masques : la dérivée agrandit le domaine significativement > 0, cela mène à la règle du pouce :

★ordre 1 : $N > 7\sigma + 1$

★ordre 2 : $N > 8\sigma + 1$

Source :

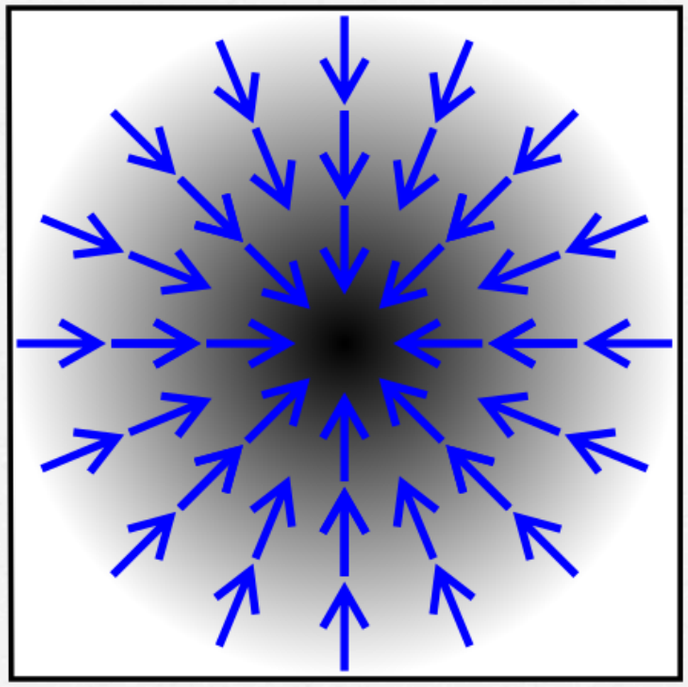
©Marie-Flavie Auclair-Fortier

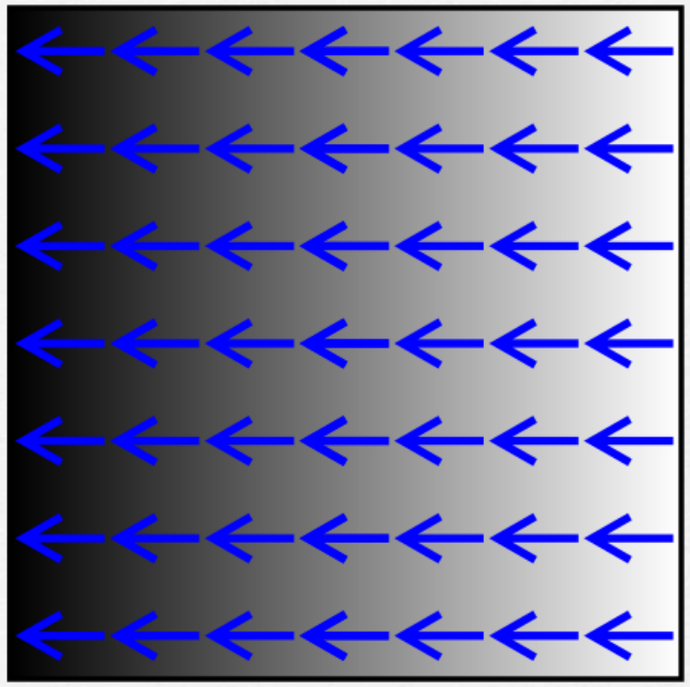
1. Calcul des dérivées d'image

2. Gradient

Important

- ∇f représente la direction et l'amplitude du changement maximal de la fonction f





Source : https://commons.wikimedia.org/wiki/File:Gradient2.svg

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

$f[m,n]$
Image originale

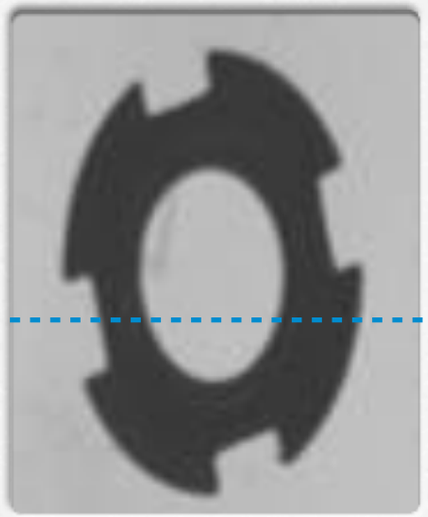
Source :

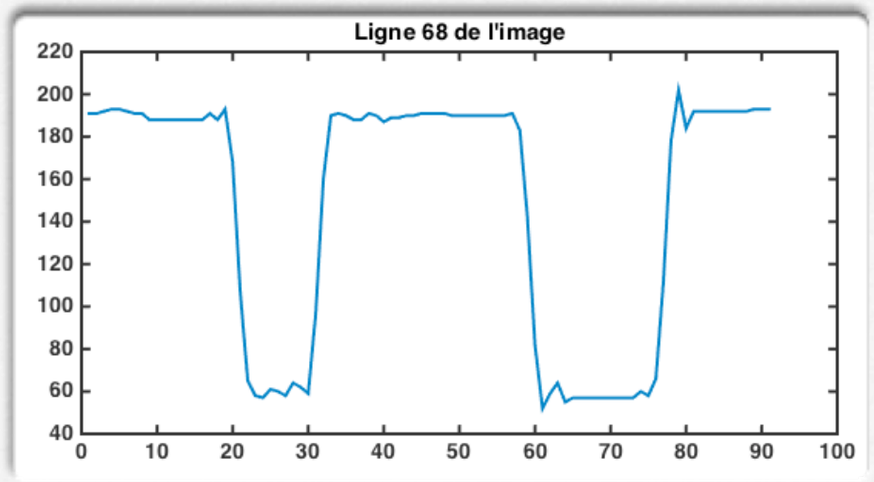
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

$f[m,n]$
Image originale





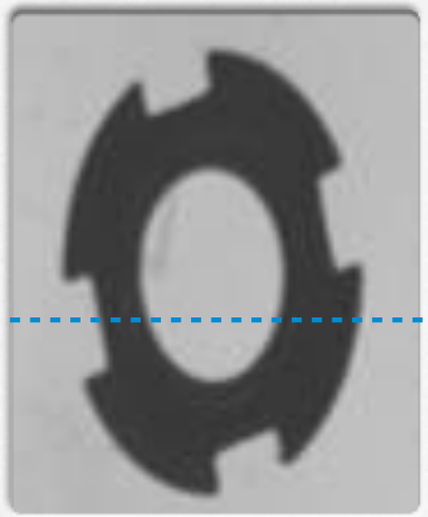
Source :

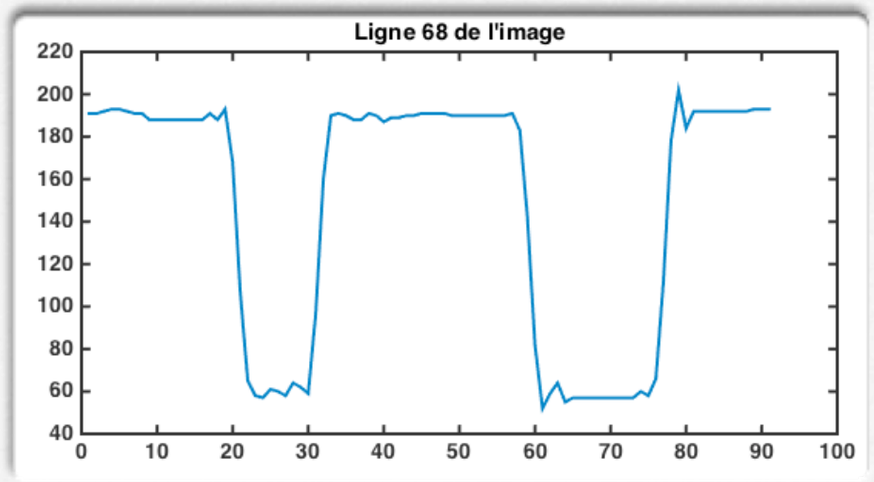
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

$f[m,n]$
Image originale





$f_m[m,n]$
Dérivée en x à base de lissage
(masques gaussiens)

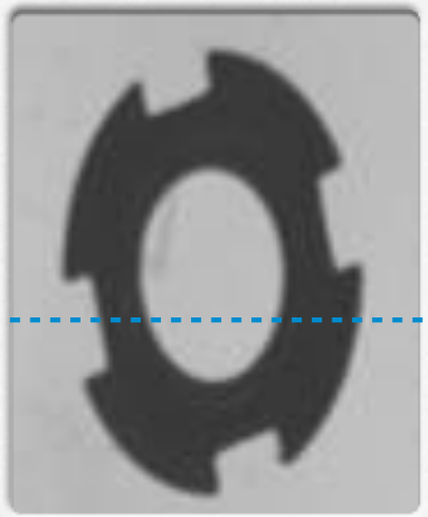
Source :

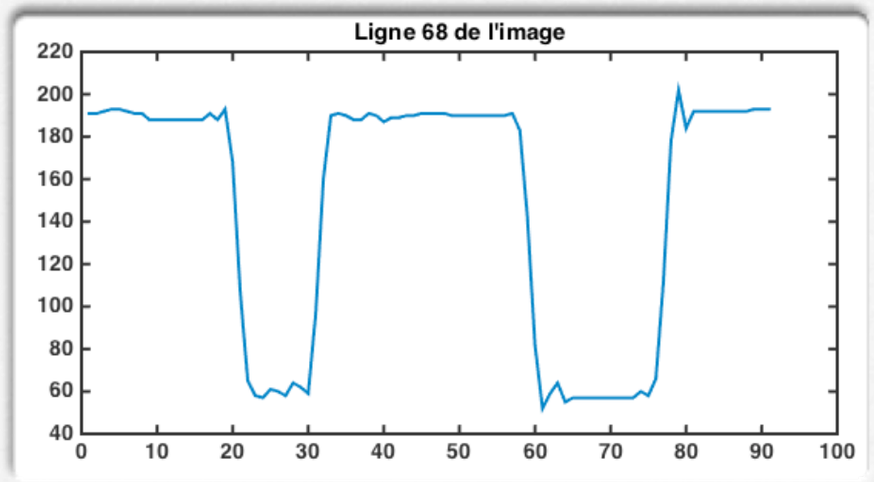
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

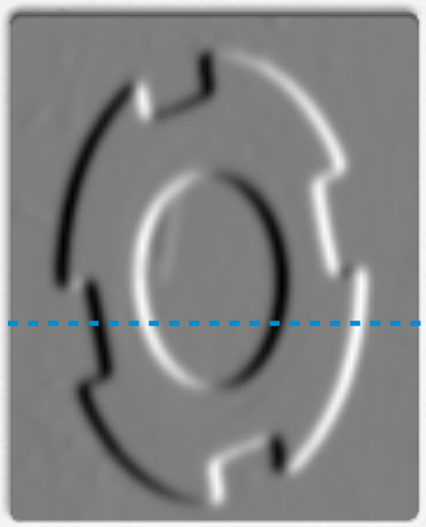
2. Gradient

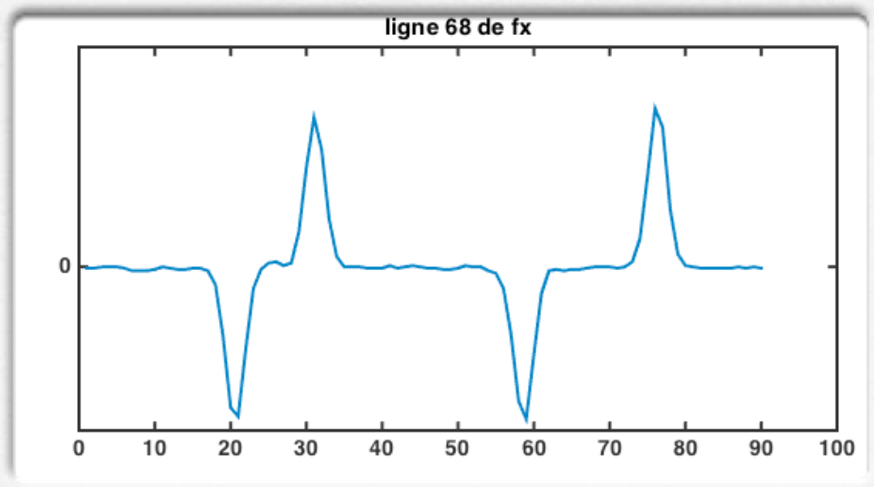
$f[m,n]$
Image originale





$f_m[m,n]$
Dérivée en x à base de lissage
(masques gaussiens)





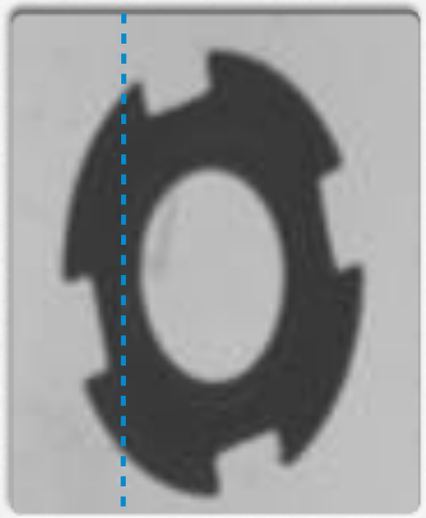
Source :

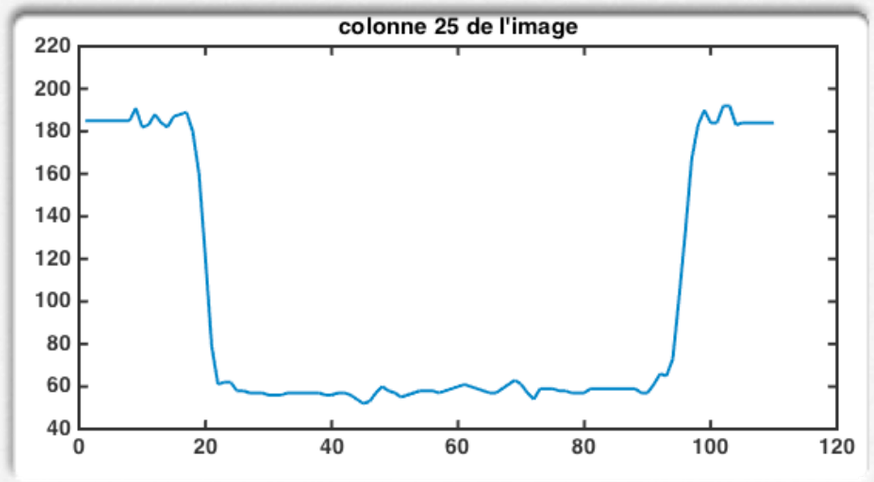
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

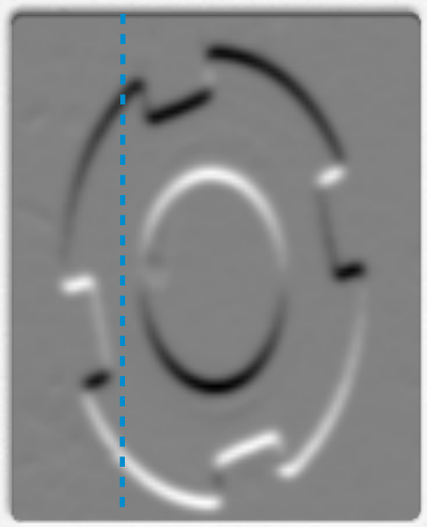
2. Gradient

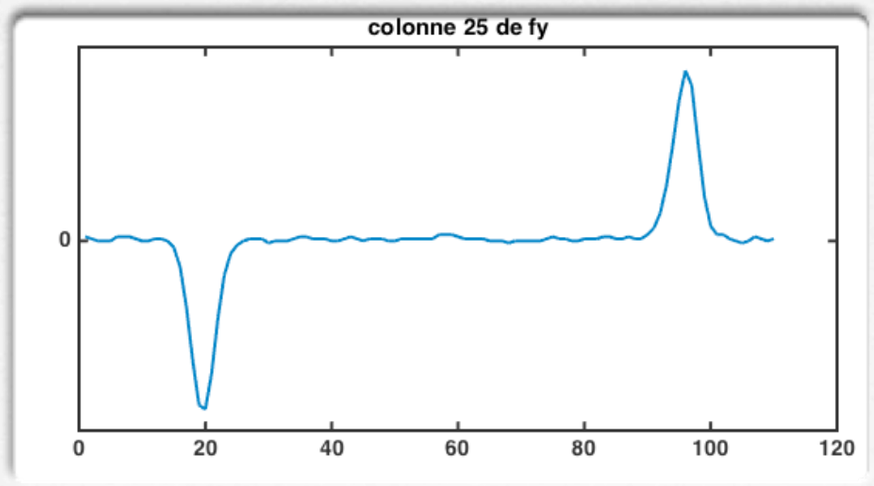
$f[m,n]$
Image originale





$f_n[m,n]$
Dérivée en y à base de lissage
(masques gaussiens)





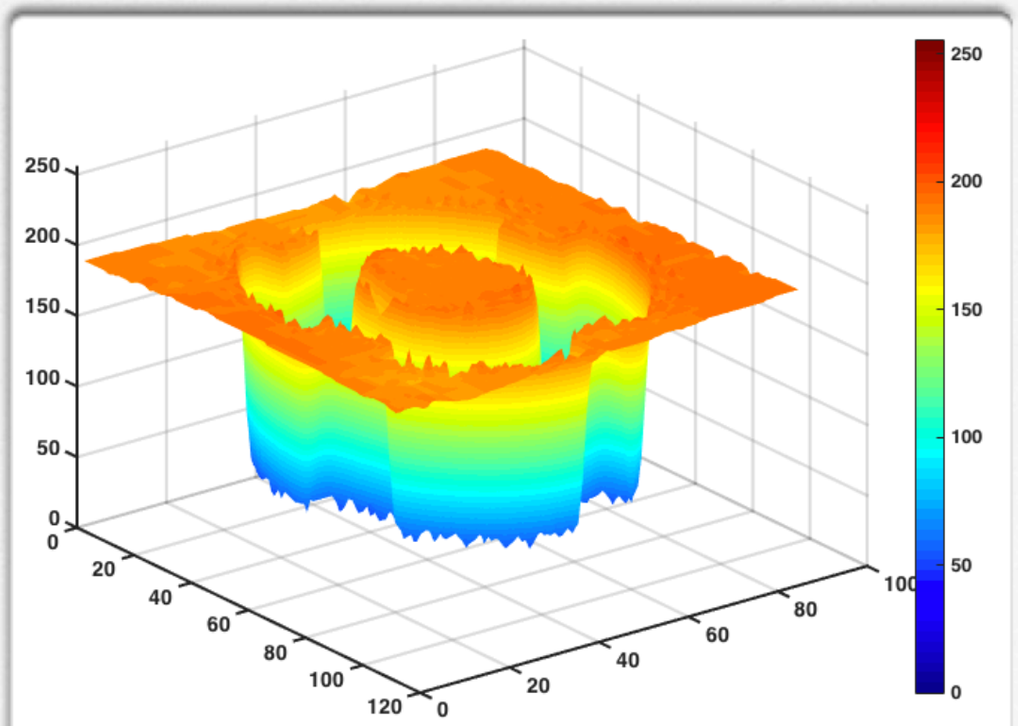
Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

Image originale



Source :

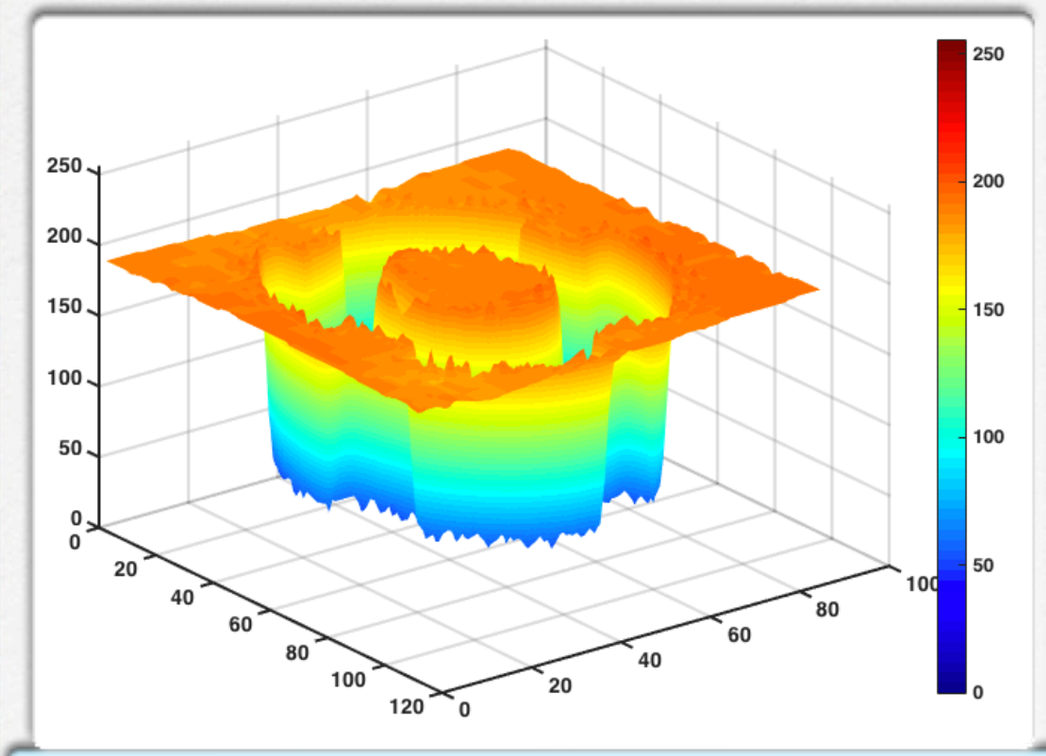
©Marie-Flavie Auclair-Fortier

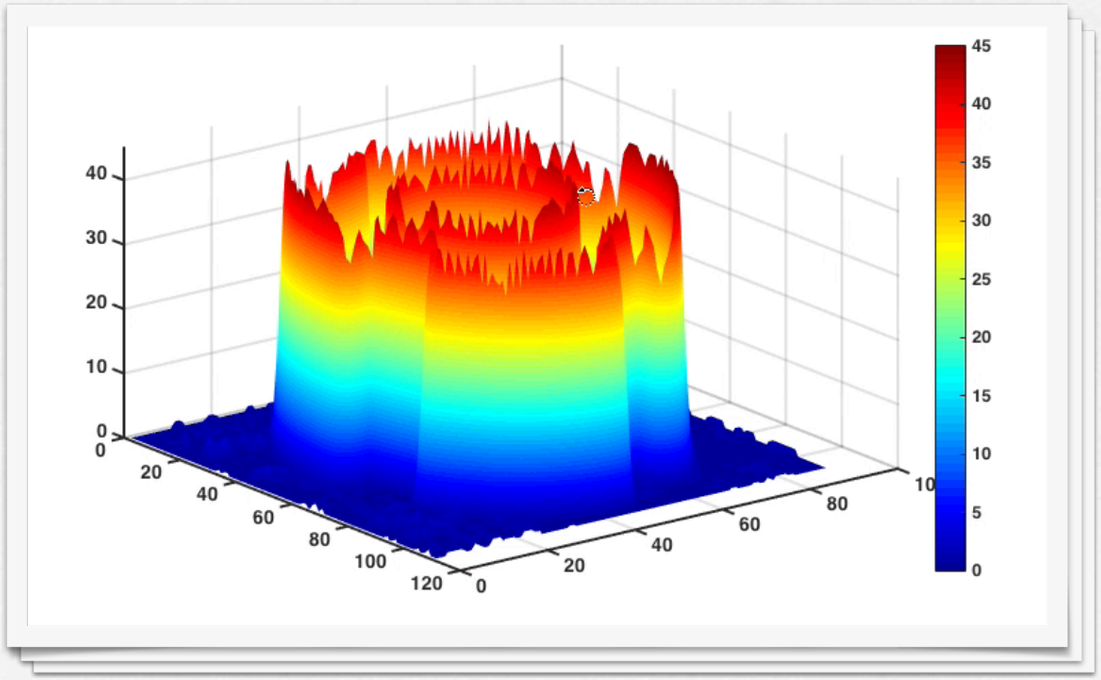
1. Calcul des dérivées d'image

2. Gradient

Image originale

Amplitude du gradient : $\left| \nabla f[m,n] \right|$



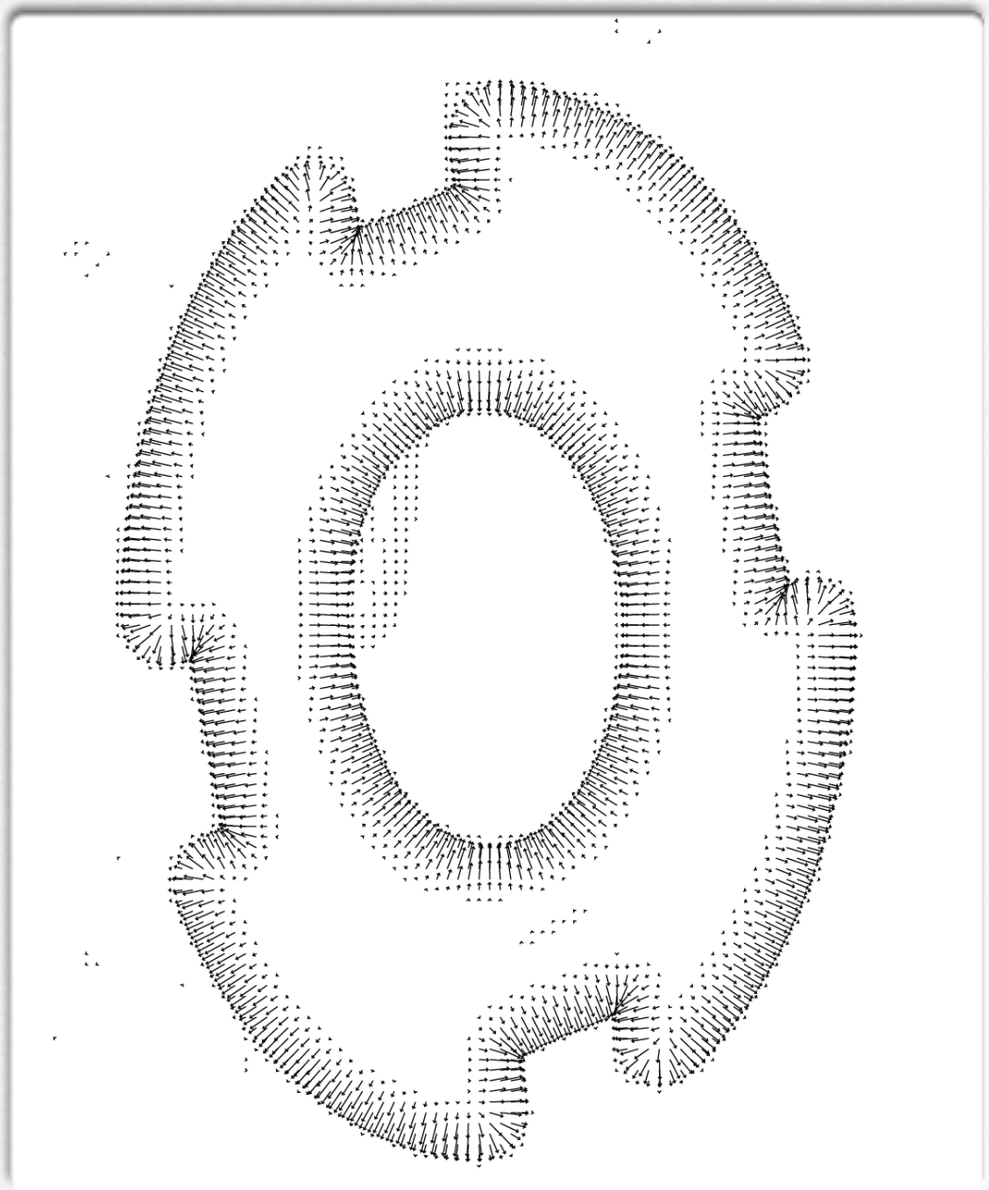


Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient



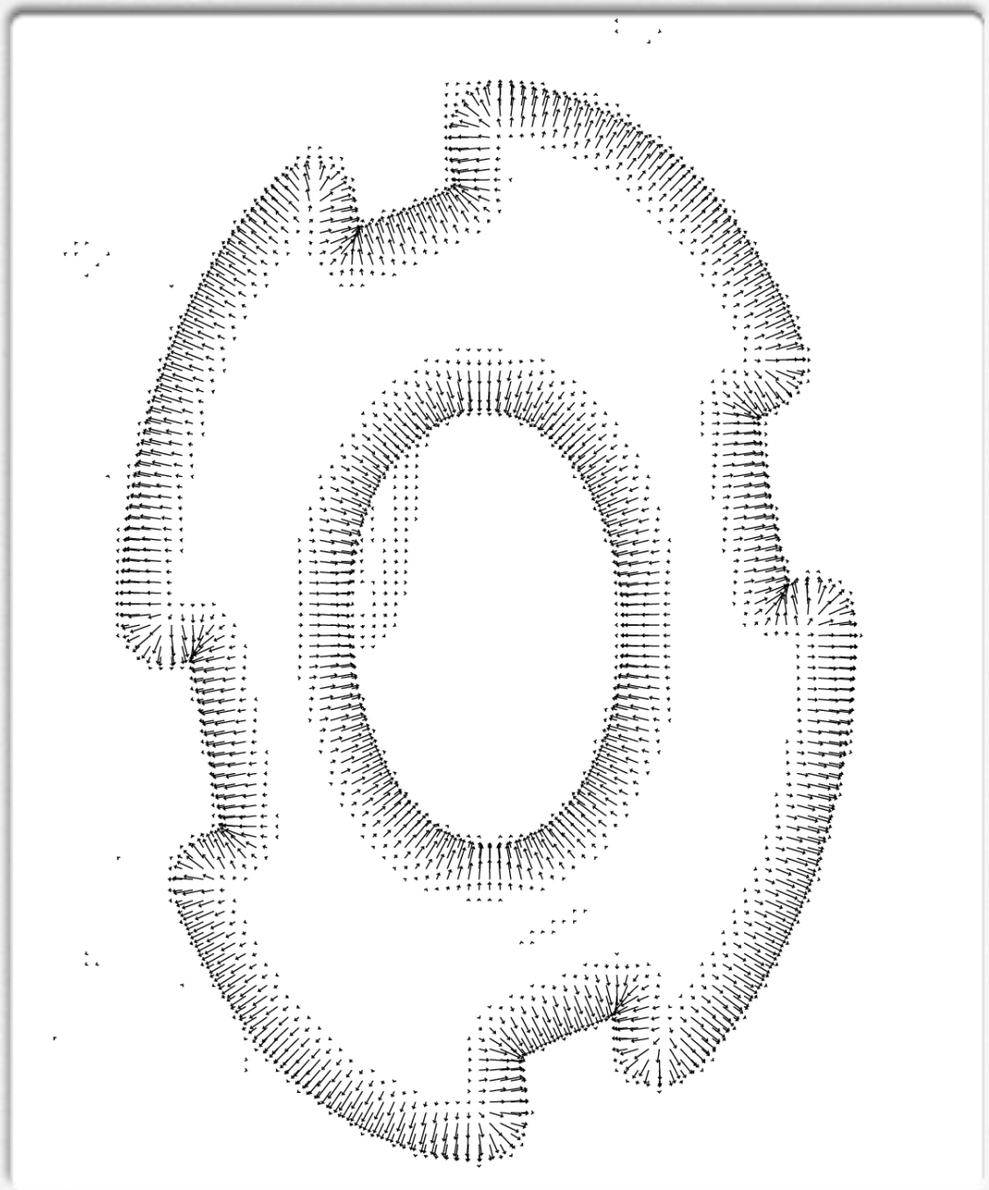
Gradients représentés
sous forme de vecteurs

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient



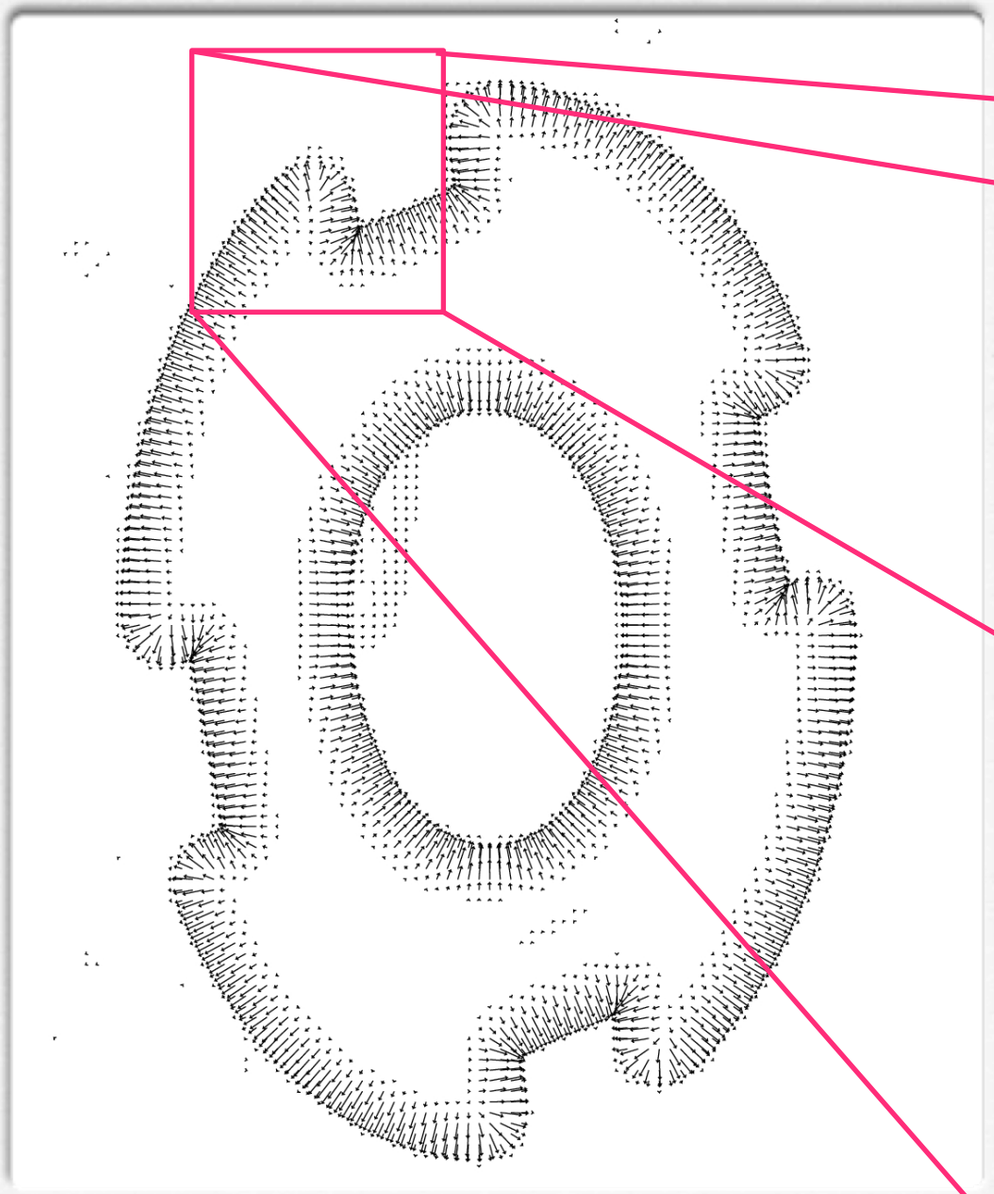
Gradients représentés
sous forme de vecteurs

Source :

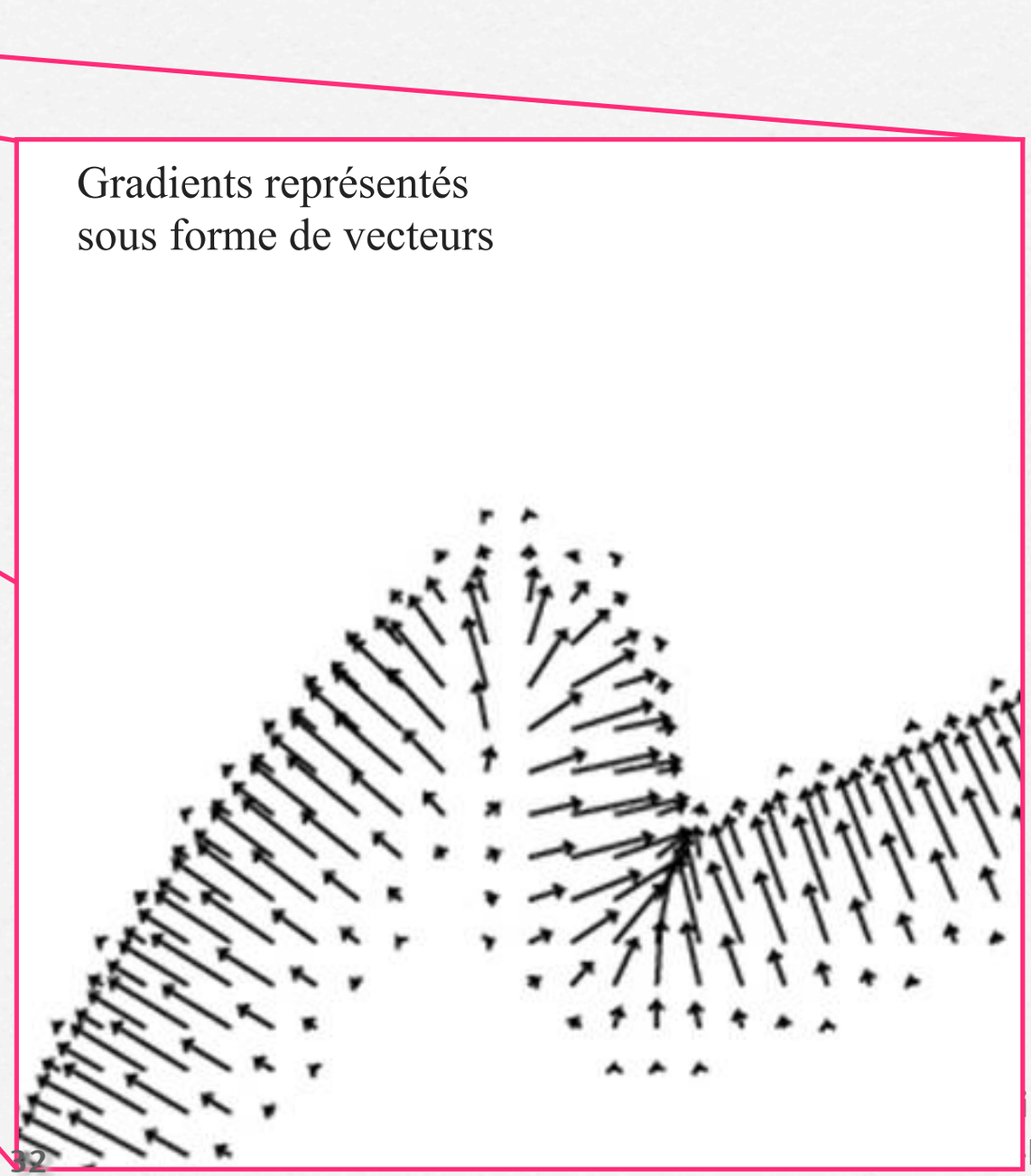
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient



Gradients représentés
sous forme de vecteurs



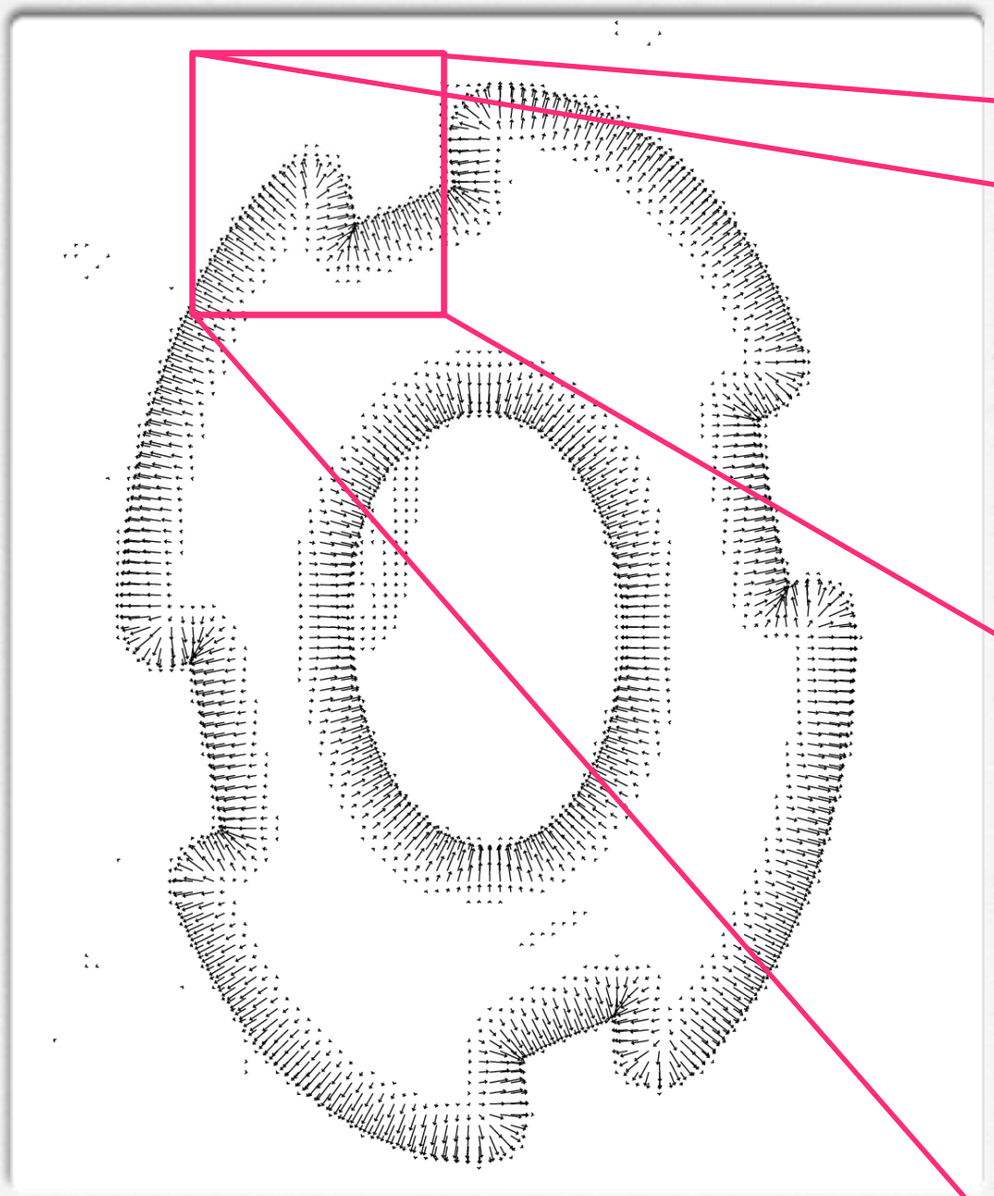
Source :

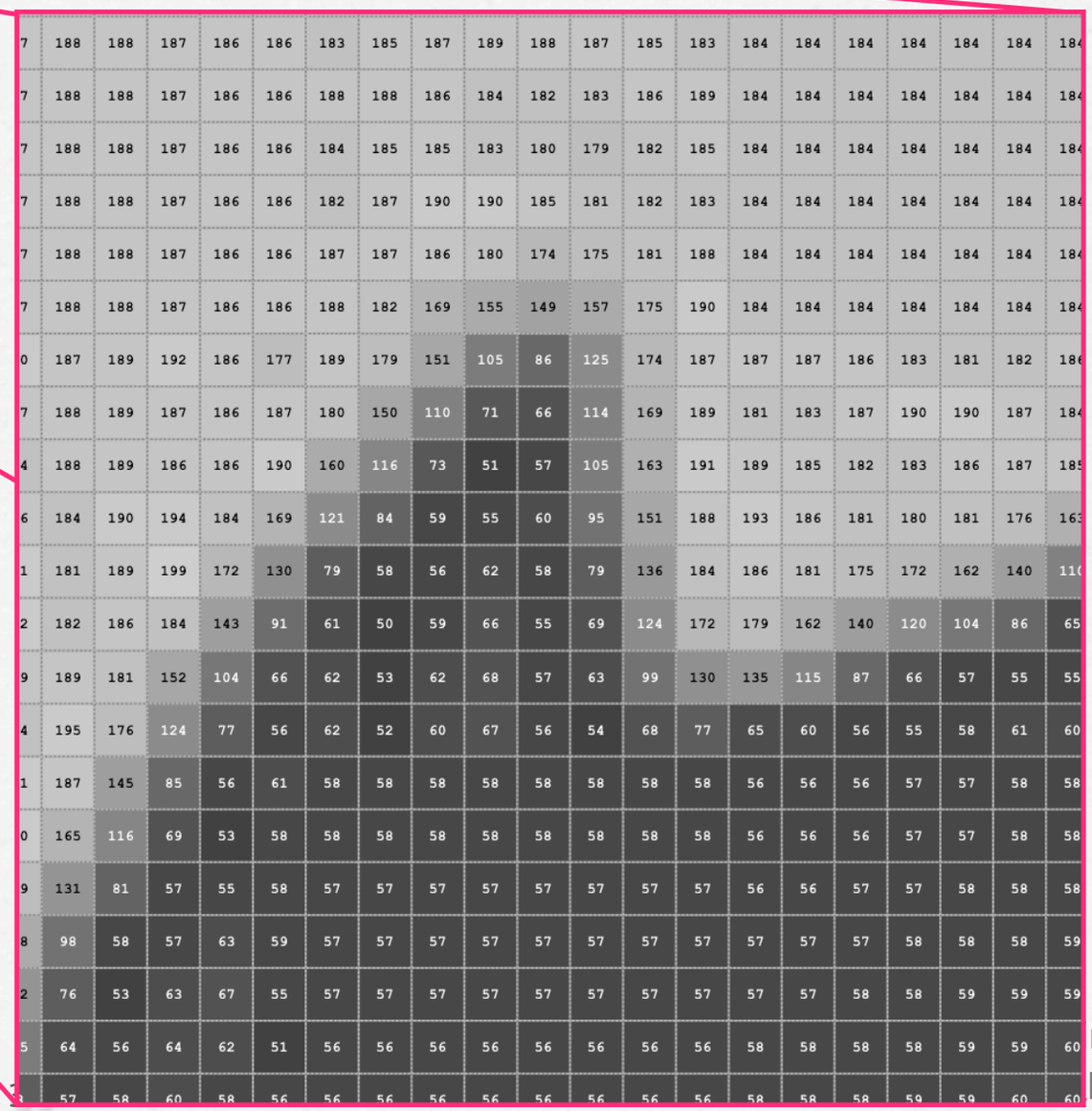
1. Calcul des dérivées d'imagi

2. Gradient

Important :

- $\nabla f[m,n]$ représente la direction et l'amplitude du changement maximal de la fonction f au point $[m,n]$





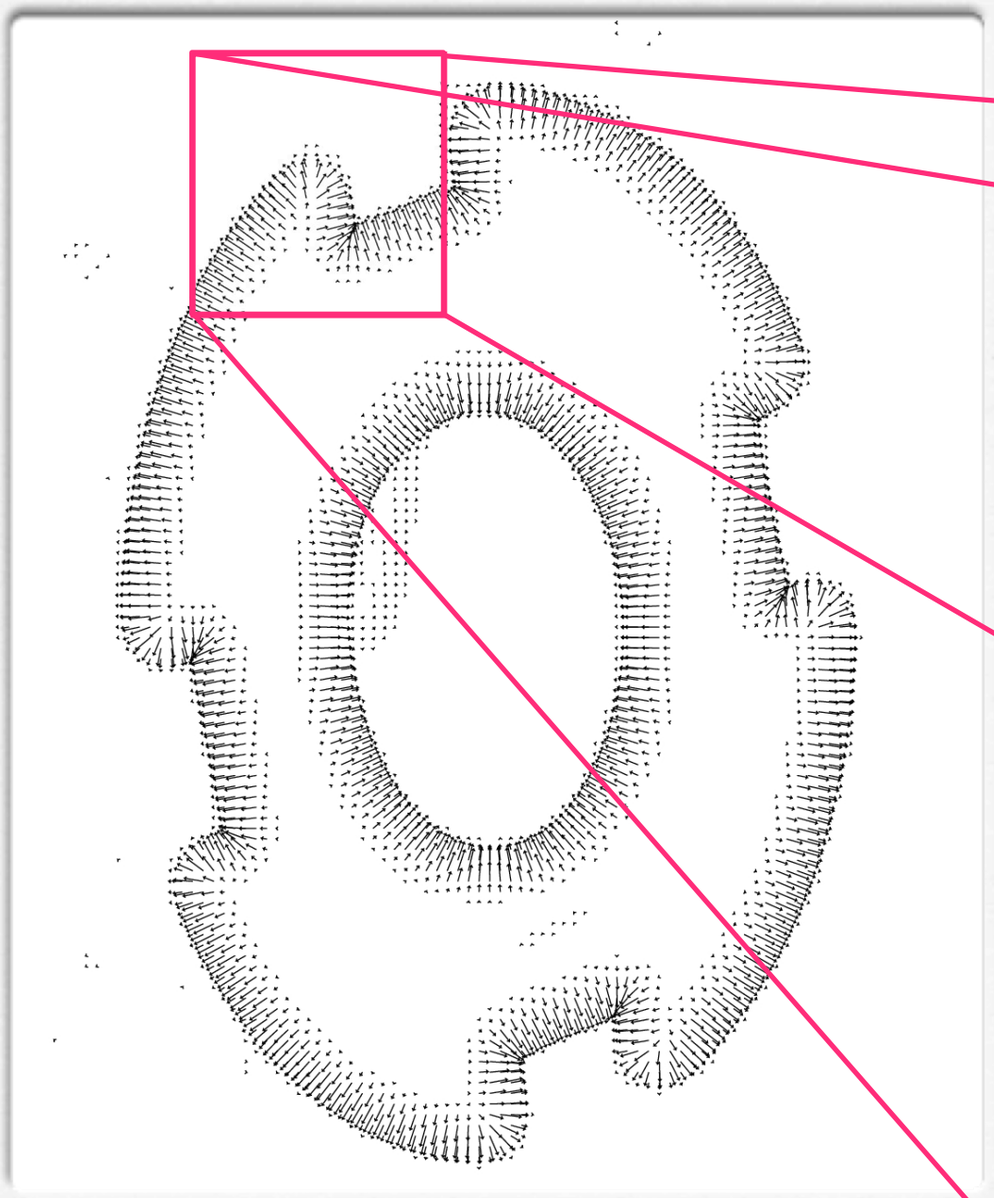
Source : mages Fortier

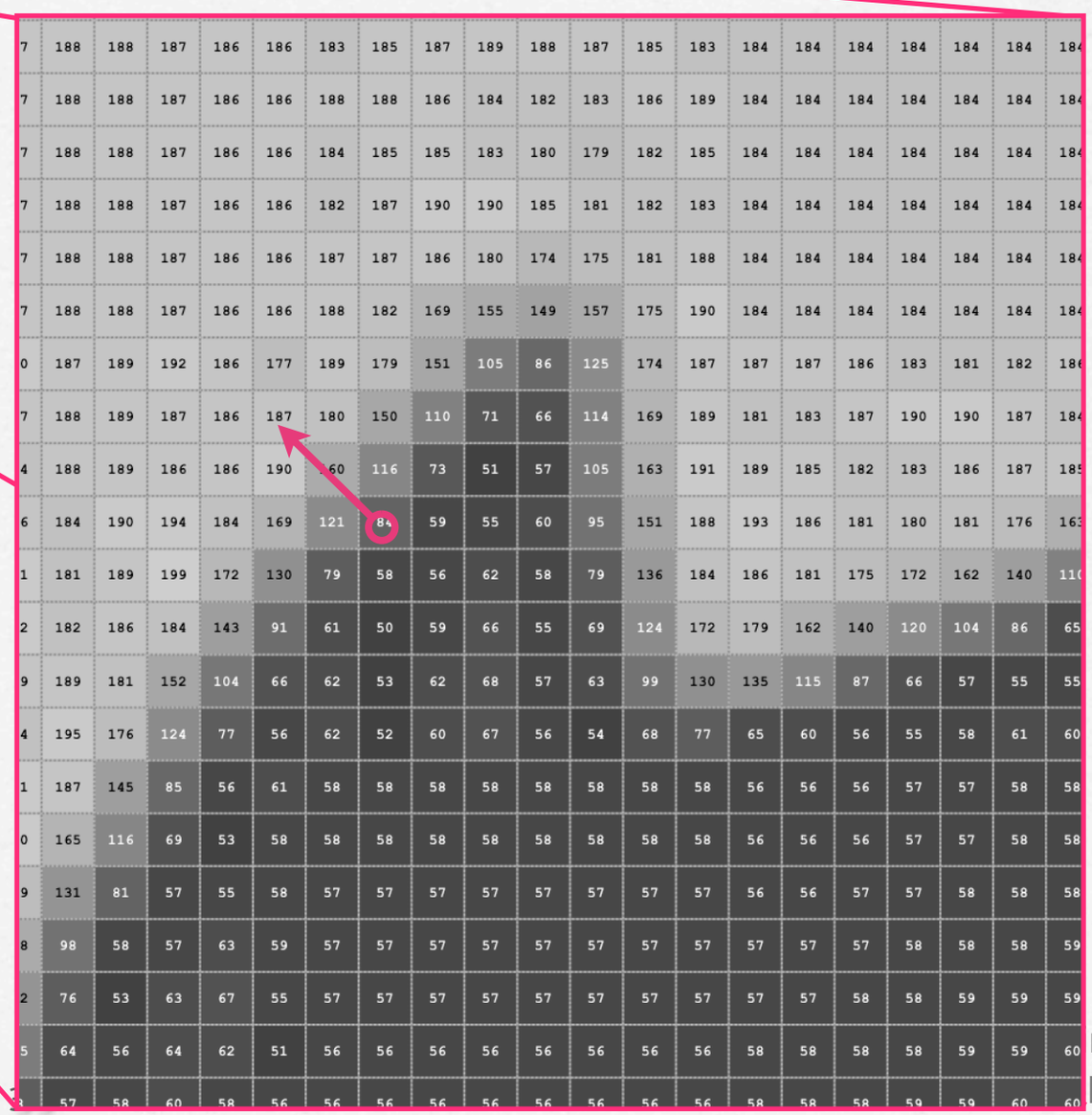
1. Calcul des dérivées d'imagi

2. Gradient

Important :

- $\nabla f[m,n]$ représente la direction et l'amplitude du changement maximal de la fonction f au point $[m,n]$





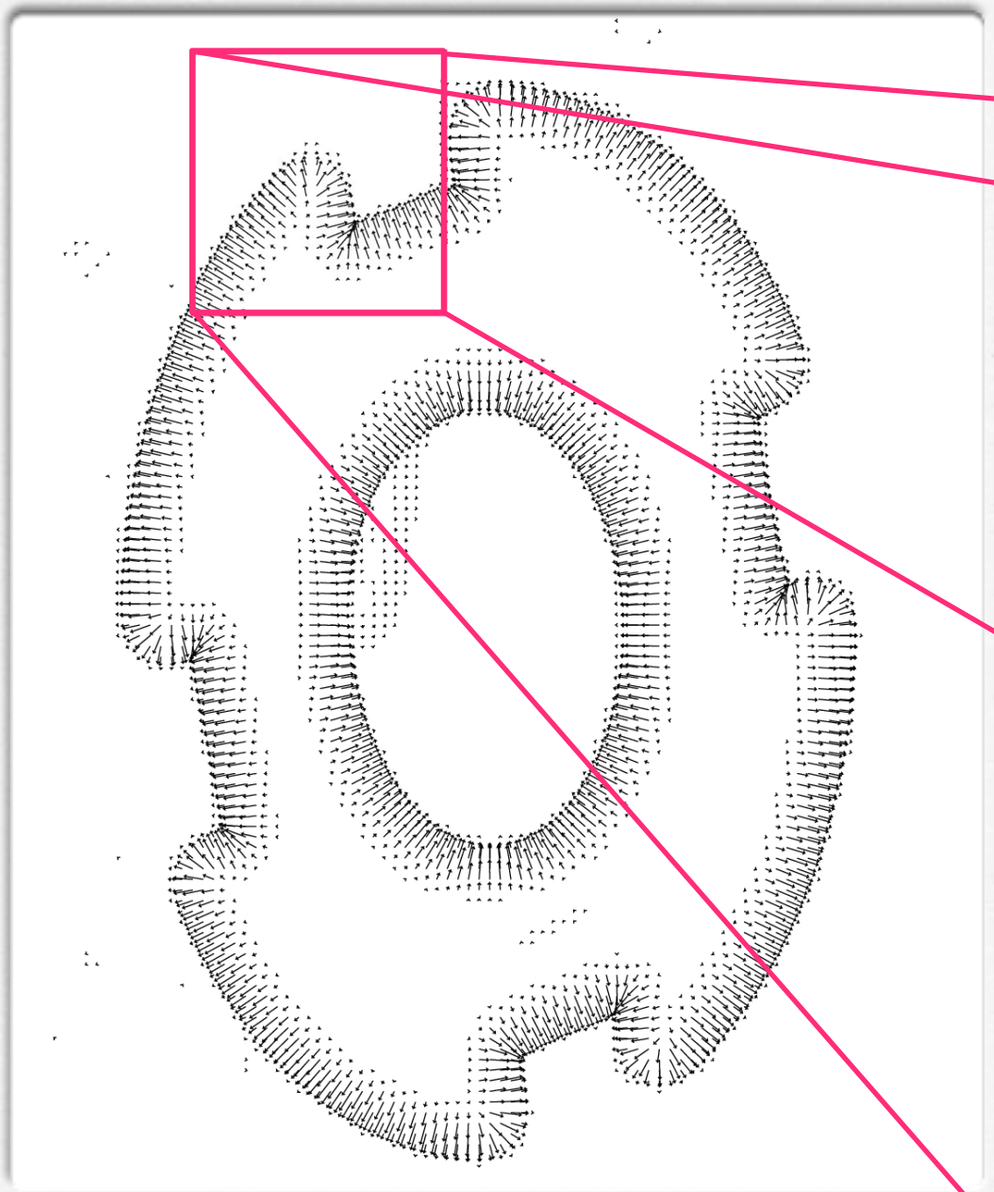
Source : mages Fortier

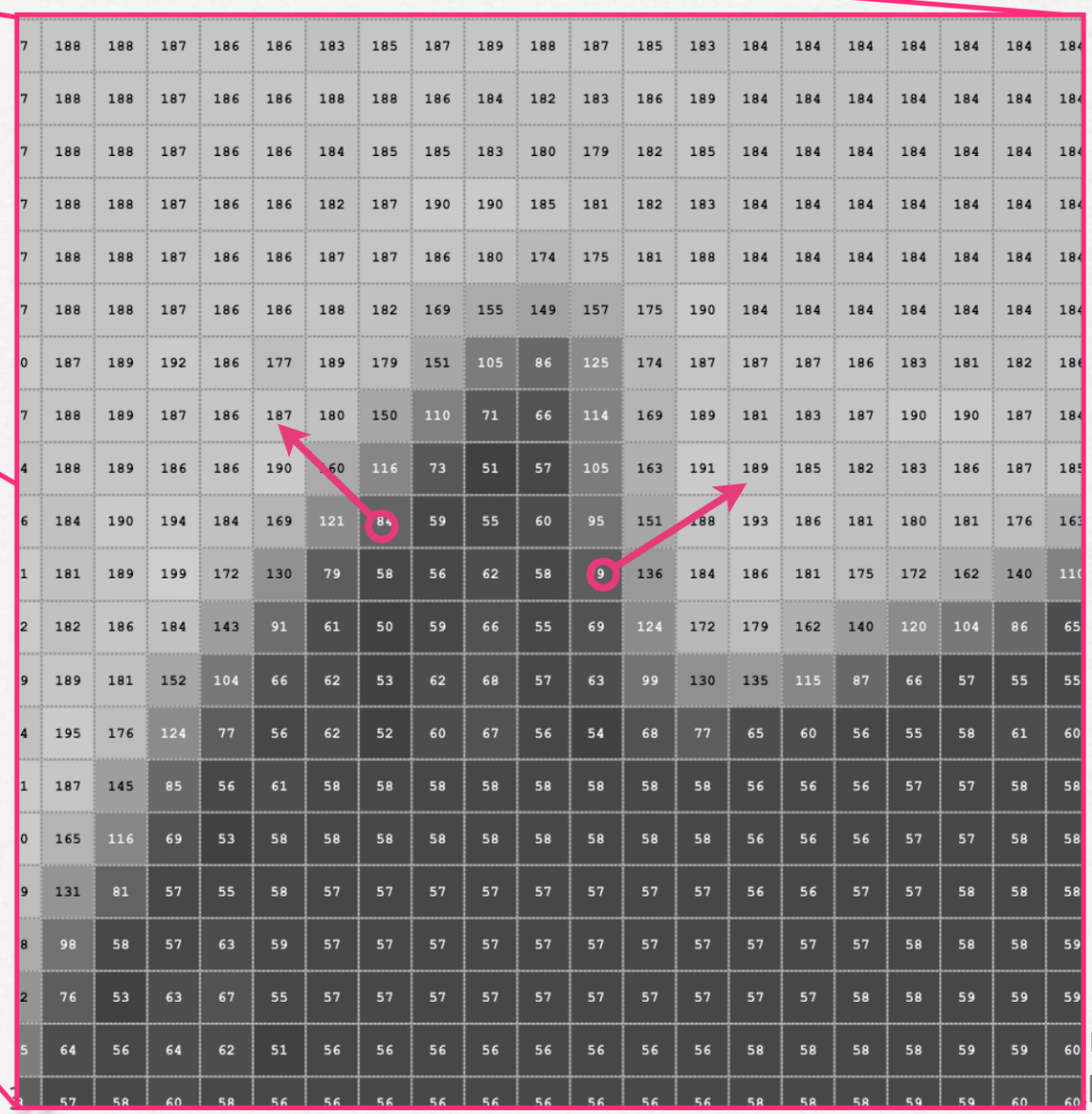
1. Calcul des dérivées d'imagi

2. Gradient

Important :

- $\nabla f[m,n]$ représente la direction et l'amplitude du changement maximal de la fonction f au point $[m,n]$





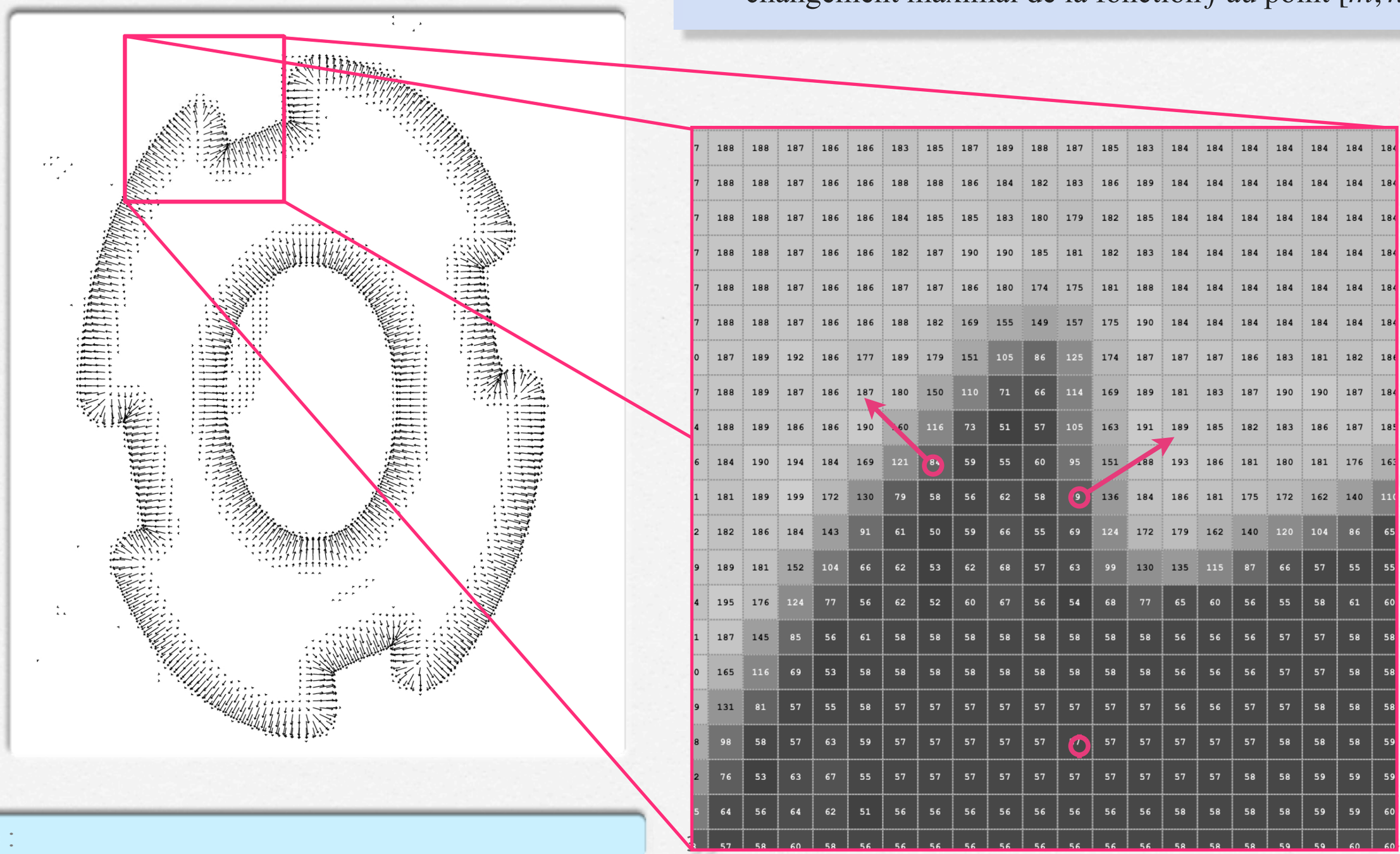
Source : mages Fortier

1. Calcul des dérivées d'imagi

2. Gradient

Important :

- $\nabla f[m,n]$ représente la direction et l'amplitude du changement maximal de la fonction f au point $[m,n]$



Source :

images Fortier

1. Calcul des dérivées d'image

2. Gradient

- Importance du gradient
  - ✓ permet la détection des points de contour d'une image

Source : Noah Snavely, CS6670: Computer Vision

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

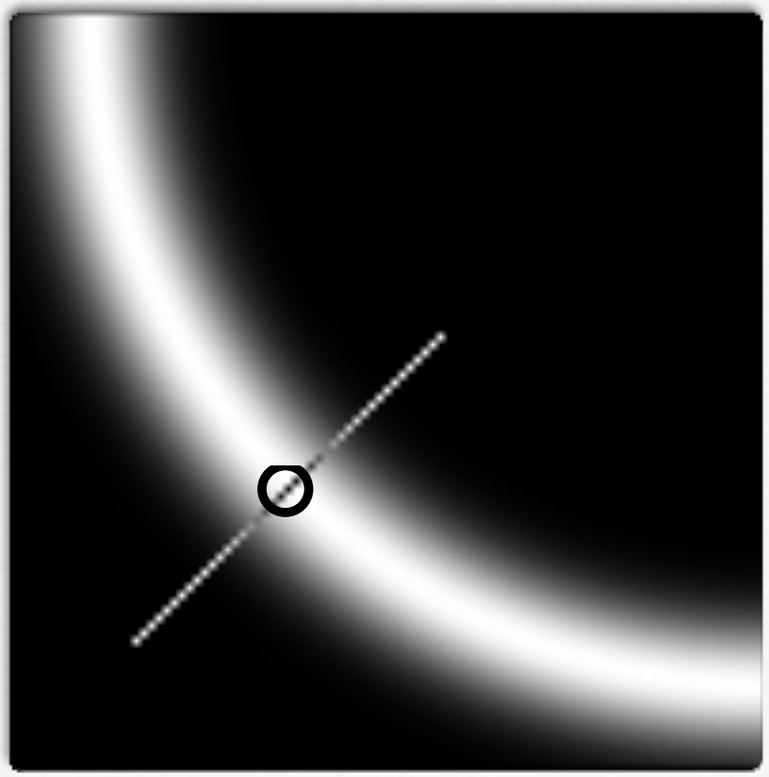
- Importance du gradient
  - ✓ permet la détection des points de contour d'une image
    - ➡ le contour est un maximum local de $\left|\nabla f\right|$ dans la direction $\angle \nabla f$ au pixel $[m, n]$

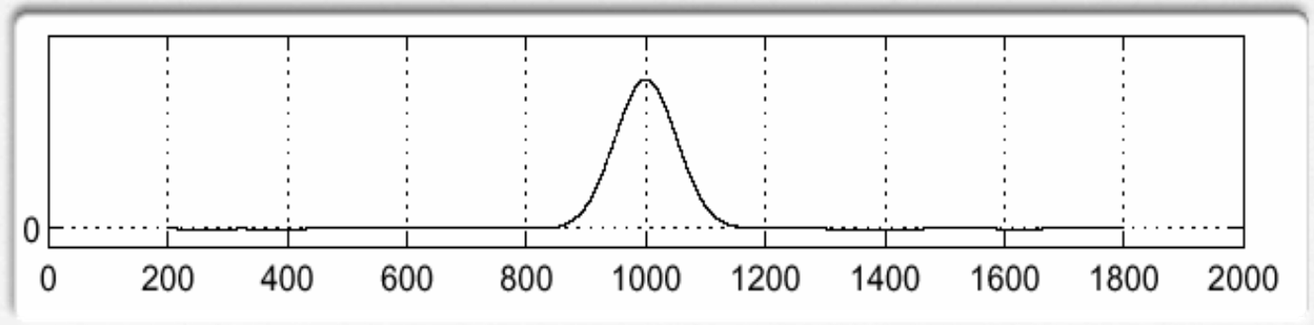
Source : Noah Snavely, CS6670: Computer Vision

1. Calcul des dérivées d'image

2. Gradient

- Importance du gradient
  - ✓ permet la détection des points de contour d'une image
    - ➡ le contour est un maximum local de $\left|\nabla f\right|$ dans la direction $\angle \nabla f$ au pixel $[m, n]$
    - ➡ $\angle \nabla f$ est perpendiculaire au contour au pixel $[m, n]$





section 2, détection de contours

Source : Noah Snavely, CS6670: Computer Vision

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

- *Remarque* :
  - ✓ Dans le contexte de la détection de contours, on veut avoir l'orientation ($\in [-\pi/2,\pi/2]$) du contour et pas tant l'orientation du changement max ($\in [-\pi, \pi]$)

1. Calcul des dérivées d'image

2. Gradient

- *Remarque* :
  - ✓ Dans le contexte de la détection de contours, on veut avoir l'orientation ($\in [-\pi/2,\pi/2]$) du contour et pas tant l'orientation du changement max ($\in [-\pi, \pi]$)

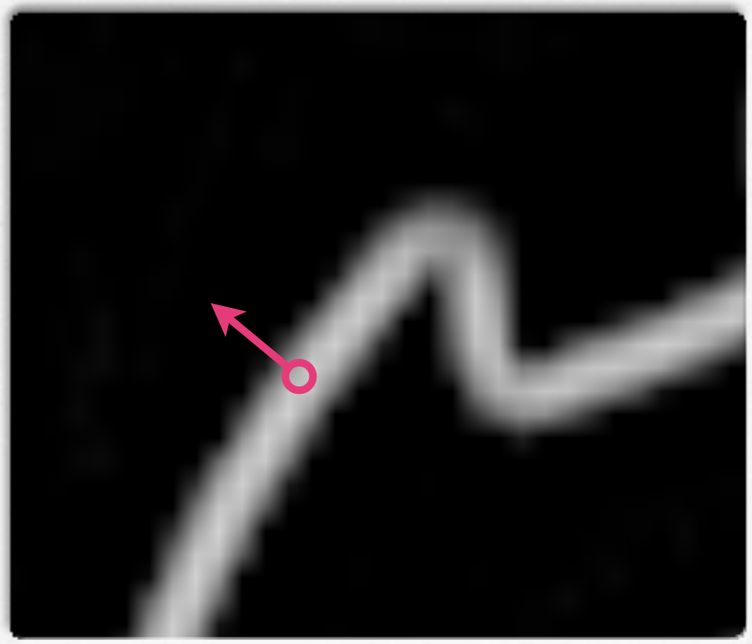
Source : Noah Snavely, CS6670: Computer Vision

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

- *Remarque* :
  - ✓ Dans le contexte de la détection de contours, on veut avoir l'orientation ($\in [-\pi/2,\pi/2]$) du contour et pas tant l'orientation du changement max ($\in [-\pi, \pi]$)



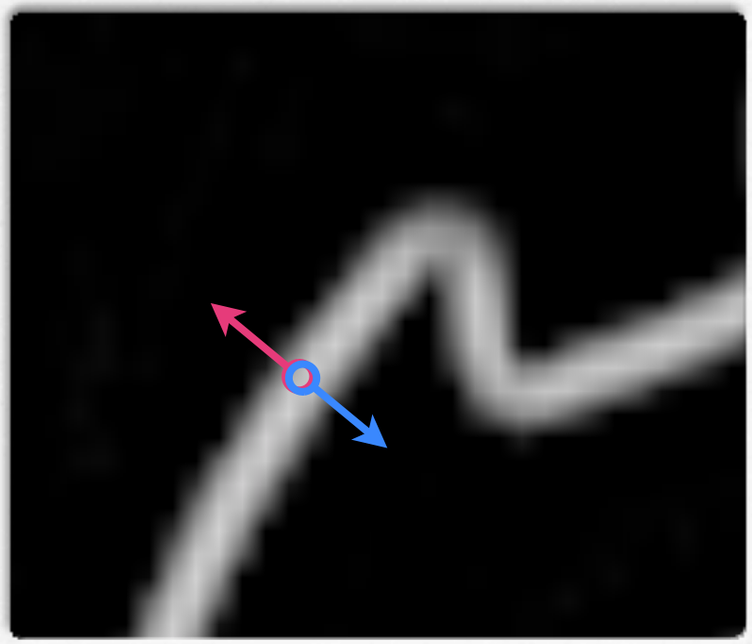
Source : Noah Snavely, CS6670: Computer Vision

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

- *Remarque* :
  - ✓ Dans le contexte de la détection de contours, on veut avoir l'orientation ($\in [-\pi/2,\pi/2]$) du contour et pas tant l'orientation du changement max ($\in [-\pi, \pi]$)



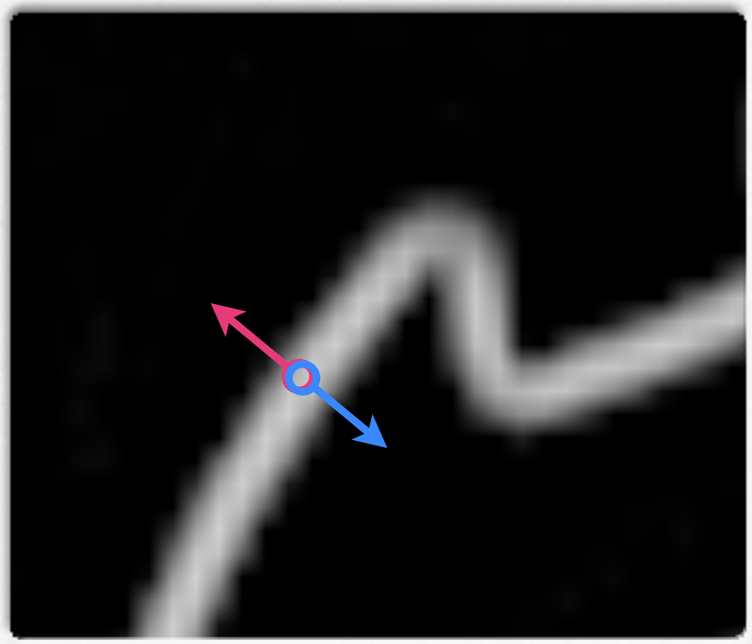
Source : Noah Snavely, CS6670: Computer Vision

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

2. Gradient

- *Remarque* :
  - ✓ Dans le contexte de la détection de contours, on veut avoir l'orientation ($\in [-\pi/2,\pi/2]$) du contour et pas tant l'orientation du changement max ($\in [-\pi, \pi]$)



ou : aucune importance

- revient à dire que la plupart du temps, on ne s'intéresse pas à savoir si c'est une marche montante ou descendante

Source : Noah Snavely, CS6670: Computer Vision

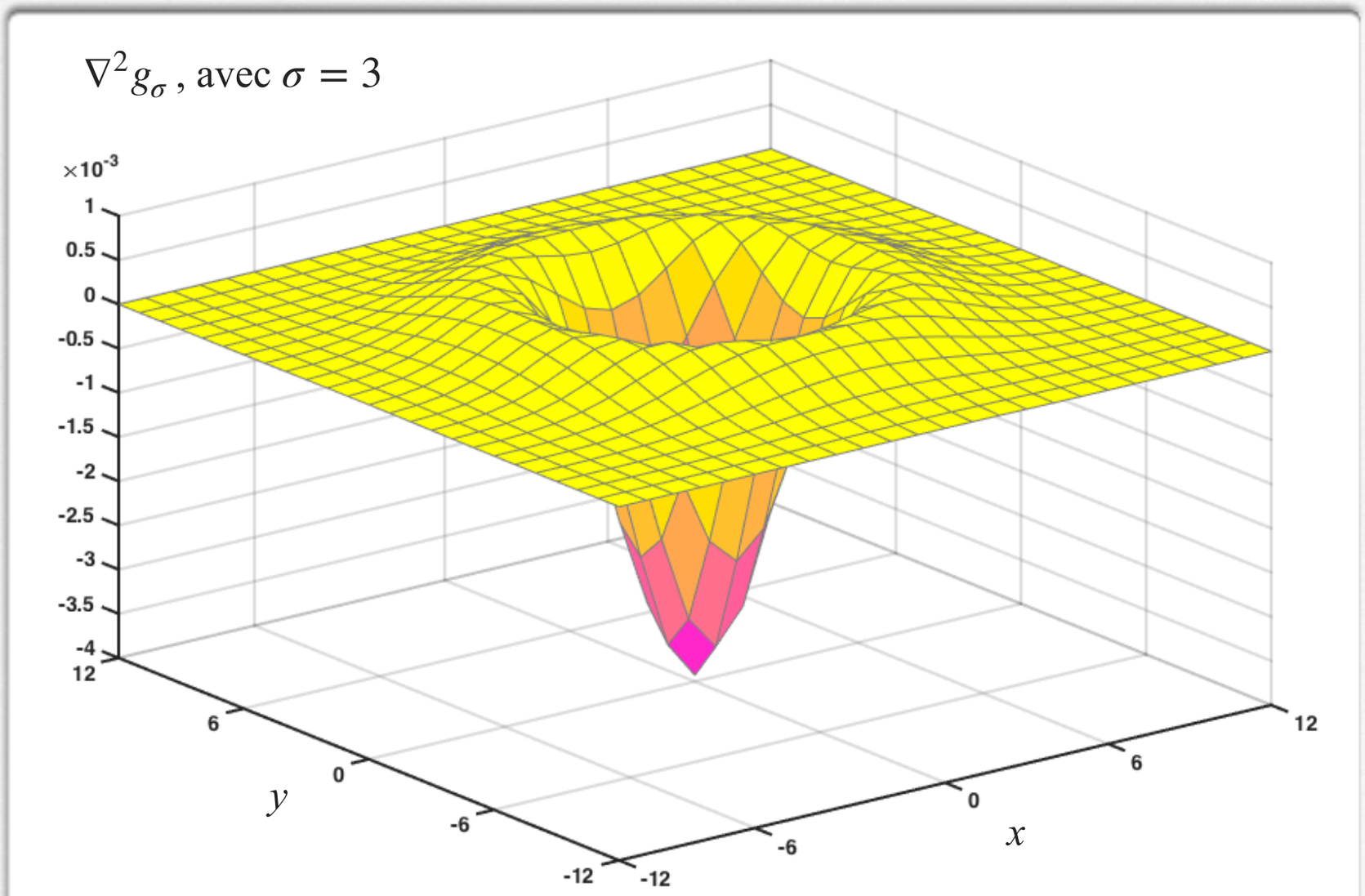
©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

3. Laplacien (somme des 2e dérivées)

a) Masques

iv) Marr & Hildreth



Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

3. Laplacien

a) Masques

iv) Marr & Hildreth

✓ Séparabilité de la fonction gaussienne

$\left(f * \nabla^2 g_\sigma\right)[m,n]$: requiert N^2 multiplications pour chaque pixel

$$= \left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial m^2}[m] \right) * g_\sigma[n] + \left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial n^2}[n] \right) * g_\sigma[m]$$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

3. Laplacien

a) Masques

iv) Marr & Hildreth

✓ Séparabilité de la fonction gaussienne

$(f * \nabla^2 g_\sigma)[m,n]$: requiert N^2 multiplications pour chaque pixel

$$= \underbrace{\left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial m^2}[m] \right) * g_\sigma[n]}_{\text{Convolution 1D ligne par ligne}} + \underbrace{\left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial n^2}[n] \right) * g_\sigma[m]}_{\text{Convolution 1D colonne par colonne}}$$

Convolution 1D colonne par colonne

Convolution 1D ligne par ligne

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

3. Laplacien

a) Masques

iv) Marr & Hildreth

✓ Séparabilité de la fonction gaussienne

$(f * \nabla^2 g_\sigma)[m,n]$: requiert N^2 multiplications pour chaque pixel

$$= \underbrace{\left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial m^2}[m] \right) * g_\sigma[n]}_{\text{Convolution 1D ligne par ligne}} + \underbrace{\left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial n^2}[n] \right) * g_\sigma[m]}_{\text{Convolution 1D colonne par colonne}}$$

Convolution 1D colonne par colonne

Convolution 1D ligne par ligne

✓ requiert $4N$ multiplications pour chaque pixel

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

3. Laplacien

a) Masques

iv) Marr & Hildreth

✓ Séparabilité de la fonction gaussienne

$(f * \nabla^2 g_\sigma)[m,n]$: requiert N^2 multiplications pour chaque pixel

$$= \underbrace{\left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial m^2}[m] \right) * g_\sigma[n]}_{\text{Convolution 1D ligne par ligne}} + \underbrace{\left(f[m,n] * \frac{\partial^2 g_\sigma}{\partial n^2}[n] \right) * g_\sigma[m]}_{\text{Convolution 1D colonne par colonne}}$$

Convolution 1D colonne par colonne

Convolution 1D ligne par ligne

- ✓ requiert $4N$ multiplications pour chaque pixel
- ✓ plus avantageux si $\sigma > 1$

Source :

©Marie-Flavie Auclair-Fortier

1. Calcul des dérivées d'image

3. Laplacien

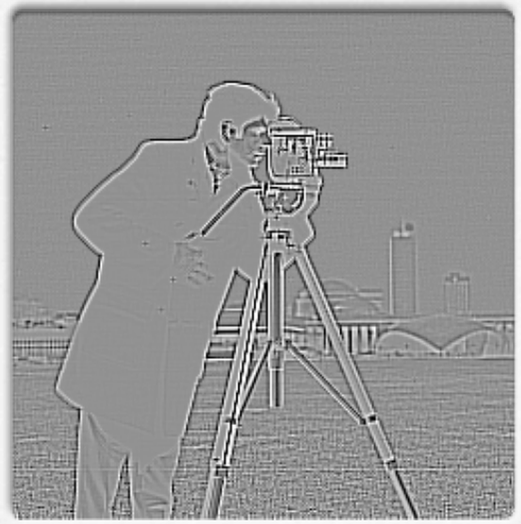
a) Masques

- comparaison

différences finies
($h = 1/2$)

Variante 2

Variante 1



Laplacien numérique (w=3)

Marr & Hildreth
($\sigma = 1, N = 9$)

Marr & Hildreth
($\sigma = 3, N = 25$)

Source :

©Marie-Flavie Auclair-Fortier

Plan

1. Calcul des dérivées d'une image
2. Contours
   - 2.1. Détecteur gradient
     a) Seuillage du gradient
     b) Suppression des non-maxima (du gradient)
     c) Retrait des faux contours (pas à l'examen)
     d) Détecteur de Canny (pas à l'examen)
   - 2.2. Détecteur Laplacien
3. Coins
4. Segmentation

©Marie-Flavie Auclair-Fortier

2. CONTOURS

- Objectif

Image originale : $f[m, n]$

2. CONTOURS

- Objectif

Image originale : $f[m, n]$

Détecteur

Image des contours : $c[m, n]$

✓ image binaire

2. CONTOURS

- Objectif

$$c[m,n] \in \{\text{contour, non-contour}\}$$

étiquettes

Image originale : $f[m, n]$

Détecteur

Image des contours : $c[m, n]$

✓ image binaire

Source : PM Jodoin

2. CONTOURS

- Objectif

$$c[m,n] \in \{\text{contour, non-contour}\}$$

étiquettes

Image originale : $f[m, n]$

Détecteur

Image des contours : $c[m, n]$

✓ image binaire

Remarque :

- la valeur en cas de contour doit être différente de 0, ça pourrait être $\left|\nabla f[m,n]\right|$ ou 1 ou 255
  - ✓ souvent on préférera conserver $\left|\nabla f[m,n]\right|$ le plus longtemps possible

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

Module du gradient : $\left|\nabla f[m,n]\right|$

Changement de dynamique de $\left|\nabla f[m,n]\right|$

$\left|\nabla f[m,n]\right|$ seuillée à 25

Source :

Gradient appliqué à l'image

Exemples

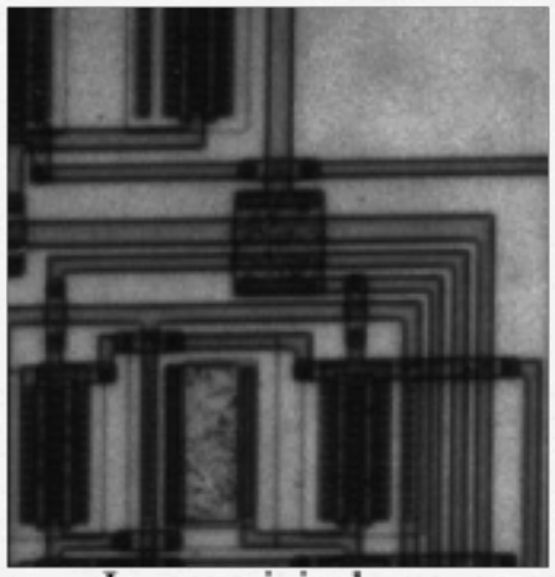
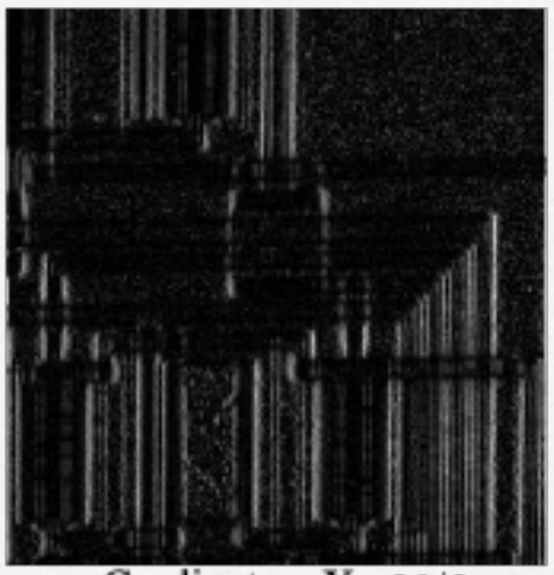
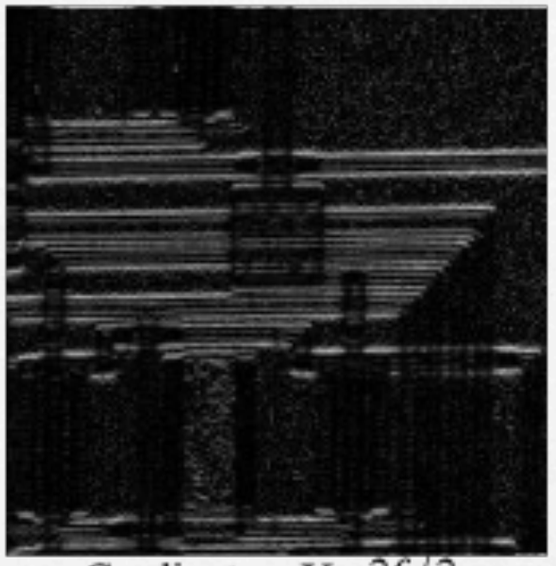


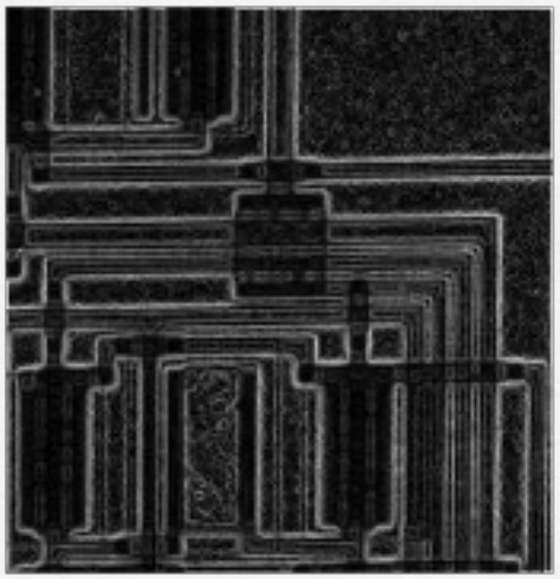
Image originale



Gradient en X : $\partial f / \partial x$



Gradient en Y : $\partial f / \partial y$



Magnitude : $\sqrt{(\partial f / \partial x)^2 + (\partial f / \partial y)^2}$

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

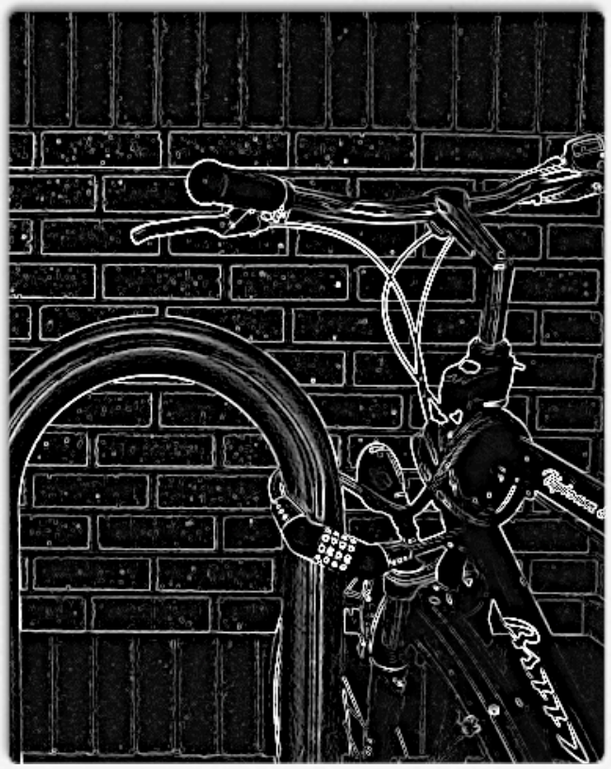
2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$
Différences finies,
schéma avant



©Marie-Flavie Auclair-Fortier

2. CONTOURS

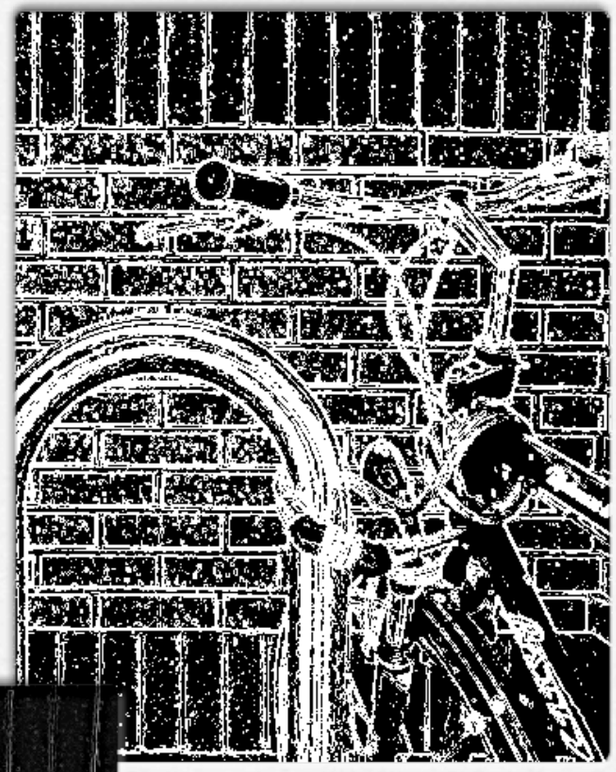
1. Détecteur gradient

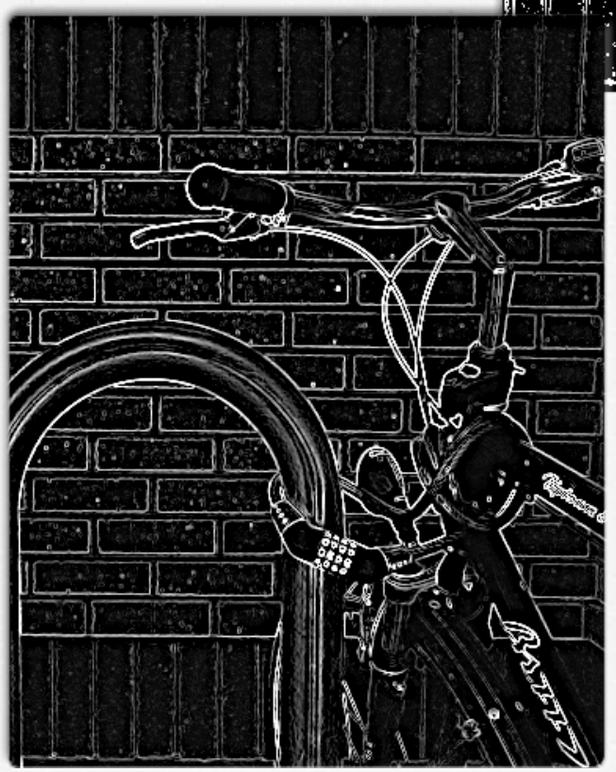
a) Seuillage de la norme du gradient

Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$
Différences finies,
schéma avant

seuil = 10





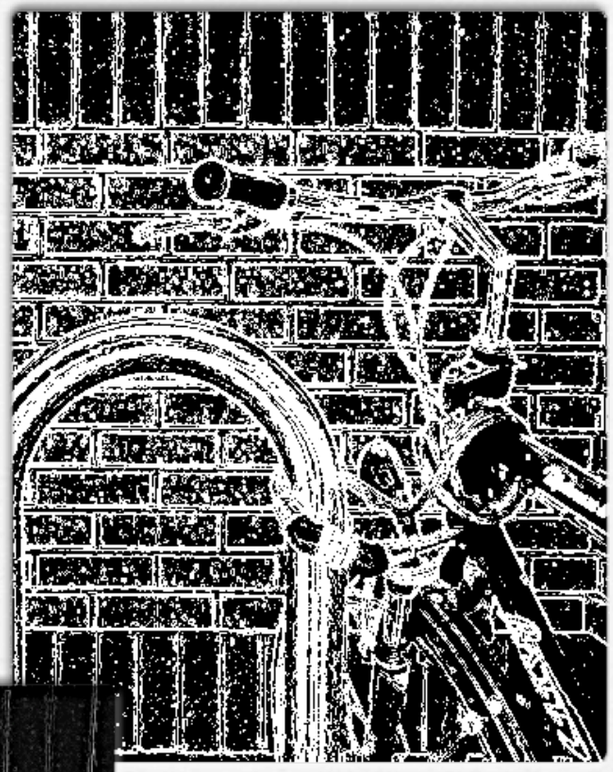
2. CONTOURS

1. Détecteur gradient

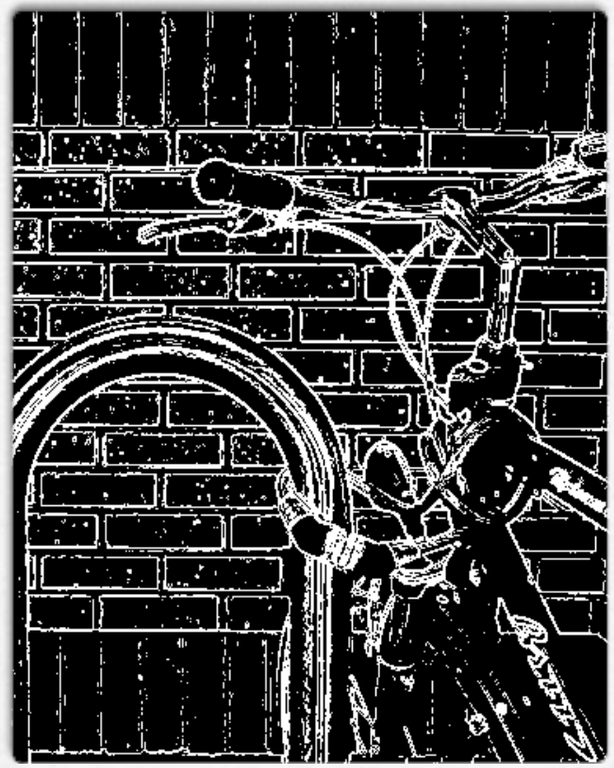
a) Seuillage de la norme du gradient

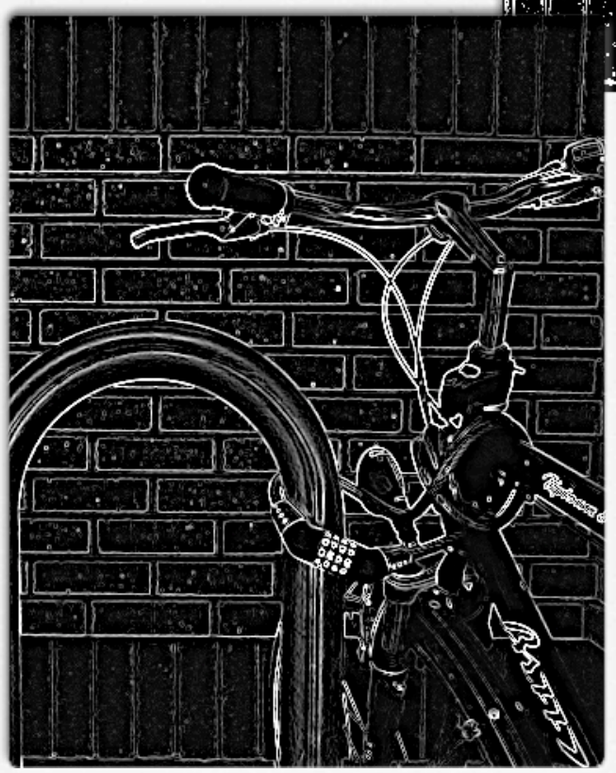
Image originale : $f[m, n]$

seuil = 10



seuil = 20





$\left| \nabla f[m, n] \right|$
Différences finies,
schéma avant

2. CONTOURS

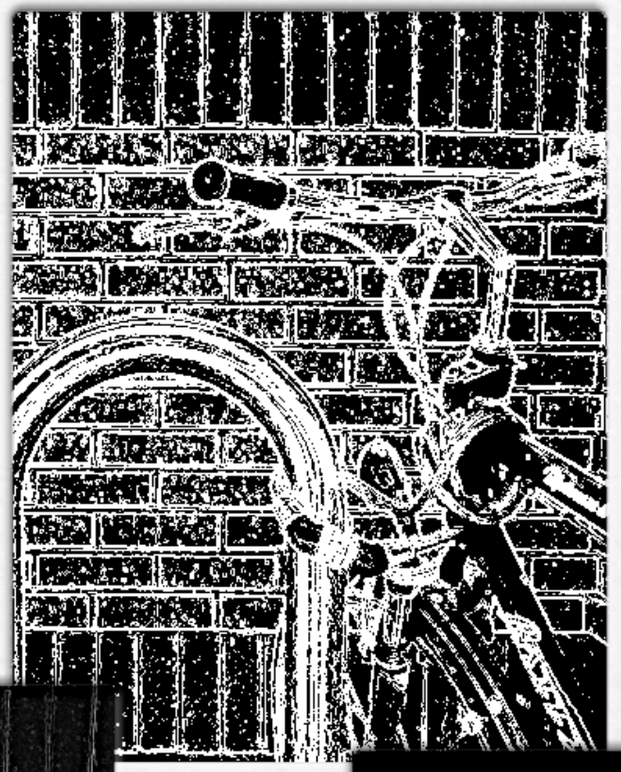
1. Détecteur gradient

a) Seuillage de la norme du gradient

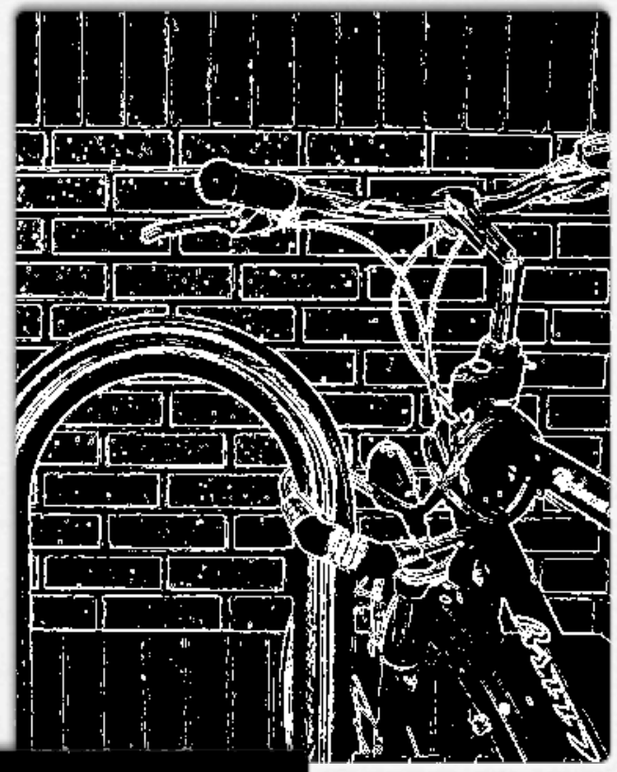
Image originale : $f[m, n]$

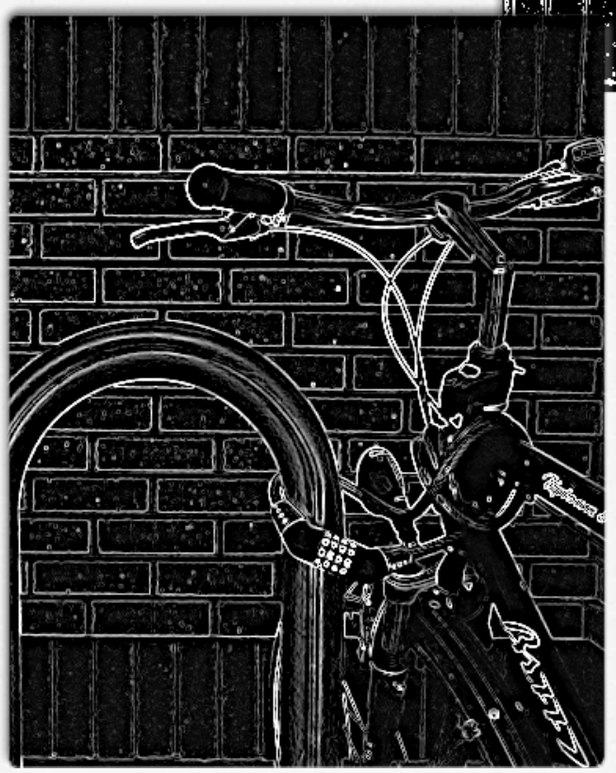
$\left| \nabla f[m, n] \right|$
Différences finies,
schéma avant

seuil = 10



seuil = 20





seuil = 50

nalyse d'images
©Marie-Flavie Auclair-Fortier

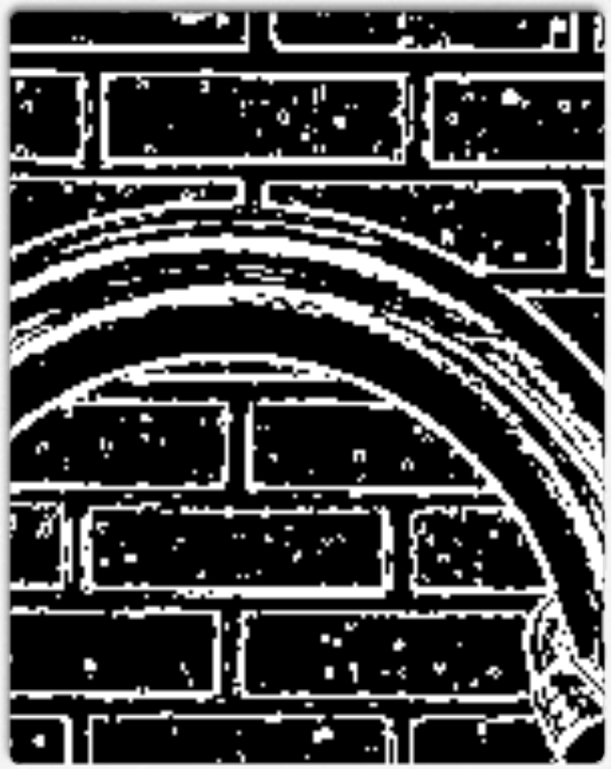
2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

- *Problèmes* (*différences finies*)
  - ✓ Méthode sensible au bruit : difficile d'éliminer le bruit sans éliminer les contours

seuil = 10



seuil = 20

seuil = 50

2. CONTOURS

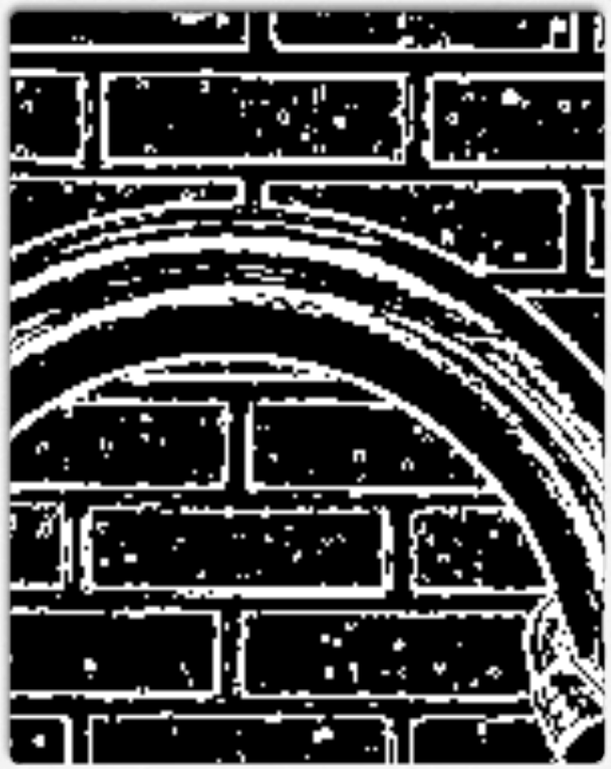
1. Détecteur gradient

Utilisation de Sobel
Plutôt que diff. finies

a) Seuillage de la norme du gradient

- *Problèmes* (*différences finies*)
  - ✓ Méthode sensible au bruit : difficile d'éliminer le bruit sans éliminer les contours

seuil = 10



seuil = 20

seuil = 50

2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$: Sobel

©Marie-Flavie Auclair-Fortier

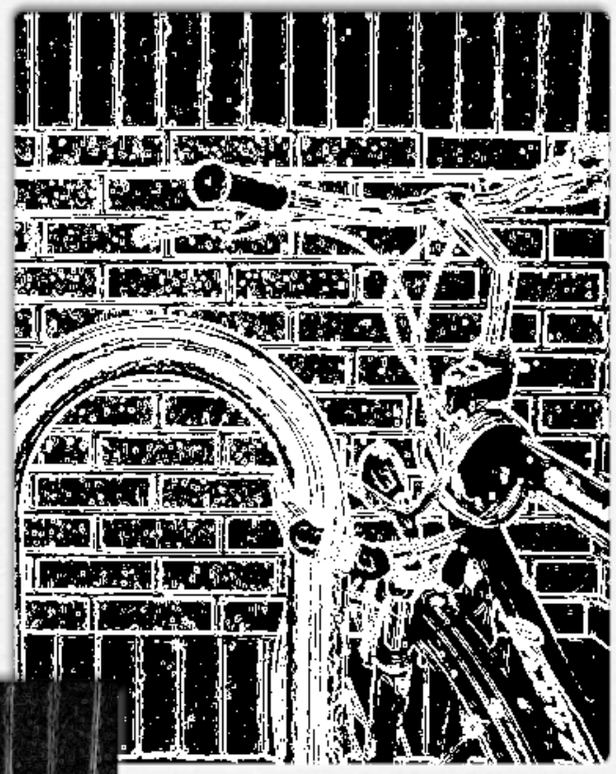
2. CONTOURS

1. Détecteur gradient

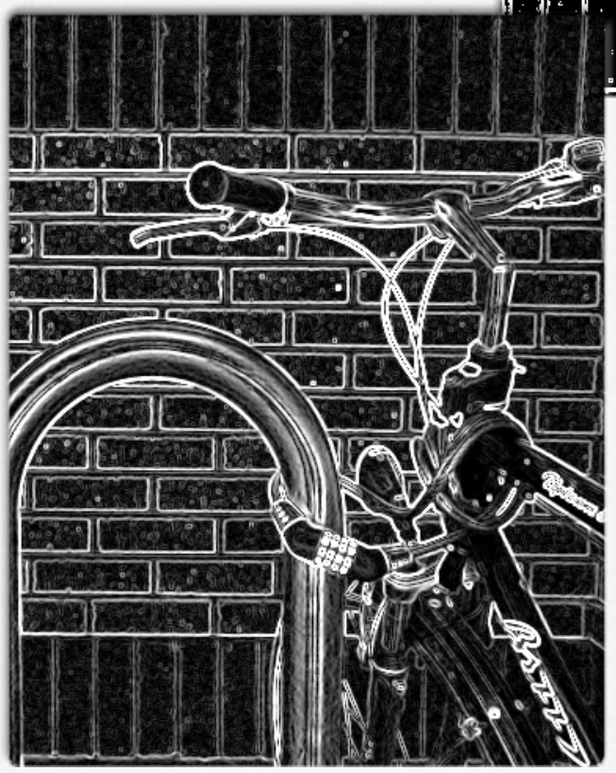
a) Seuillage de la norme du gradient

Image originale : $f[m, n]$

seuil = 40



$\left| \nabla f[m, n] \right|$: Sobel



©Marie-Flavie Auclair-Fortier

2. CONTOURS

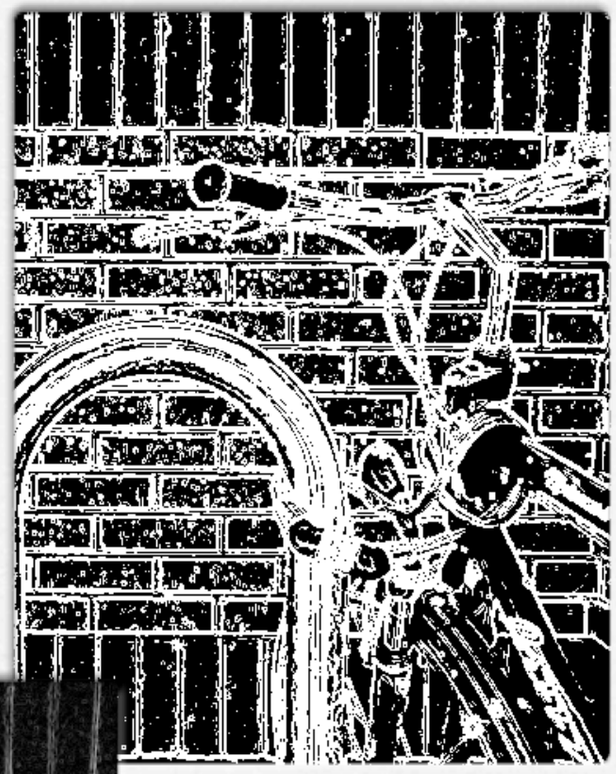
1. Détecteur gradient

a) Seuillage de la norme du gradient

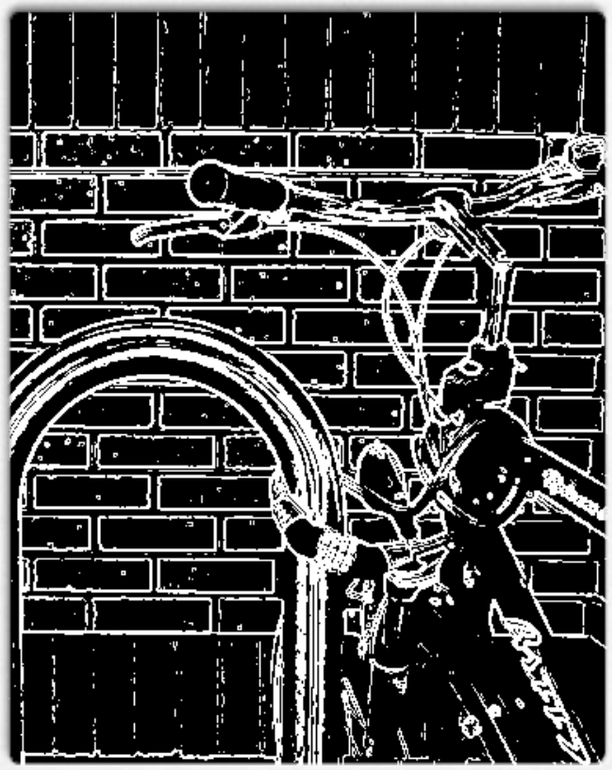
Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$: Sobel

seuil = 40



seuil = 90



©Marie-Flavie Auclair-Fortier

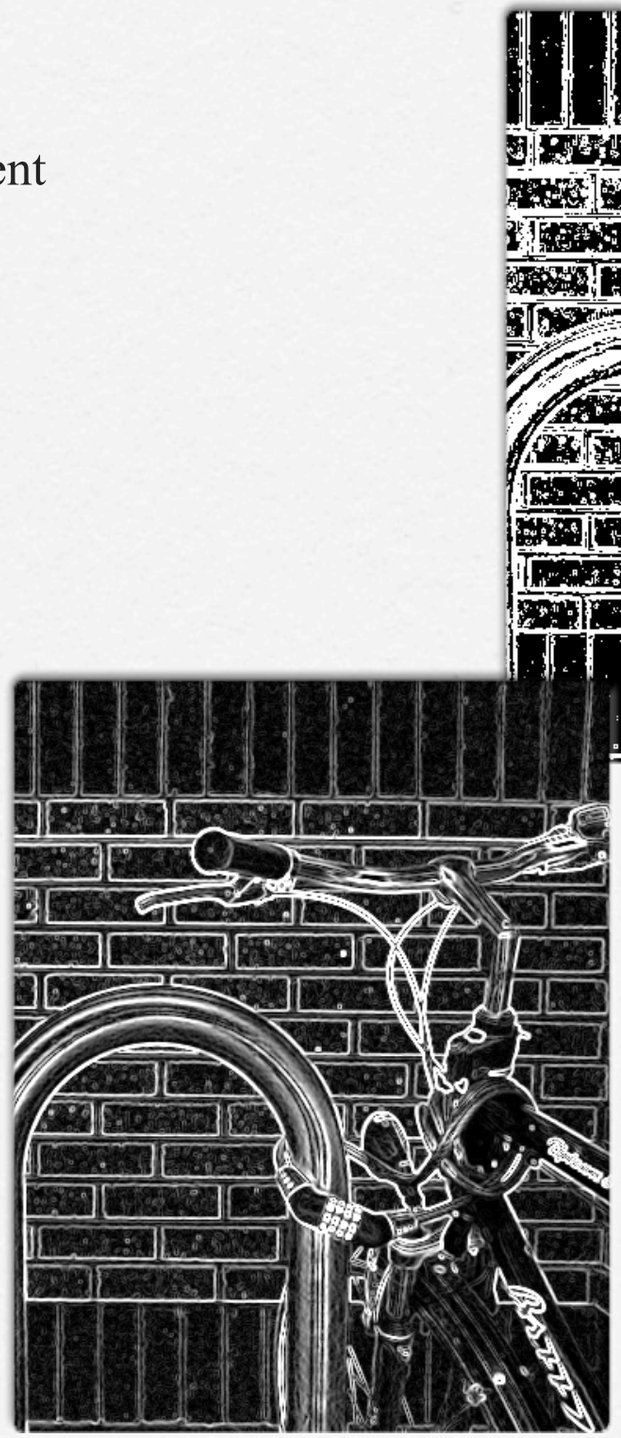
2. CONTOURS

1. Détecteur gradient

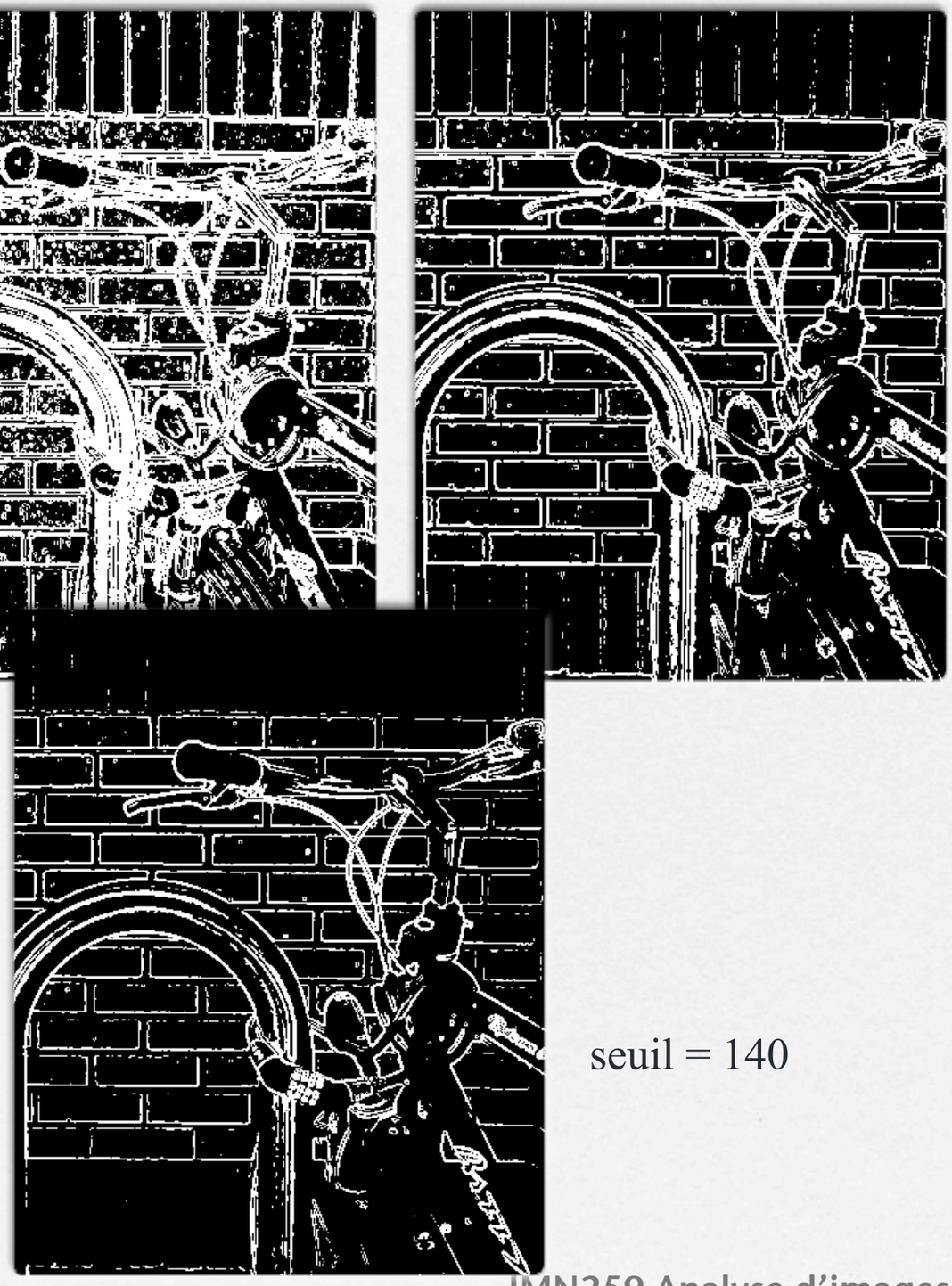
a) Seuillage de la norme du gradient

Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$: Sobel



seuil = 40



seuil = 90

seuil = 140

IMN259 Analyse d'images
©Marie-Flavie Auclair-Fortier

2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

- *Problèmes* (*Sobel*)

seuil = 40

2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

- *Problèmes* (*Sobel*)
  - ✓ Méthode sensible au bruit : moins pire mais toujours problématique

seuil = 40

2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

- *Problèmes* (*Sobel*)
  - ✓ Méthode sensible au bruit : moins pire mais toujours problématique

seuil = 40

seuil = 90

2. CONTOURS

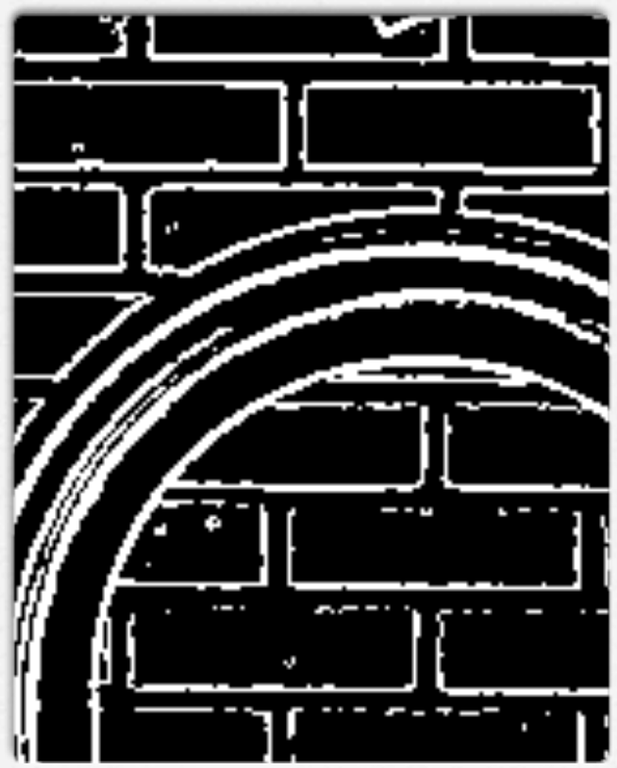
1. Détecteur gradient

a) Seuillage de la norme du gradient

- *Problèmes* (*Sobel*)
  - ✓ Méthode sensible au bruit : moins pire mais toujours problématique

seuil = 40

seuil = 90



seuil = 140

2. CONTOURS

1. Détecteur gradient

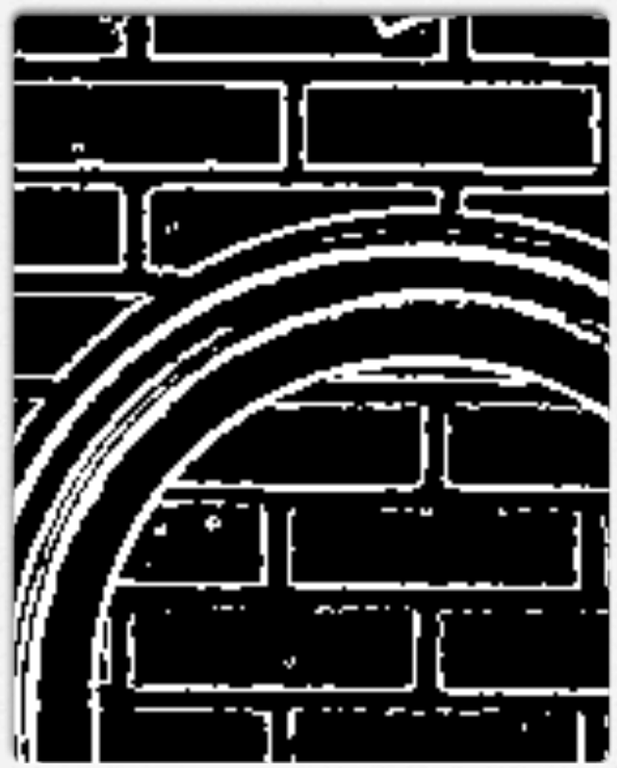
Utilisation du gradient de la Gaussienne

a) Seuillage de la norme du gradient

- *Problèmes* (*Sobel*)
  - ✓ Méthode sensible au bruit : moins pire mais toujours problématique

seuil = 40

seuil = 90



seuil = 140

2. CONTOURS

1. Détecteur gradient

a) Seuillage de la norme du gradient

Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$
gradient de la
gaussienne ($\sigma = 1$)

2. CONTOURS

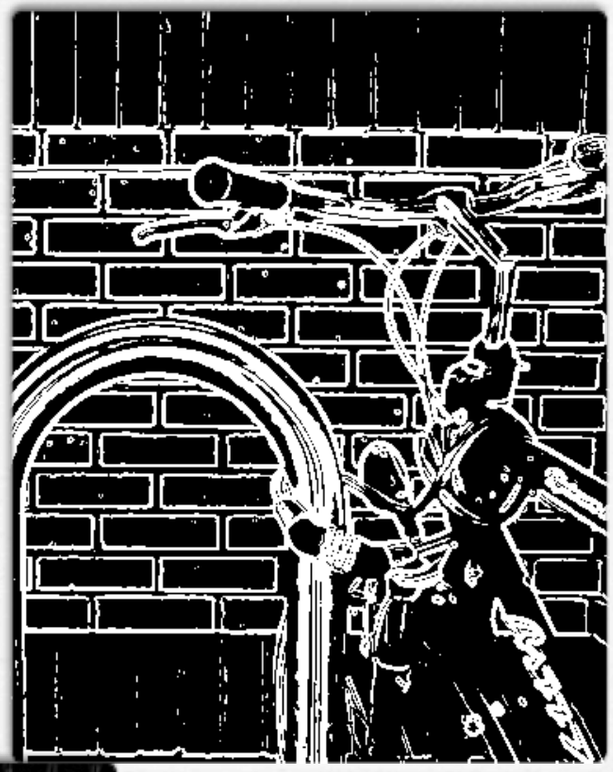
1. Détecteur gradient

a) Seuillage de la norme du gradient

Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$
gradient de la
gaussienne ($\sigma = 1$)

seuil = 40



2. CONTOURS

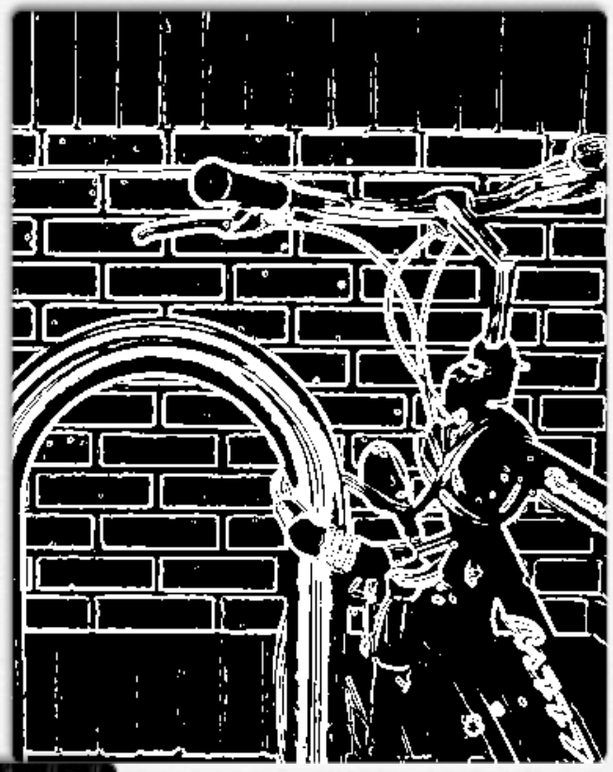
1. Détecteur gradient

a) Seuillage de la norme du gradient

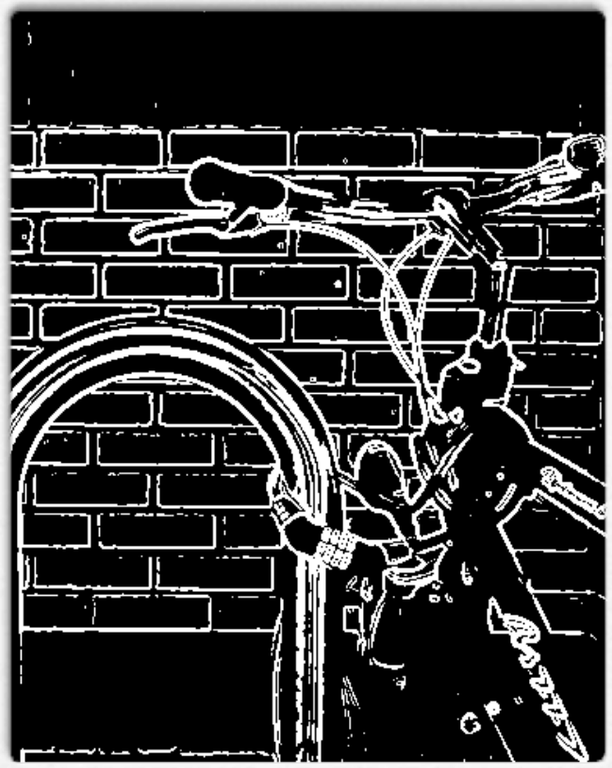
Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$
gradient de la
gaussienne ($\sigma = 1$)

seuil = 40



seuil = 60



©Marie-Flavie Auclair-Fortier

2. CONTOURS

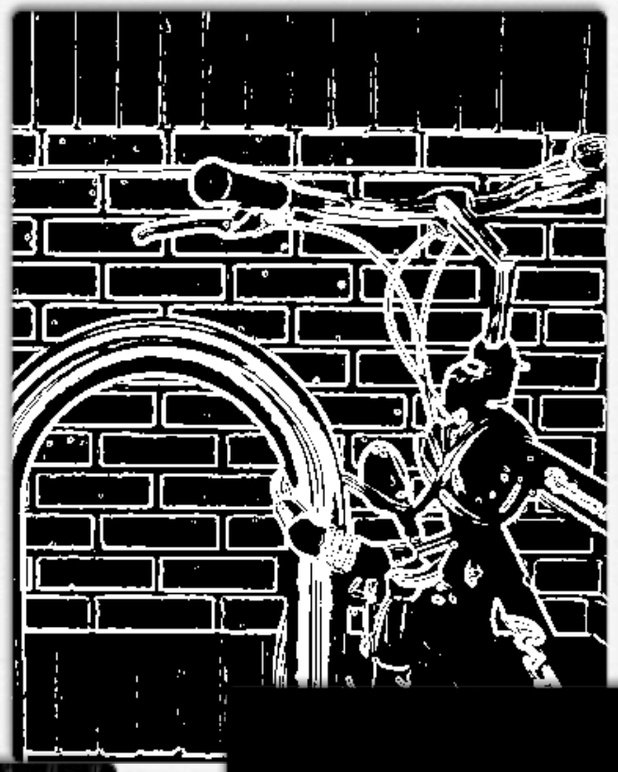
1. Détecteur gradient

a) Seuillage de la norme du gradient

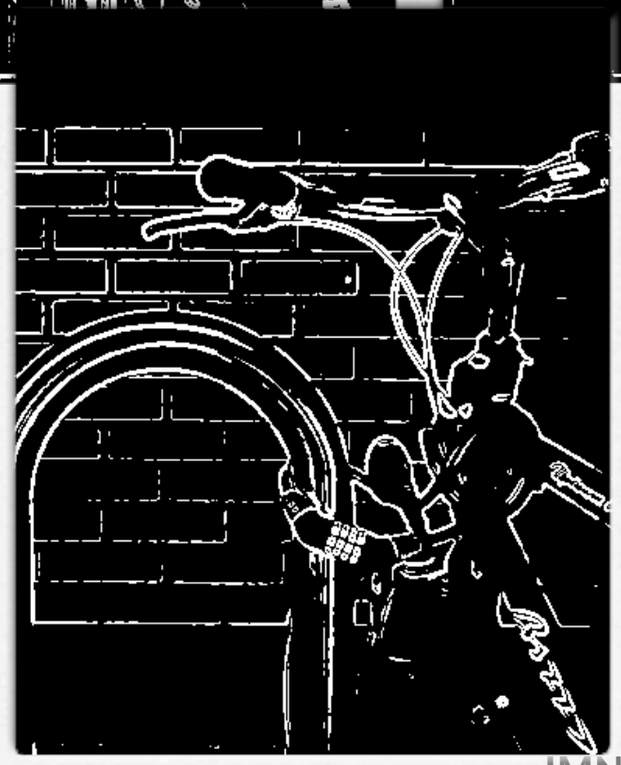
Image originale : $f[m, n]$

$\left| \nabla f[m, n] \right|$
gradient de la
gaussienne ($\sigma = 1$)

seuil = 40



seuil = 60



seuil = 90

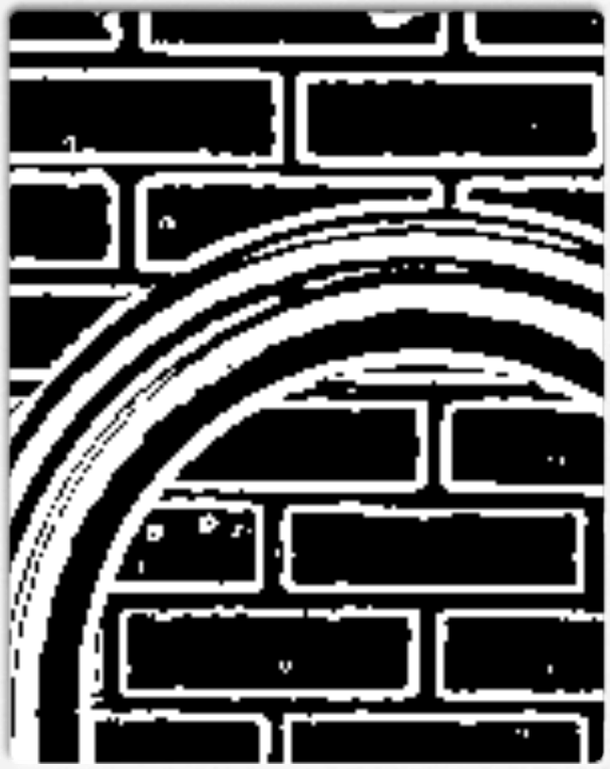
2. CONTOURS

1. Détecteur gradient

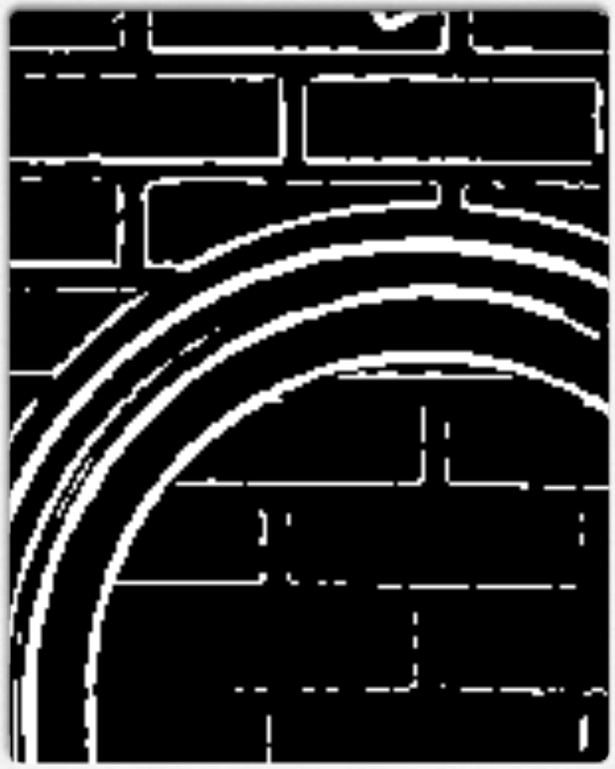
a) Seuillage de la norme du gradient gradient

- *Problèmes du seuillage* :

seuil = 40



seuil = 90



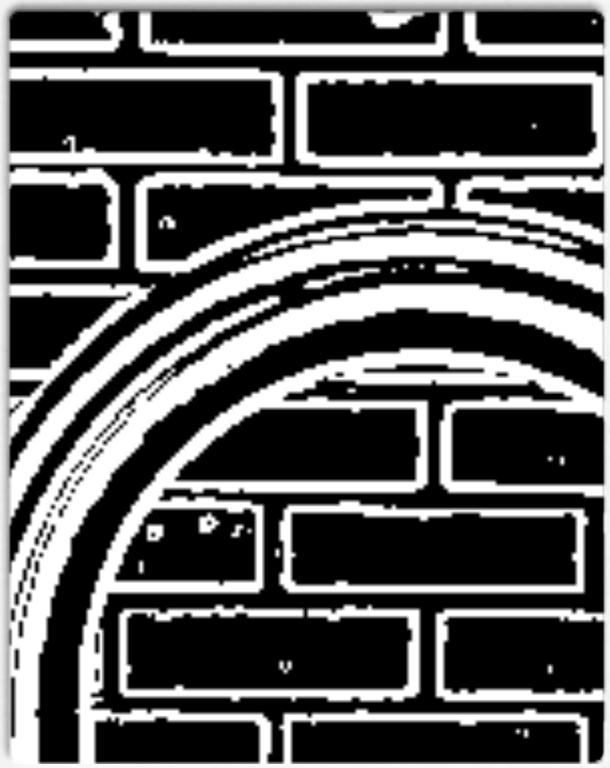
2. CONTOURS

1. Détecteur gradient

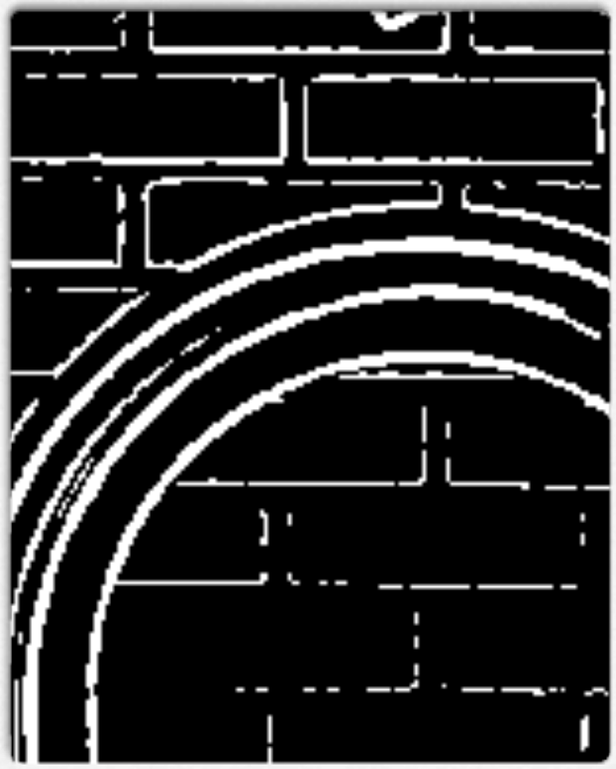
a) Seuillage de la norme du gradient gradient

- *Problèmes du seuillage* :
  - ✓ Seuillage est empirique (détermination du seuil par essai-erreur)

seuil = 40



seuil = 90



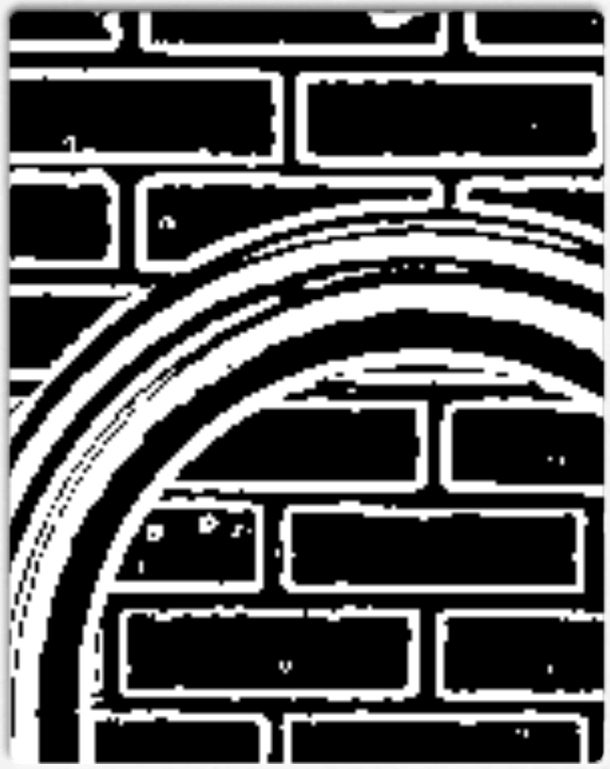
2. CONTOURS

1. Détecteur gradient

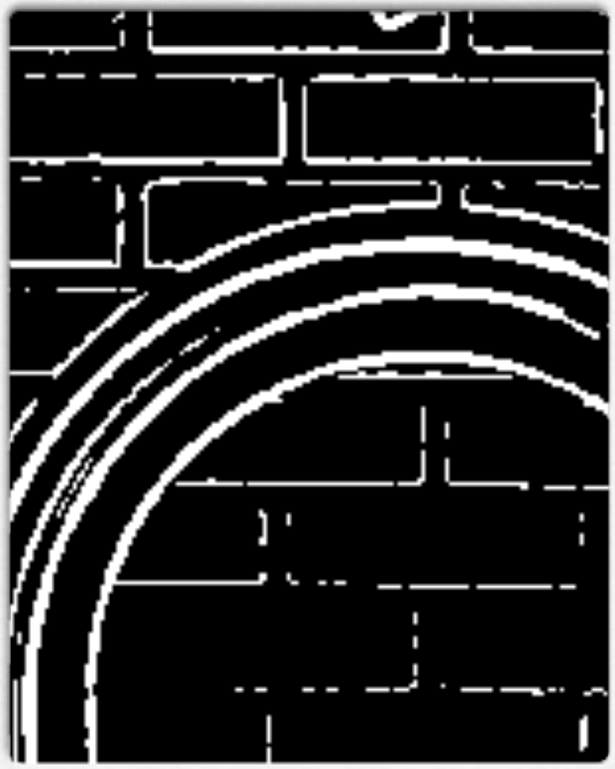
a) Seuillage de la norme du gradient gradient

- *Problèmes du seuillage* :
  - ✓ Seuillage est empirique (détermination du seuil par essai-erreur)
  - ✓ Apparition de pixels et segments isolés

seuil = 40



seuil = 90



©Marie-Flavie Auclair-Fortier

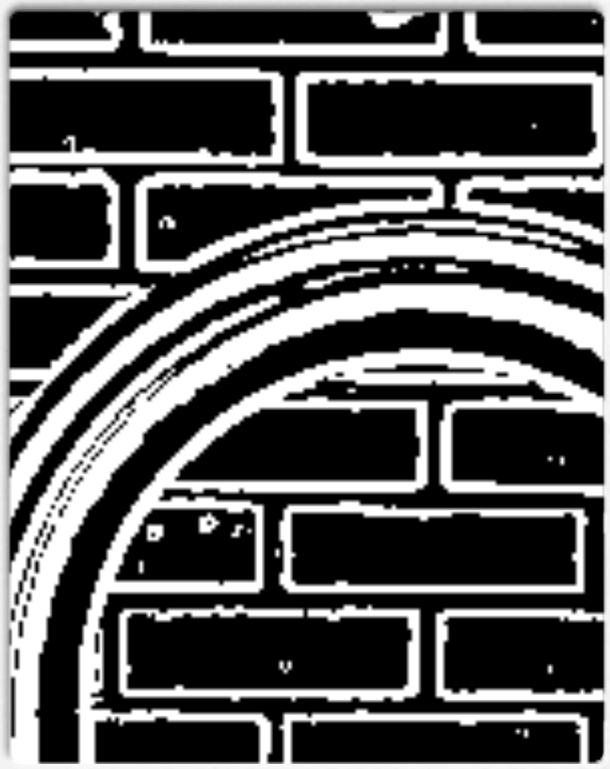
2. CONTOURS

1. Détecteur gradient

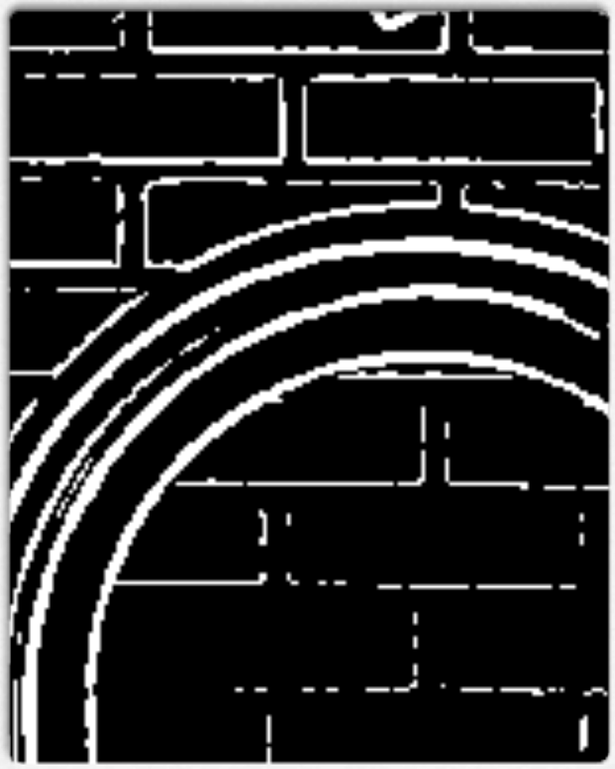
a) Seuillage de la norme du gradient gradient

- *Problèmes du seuillage* :
  - ✓ Seuillage est empirique (détermination du seuil par essai-erreur)
  - ✓ Apparition de pixels et segments isolés
  - ✓ Les contours ne sont pas filiformes (épaisseur > 1)

seuil = 40



seuil = 90



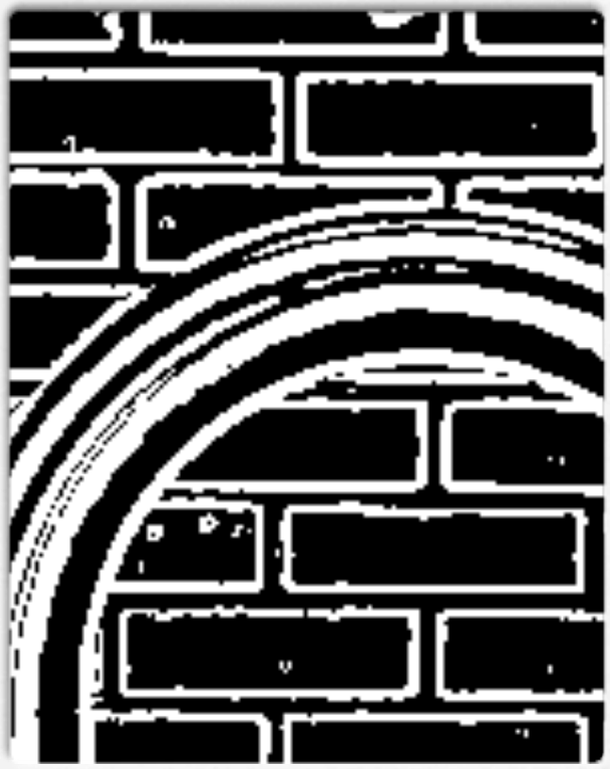
2. CONTOURS

1. Détecteur gradient

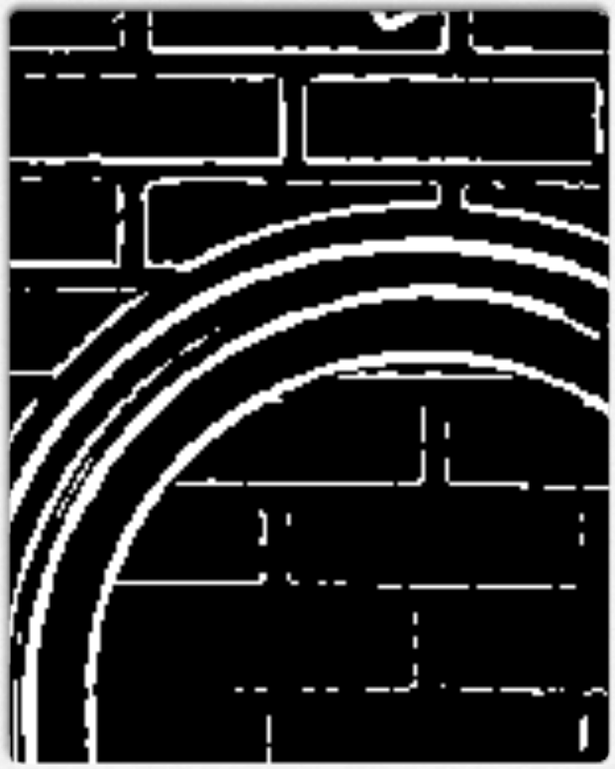
a) Seuillage de la norme du gradient gradient

- *Problèmes du seuillage* :
  - ✓ Seuillage est empirique (détermination du seuil par essai-erreur)
  - ✓ Apparition de pixels et segments isolés
  - ✓ Les contours ne sont pas filiformes (épaisseur > 1)
  - ✓ Aucune utilisation de l'orientation du gradient pour chercher le maximum

seuil = 40



seuil = 90



©Marie-Flavie Auclair-Fortier

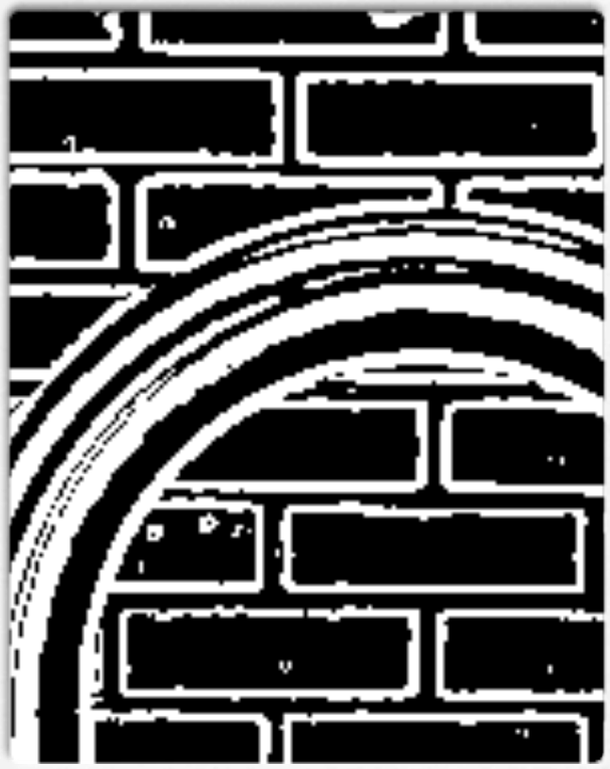
2. CONTOURS

1. Détecteur gradient

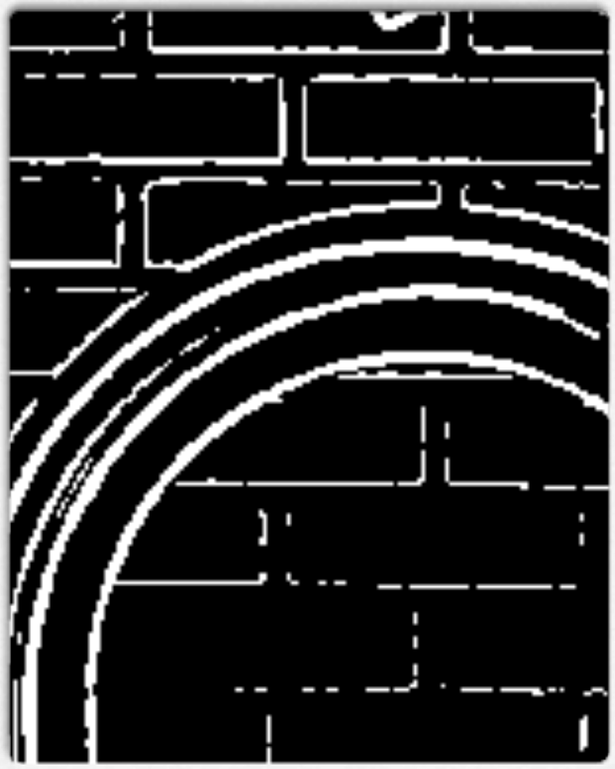
a) Seuillage de la norme du gradient gradient

- *Problèmes du seuillage* :
  - ✓ Seuillage est empirique (détermination du seuil par essai-erreur)
  - ✓ Apparition de pixels et segments isolés
  - ✓ Les contours ne sont pas filiformes (épaisseur > 1)
  - ✓ Aucune utilisation de l'orientation du gradient pour chercher le maximum

seuil = 40



seuil = 90



Remarque :

- Le seuillage du gradient ne fonctionne bien que pour des images sans bruit et avec de fortes discontinuités, peu importe les masques de dérivation utilisés

©Marie-Flavie Auclair-Fortier

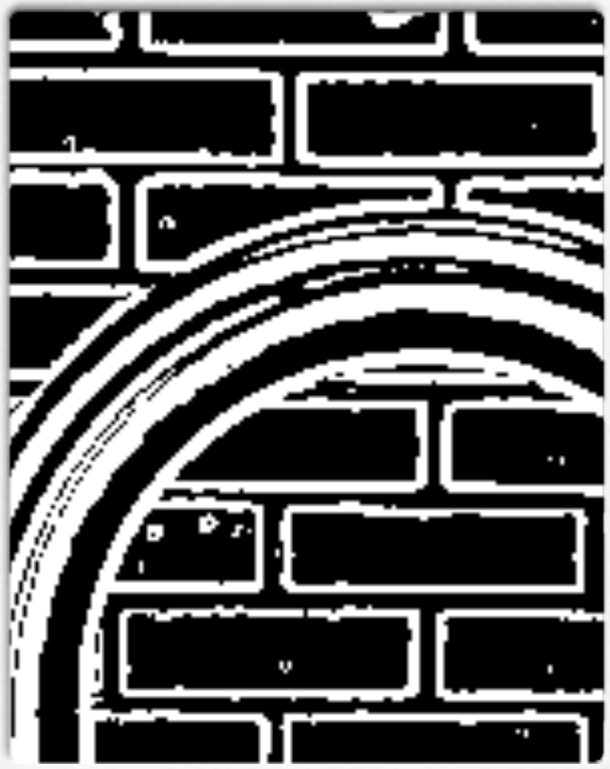
2. CONTOURS

1. Détecteur gradient

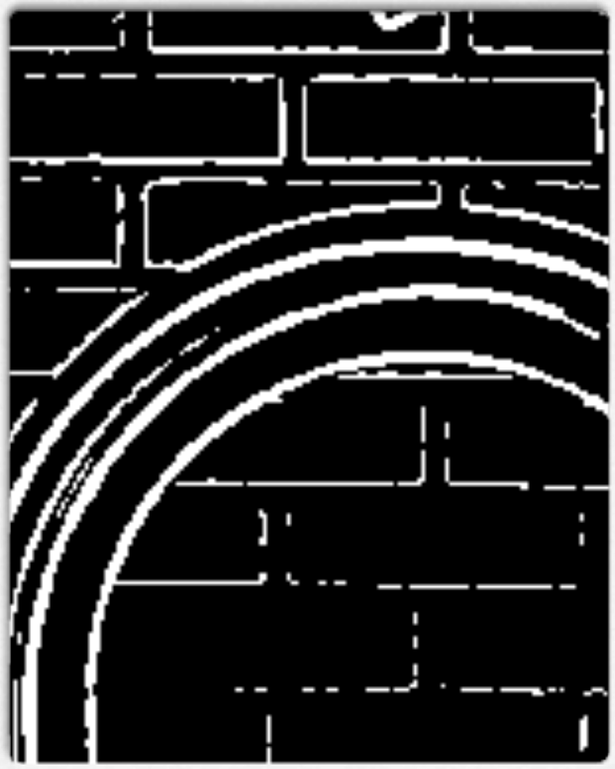
a) Seuillage de la norme du gradient gradient

- *Problèmes du seuillage* :
  - ✓ Seuillage est empirique (détermination du seuil par essai-erreur)
  - ✓ Apparition de pixels et segments isolés
  - ✓ Les contours ne sont pas filiformes (épaisseur > 1)
  - ✓ Aucune utilisation de l'orientation du gradient pour chercher le maximum

seuil = 40



seuil = 90



Remarque :

- Le seuillage du gradient ne fonctionne bien que pour des images sans bruit et avec de fortes discontinuités, peu importe les masques de dérivation utilisés

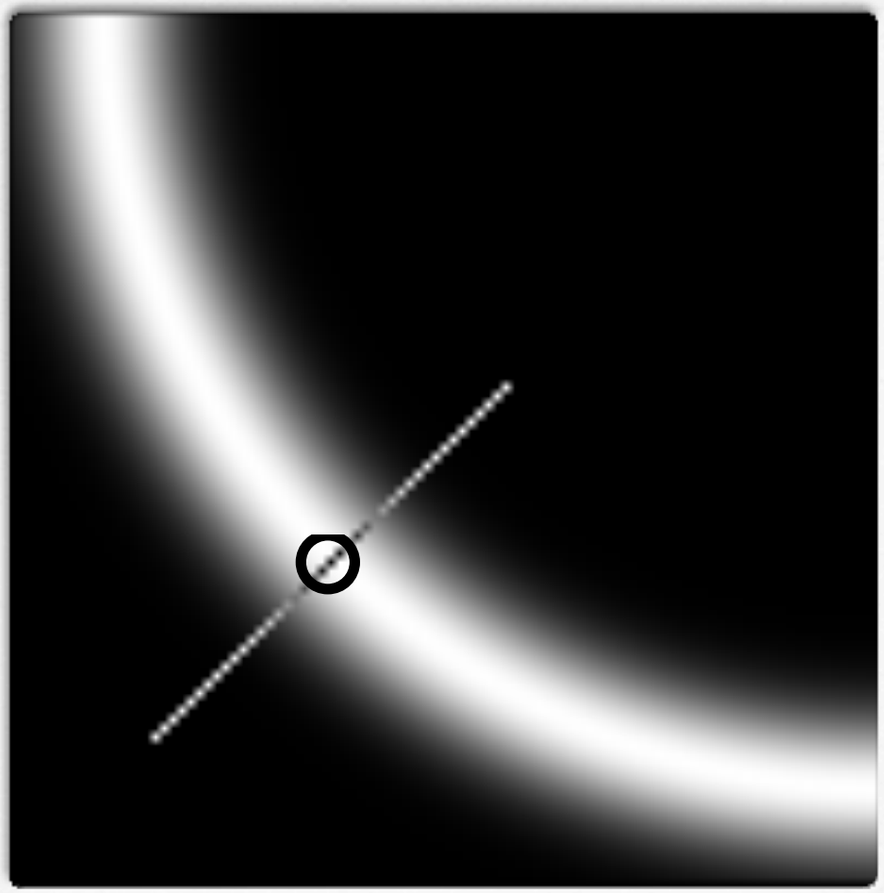
Suppression des non-maxima

2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- *Objectif* : N'identifier comme contour que les pixels où l'amplitude du gradient ($\left| \nabla f[m,n] \right|$) est maximal dans la direction du gradient ($\angle \nabla f[m,\, n]$)



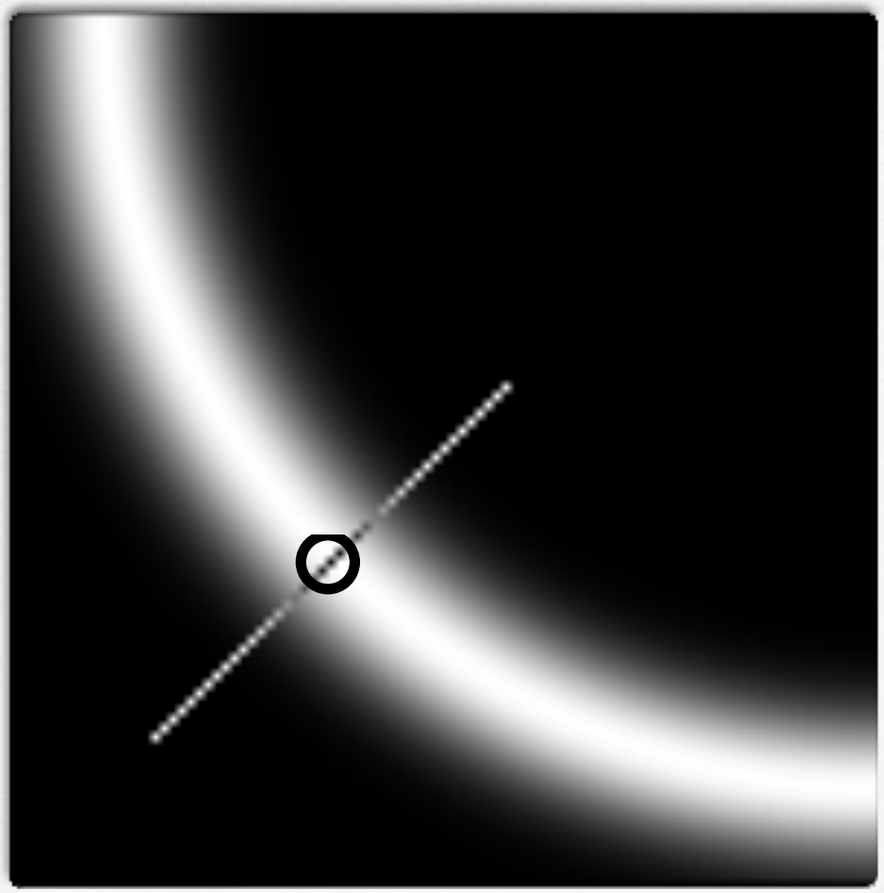
Source : Noah Snavely, CS6670: Computer Vision

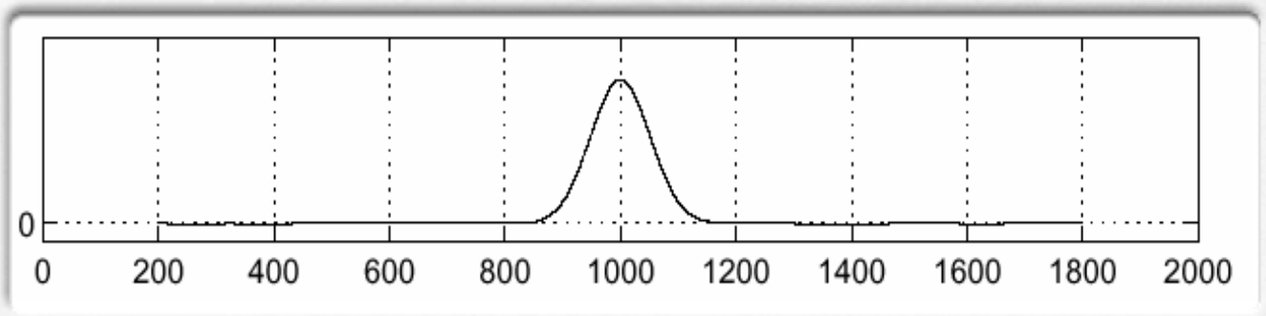
2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- *Objectif* : N'identifier comme contour que les pixels où l'amplitude du gradient ($\left| \nabla f[m,n] \right|$) est maximal dans la direction du gradient ($\angle \nabla f[m,\, n]$)





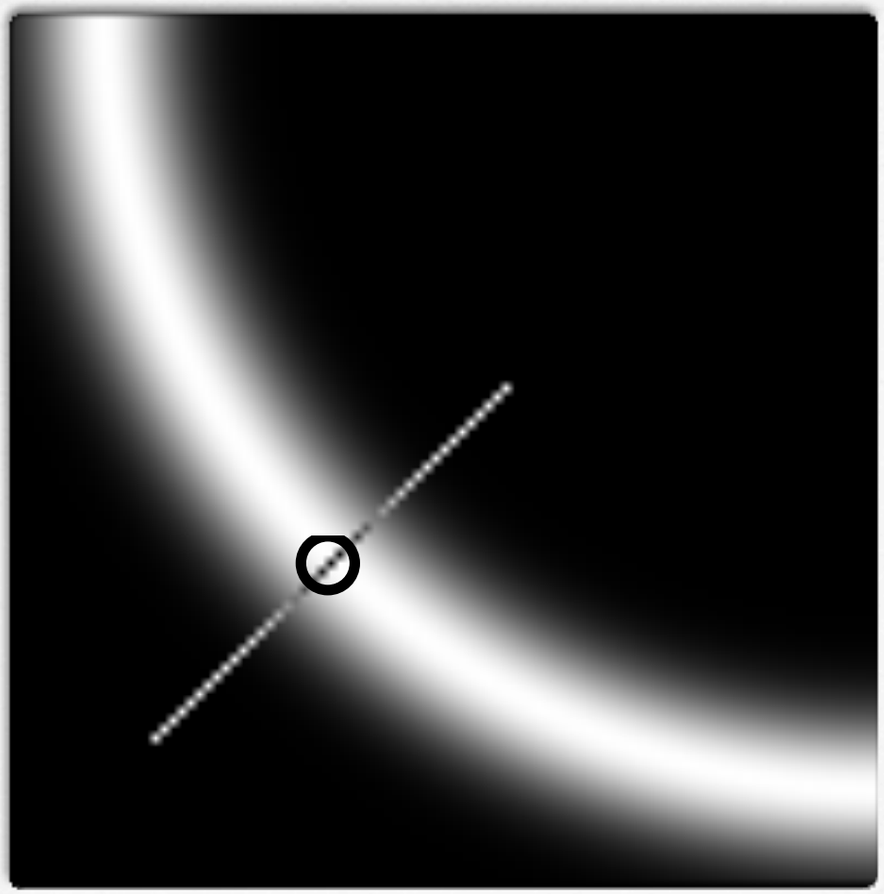
Source : Noah Snavely, CS6670: Computer Vision

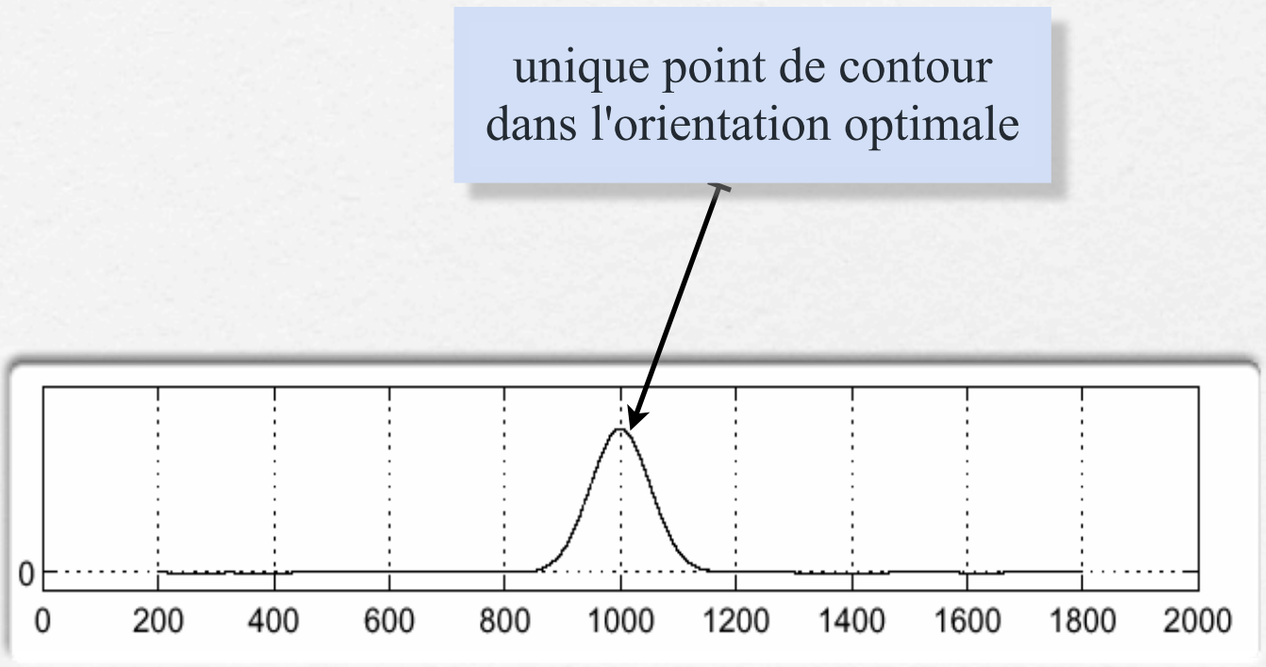
2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- *Objectif* : N'identifier comme contour que les pixels où l'amplitude du gradient ($\left| \nabla f[m,n] \right|$) est maximal dans la direction du gradient ($\angle \nabla f[m,\, n]$)





Source : Noah Snavely, CS6670: Computer Vision

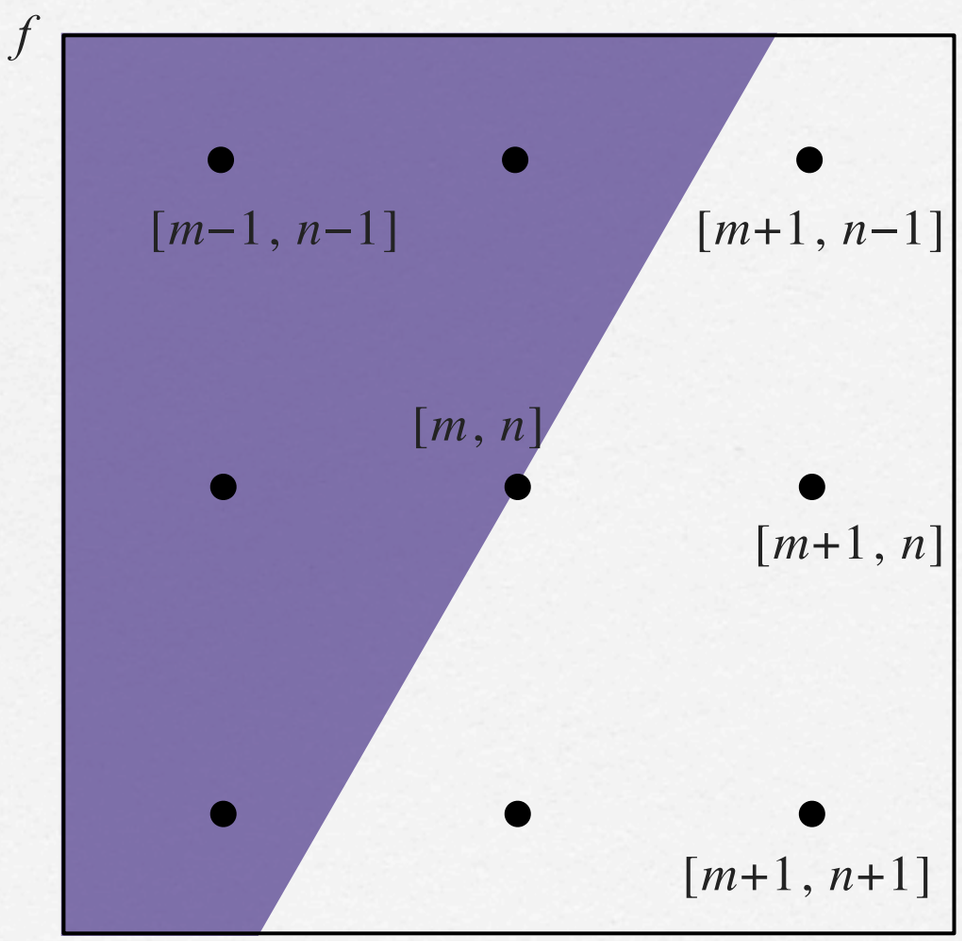
2. CONTOURS

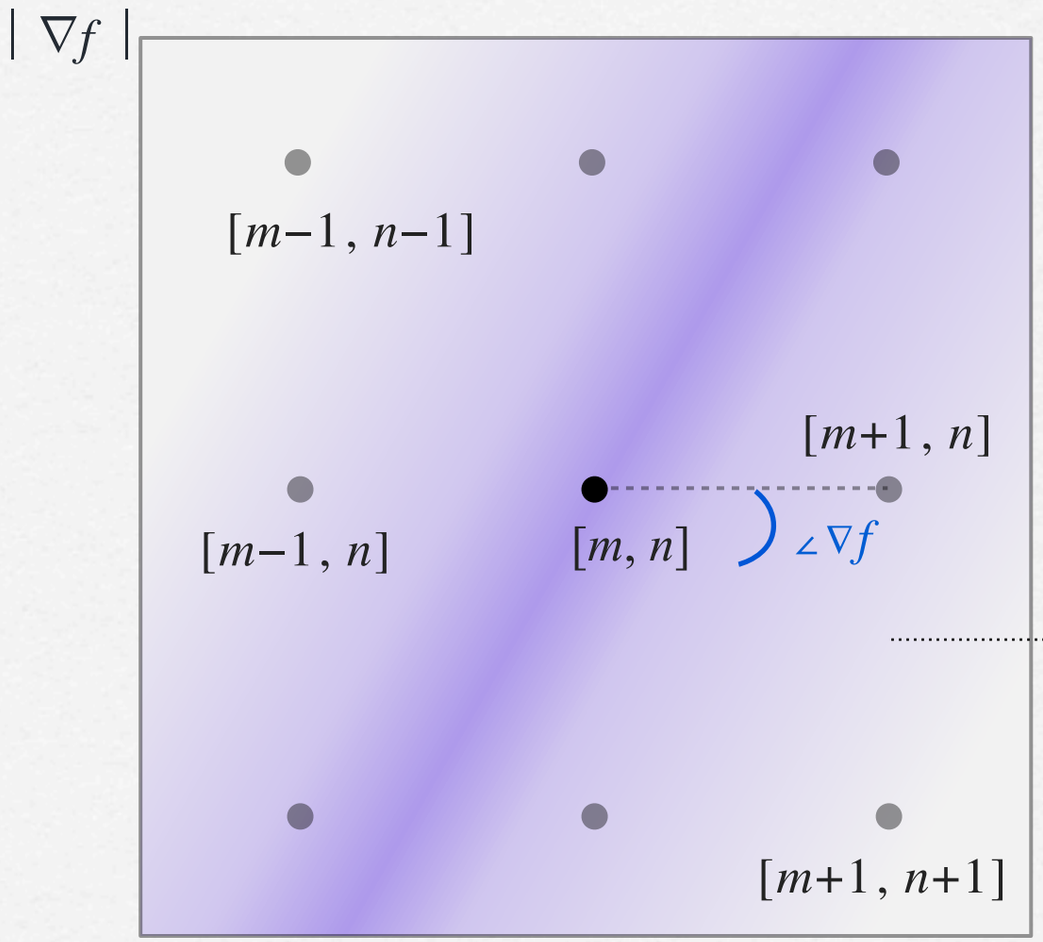
1. Détecteur gradient

b) Suppression des non-maxima

- On doit interpoler $|\nabla f(\mathbf{p}_1)|$ et $|\nabla f(\mathbf{p}_2)|$ car on ne les a pas directement
  - ✓ $\mathbf{p}_1$ et $\mathbf{p}_2$ risquent fort de tomber entre 2 pixels

- cas où $\theta[m, n] \in [-\pi/4, 0]$





Source :

©Marie-Flavie Auclair-Fortier

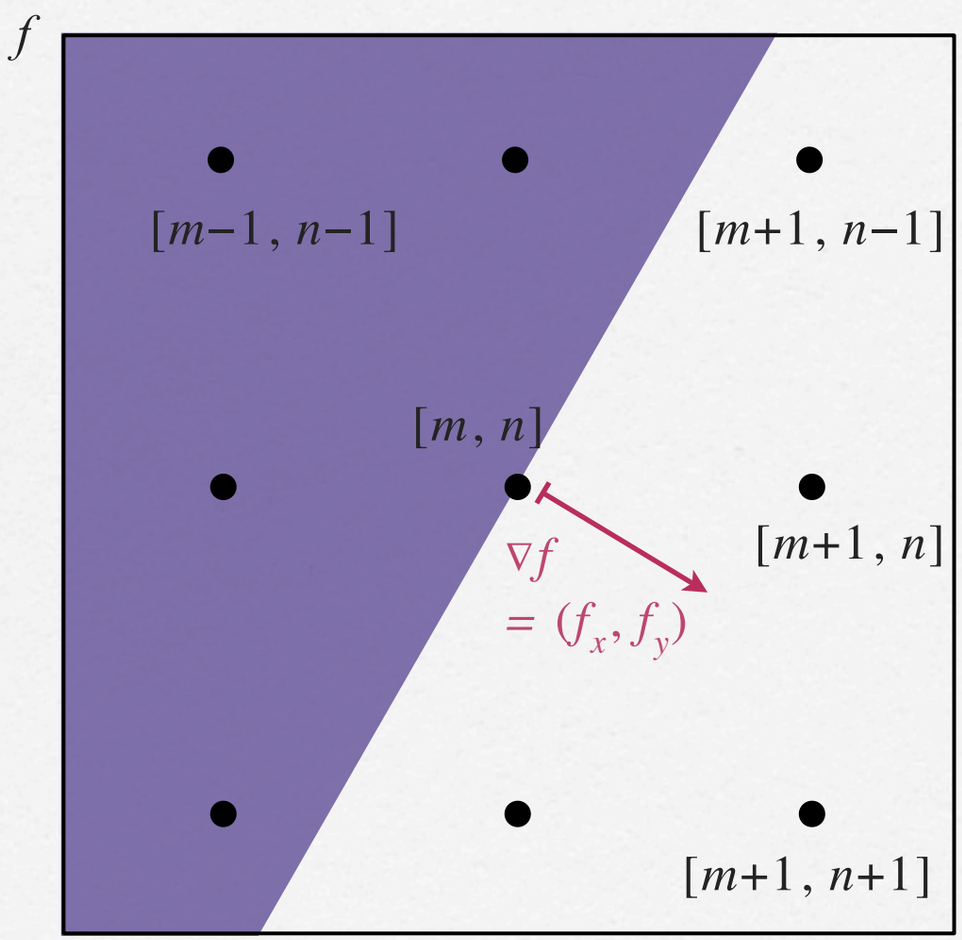
2. CONTOURS

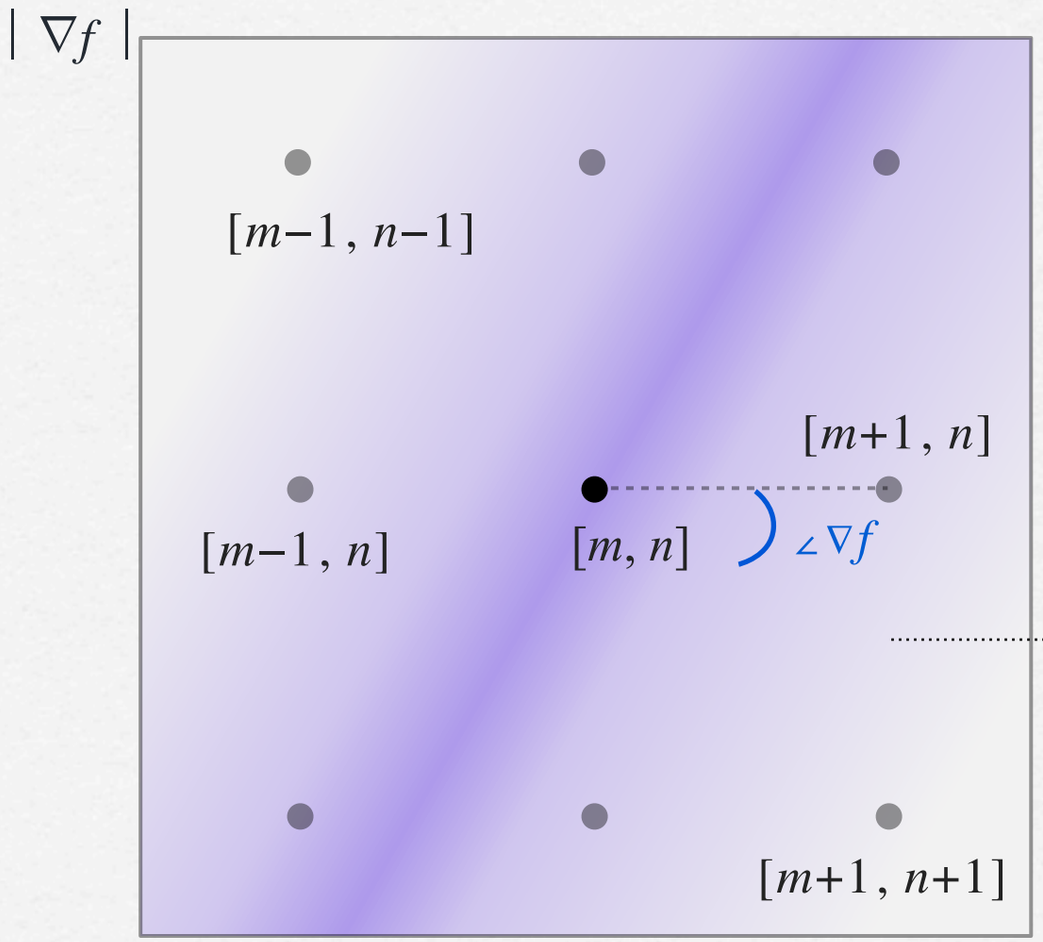
1. Détecteur gradient

b) Suppression des non-maxima

- On doit interpoler $|\nabla f(\mathbf{p}_1)|$ et $|\nabla f(\mathbf{p}_2)|$ car on ne les a pas directement
  - ✓ $\mathbf{p}_1$ et $\mathbf{p}_2$ risquent fort de tomber entre 2 pixels

- cas où $\theta[m, n] \in [-\pi/4, 0]$





Source :

©Marie-Flavie Auclair-Fortier

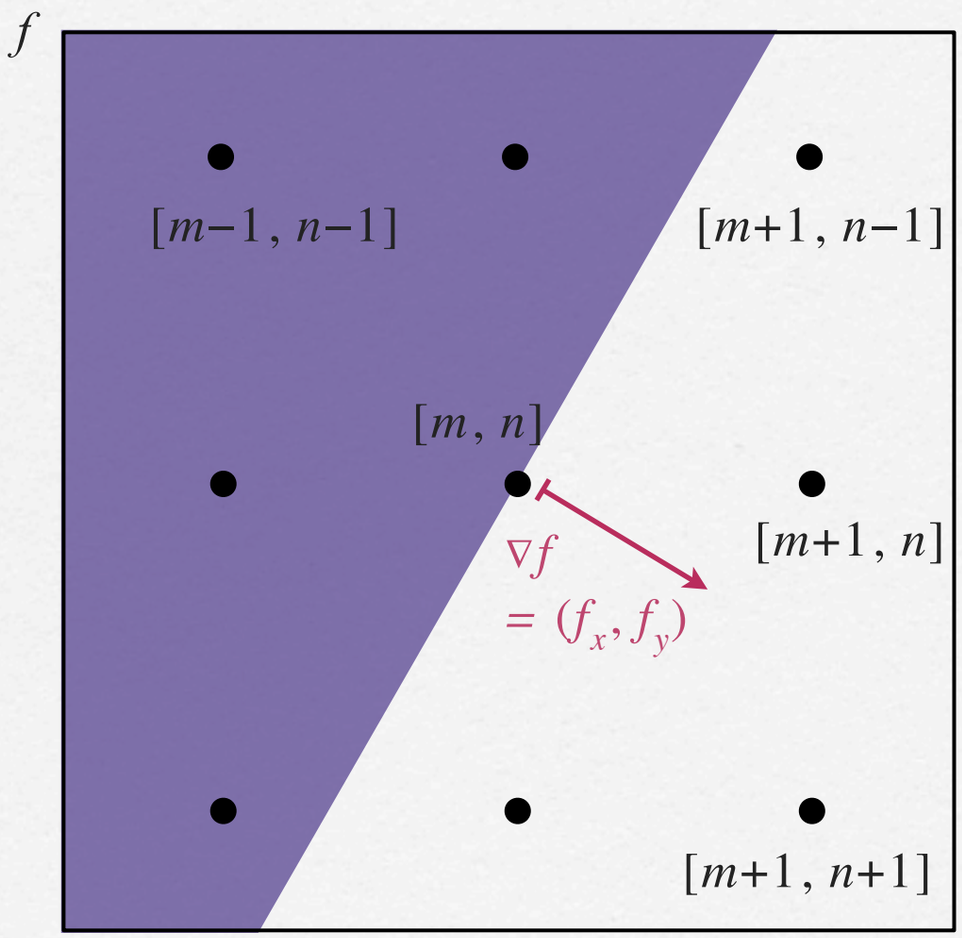
2. CONTOURS

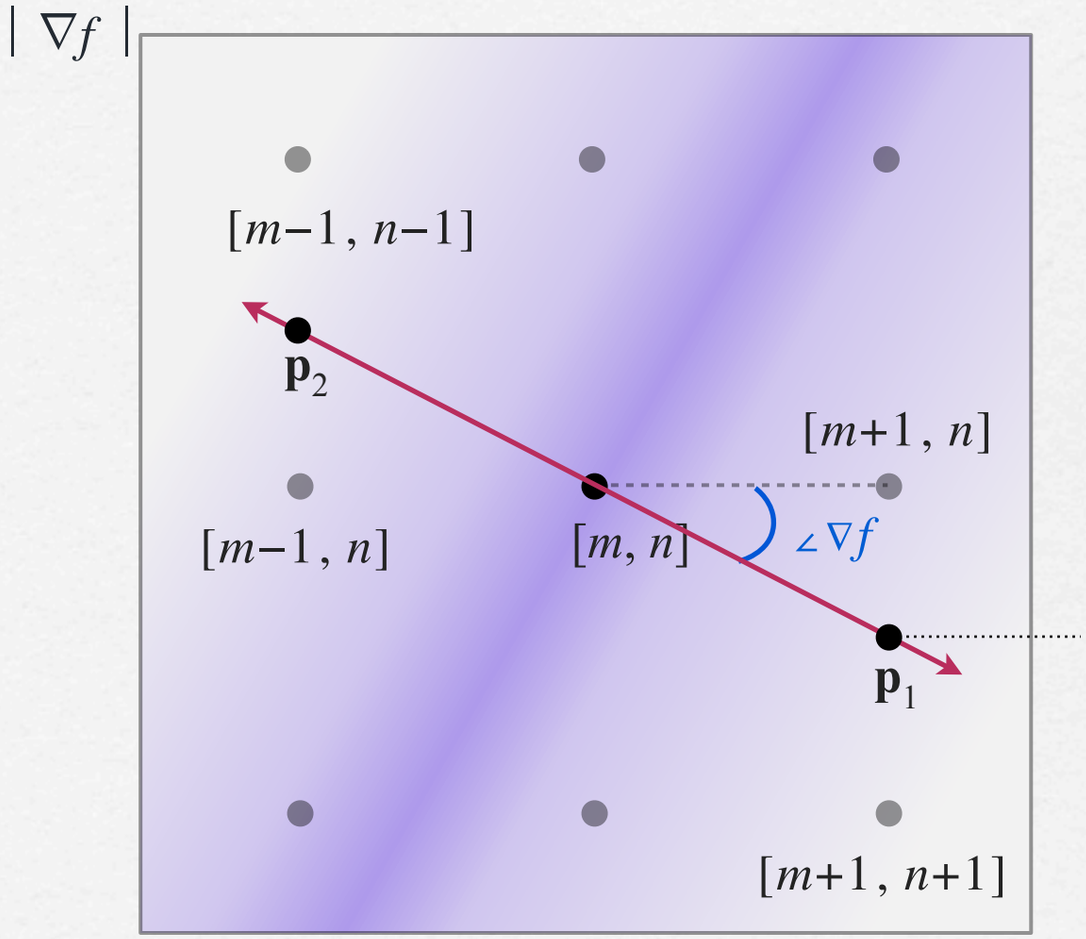
1. Détecteur gradient

b) Suppression des non-maxima

- cas où $\theta[m, n] \in [-\pi/4, 0]$

- On doit interpoler $|\nabla f(\mathbf{p}_1)|$ et $|\nabla f(\mathbf{p}_2)|$ car on ne les a pas directement
  - ✓ $\mathbf{p}_1$ et $\mathbf{p}_2$ risquent fort de tomber entre 2 pixels





Source :

©Marie-Flavie Auclair-Fortier

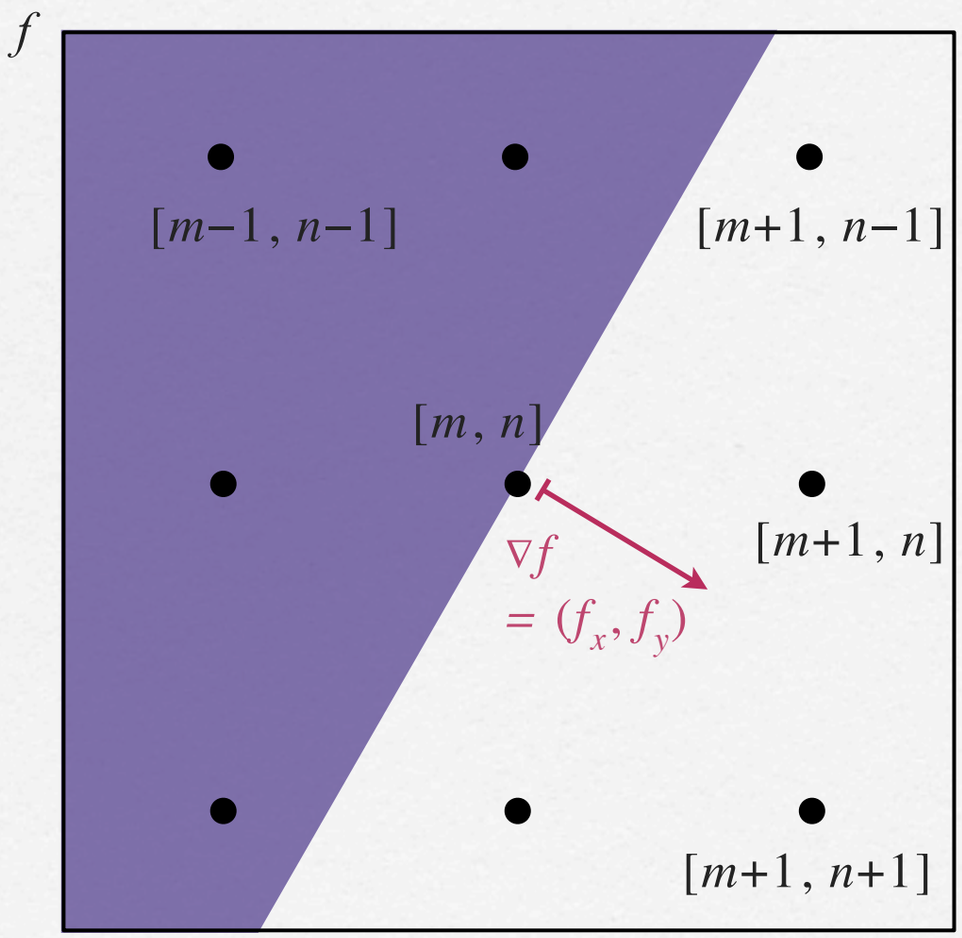
2. CONTOURS

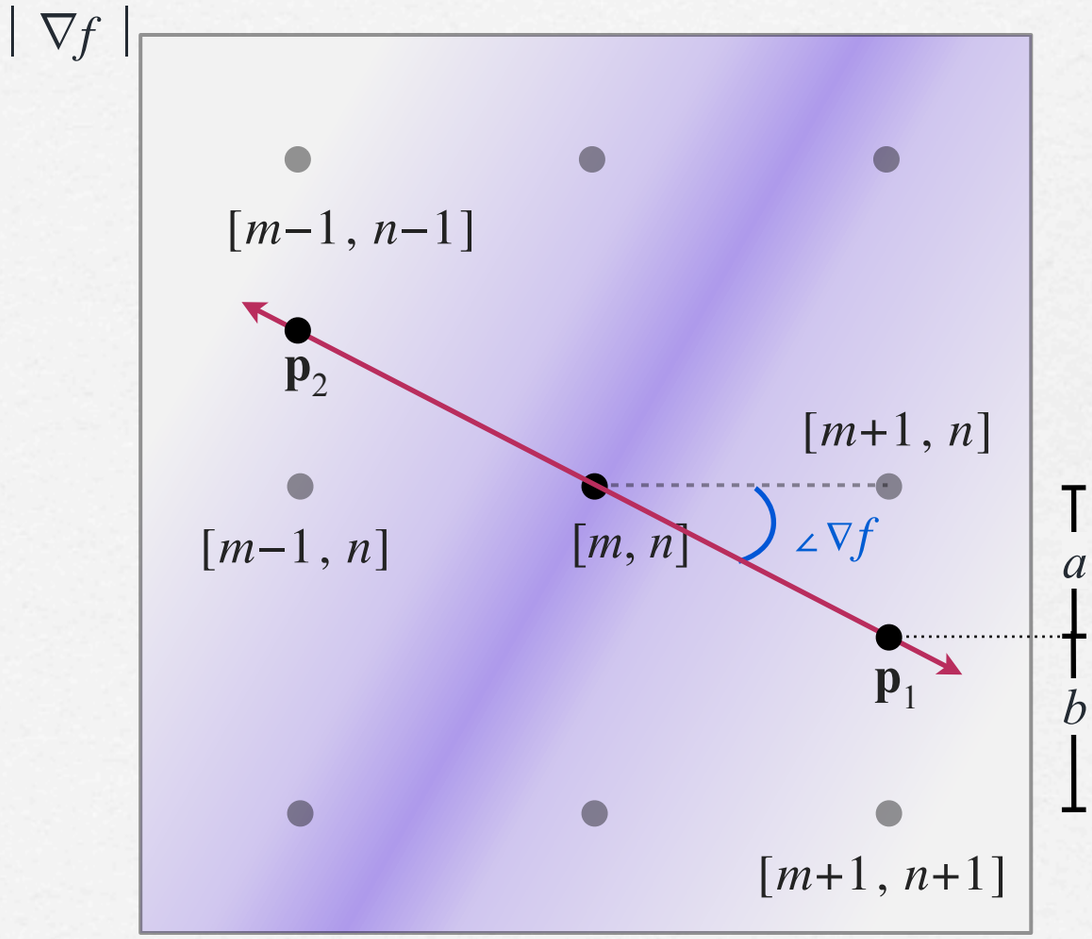
1. Détecteur gradient

b) Suppression des non-maxima

- cas où $\theta[m, n] \in [-\pi/4, 0]$

- On doit interpoler $|\nabla f(\mathbf{p}_1)|$ et $|\nabla f(\mathbf{p}_2)|$ car on ne les a pas directement
  - ✓ $\mathbf{p}_1$ et $\mathbf{p}_2$ risquent fort de tomber entre 2 pixels





Source :

2. CONTOURS

1. Détecteur gradient

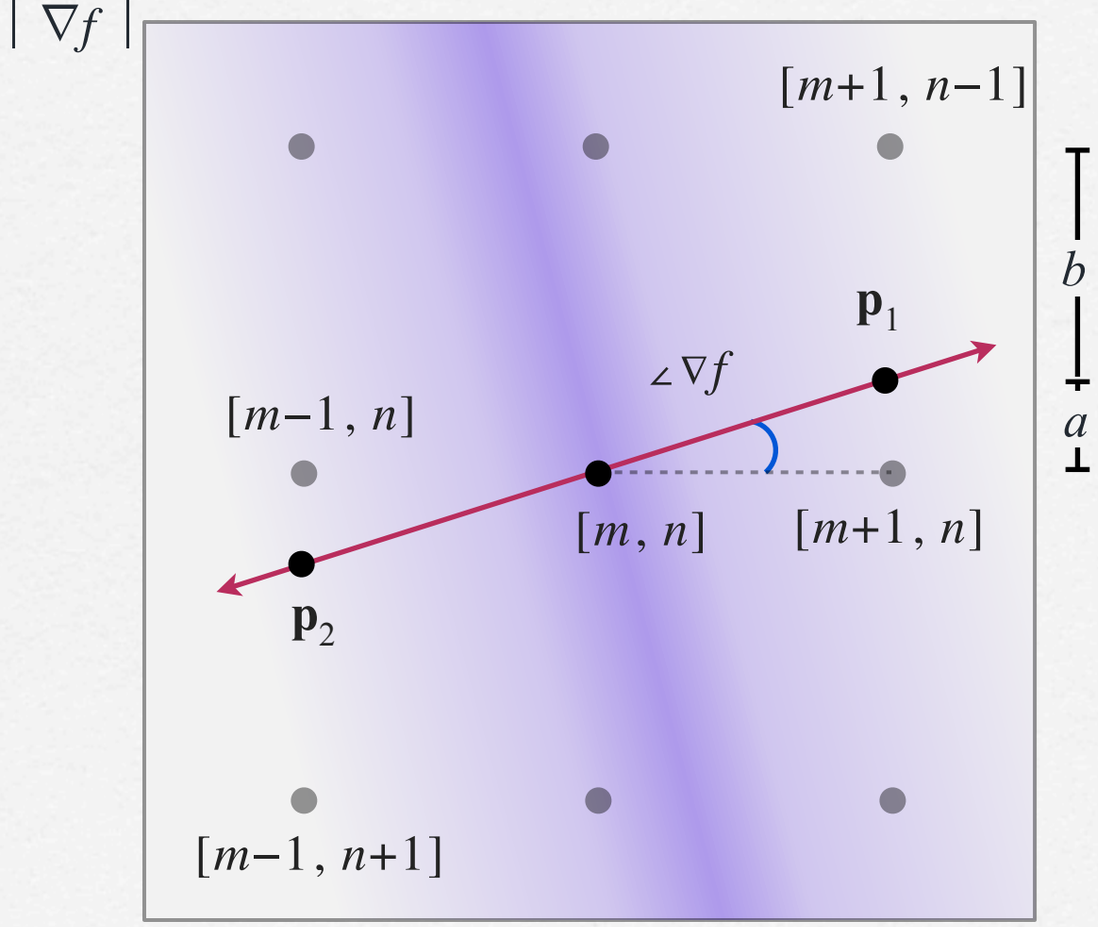
b) Suppression des non-maxima

pour trouver $\angle \nabla f[m, n]$

utiliser `atan` plutôt que `atan2`

- *attention* au dénominateur à 0

- cas où $\angle \nabla f[m, n] \in [0, \pi/4]$



Source :

©Marie-Flavie Auclair-Fortier

2. CONTOURS

1. Détecteur gradient

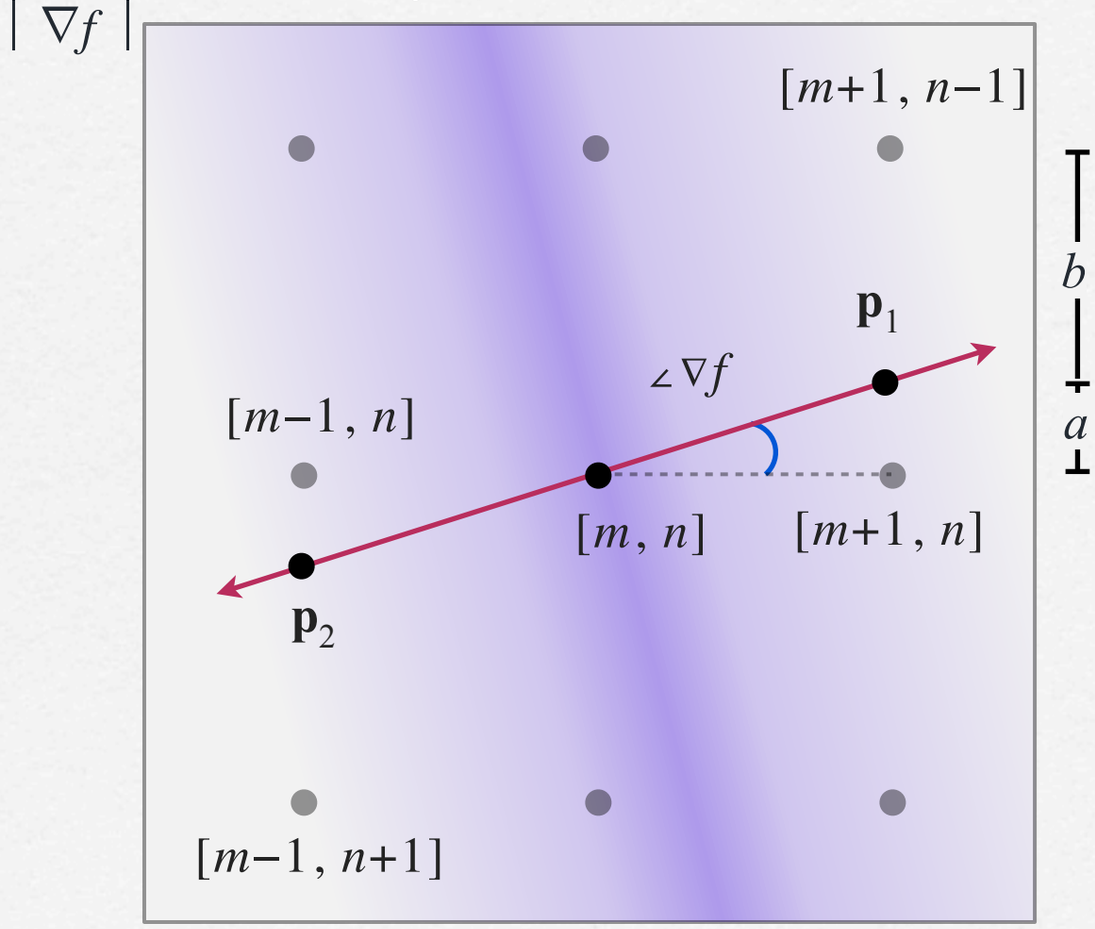
b) Suppression des non-maxima

pour trouver $\angle\nabla f[m, n]$

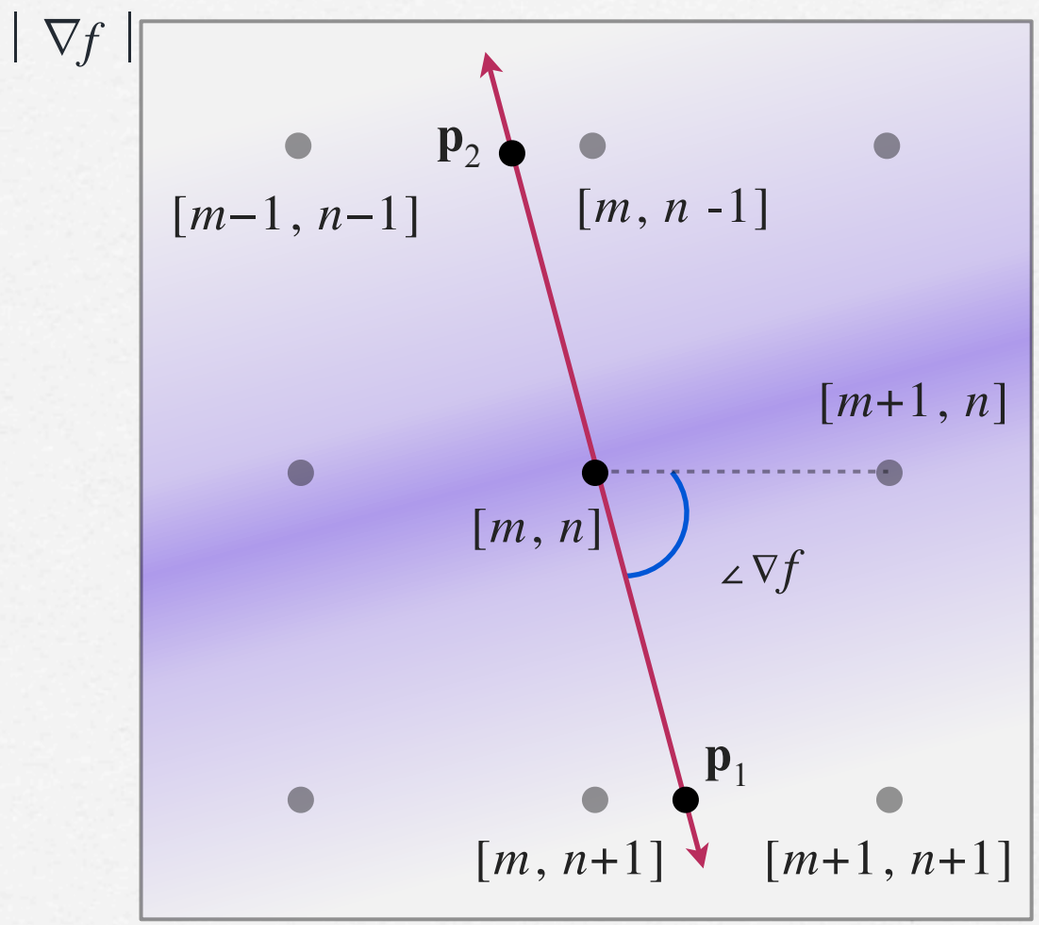
utiliser `atan` plutôt que `atan2`

- *attention* au dénominateur à 0

- cas où $\angle\nabla f[m, n] \in [0, \pi/4]$



- cas où $\angle\nabla f[m, n] \in [-\pi/2, -\pi/4]$



Source :

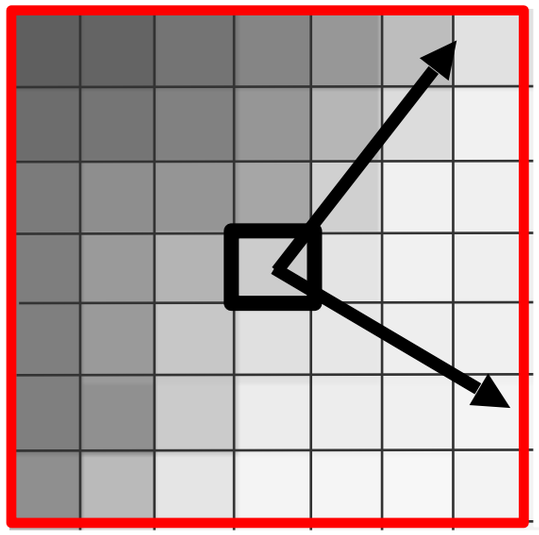
©Marie-Flavie Auclair-Fortier

Suppression des non-max (*thinning*)

Rendre les contours filiformes

$f(x,y)$

Direction du contour
(perpendiculaire à la normale)



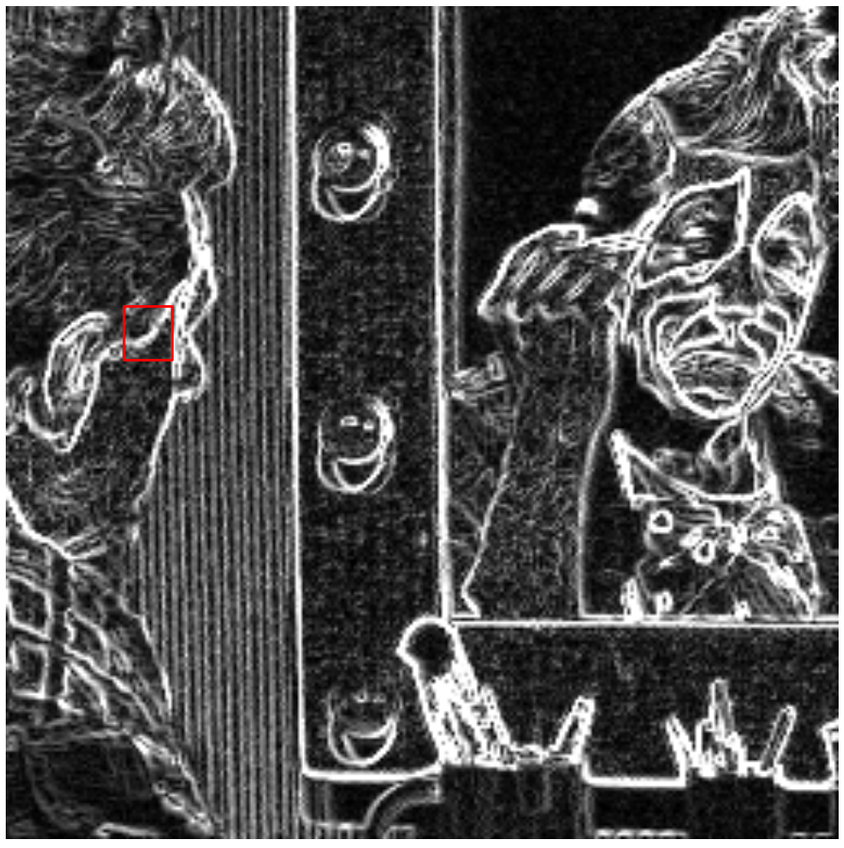
Normale au contour

$$\nabla f(x,y) = \begin{pmatrix} \partial_x f(x,y) \\ \partial_y f(x,y) \end{pmatrix}$$

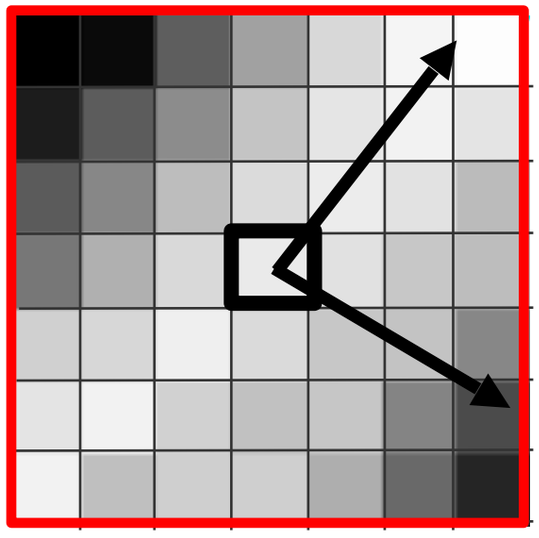
Suppression des non-max (*thinning*)

Rendre les contours filiformes

$$\left|\nabla f(x,y)\right|$$



Direction du contour
(perpendiculaire à la normale)



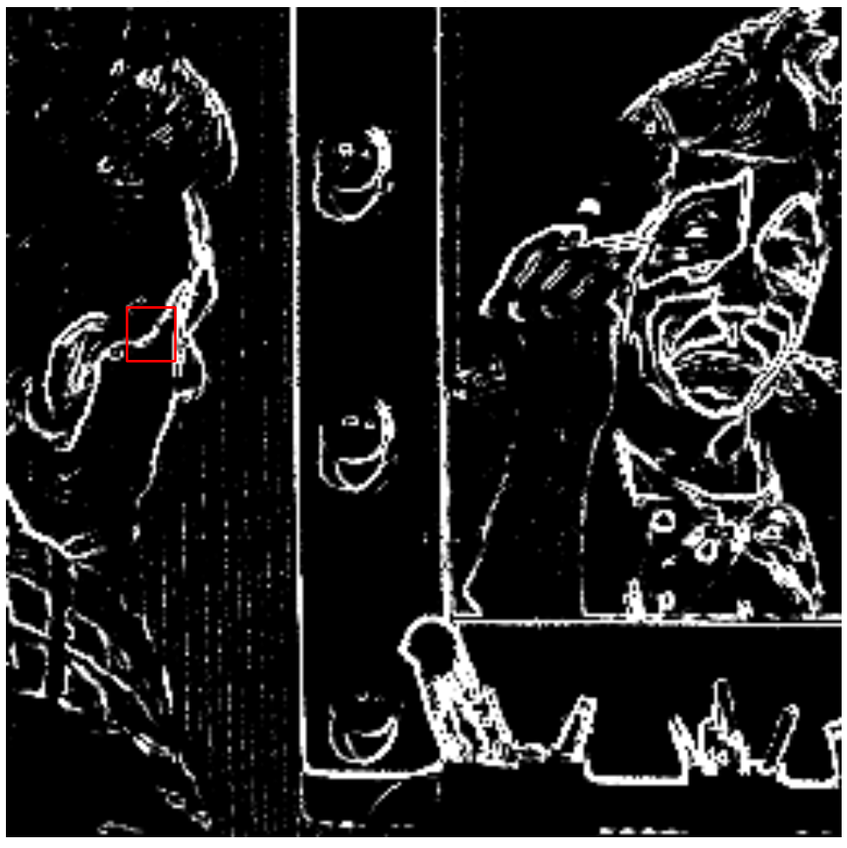
Normale au contour

$$\nabla f(x,y) = \begin{pmatrix} \partial_x f(x,y) \\ \partial_y f(x,y) \end{pmatrix}$$

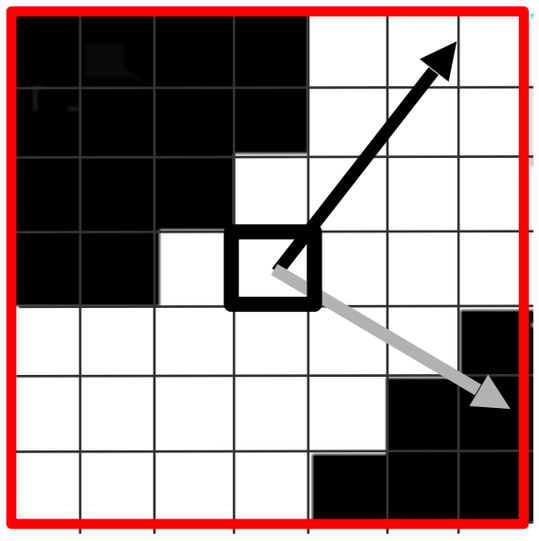
Suppression des non-max (*thinning*)

Rendre les contours filiformes

$$\left|\nabla f(x,y)\right| > Seuil$$



Direction du contour
(perpendiculaire à la normale)



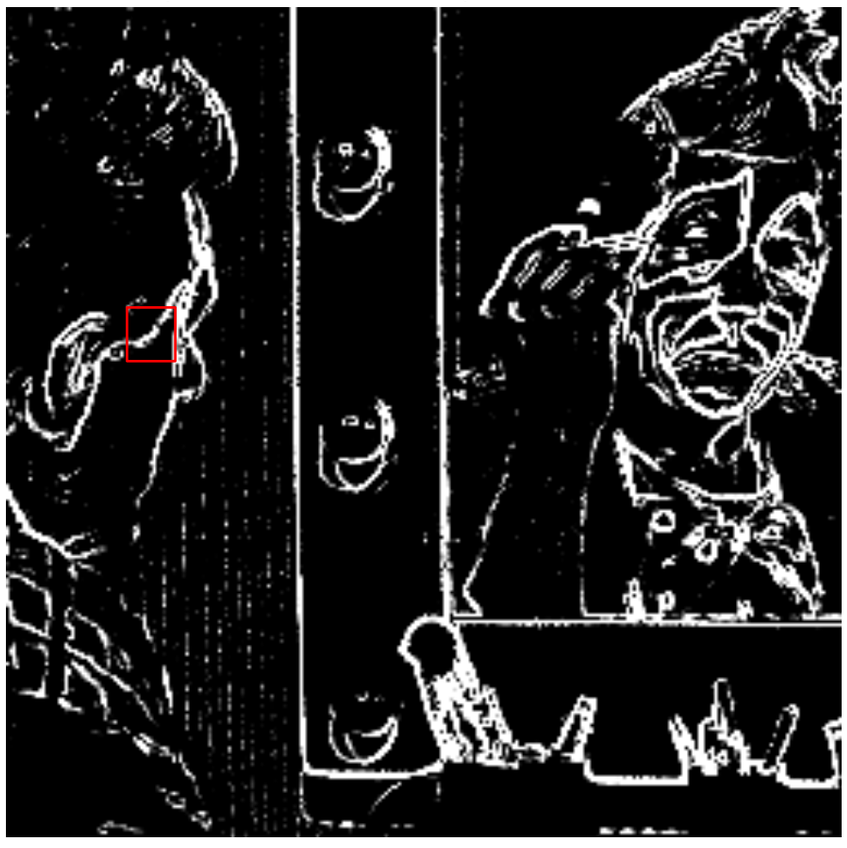
Normale au contour

Contour ayant une épaisseur > 1 pixel

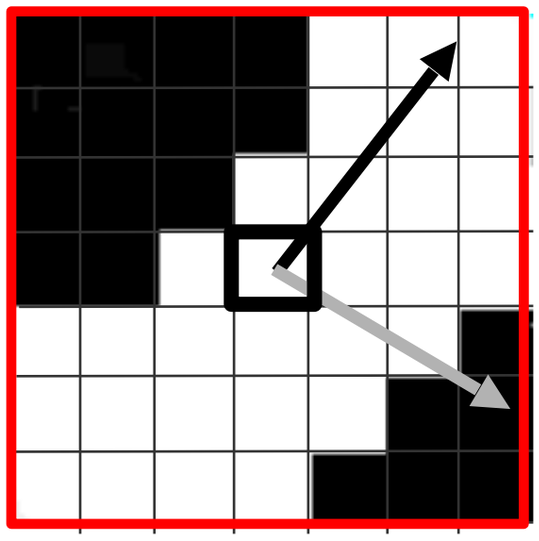
Suppression des non-max (*thinning*)

Rendre les contours filiformes

$$\left|\nabla f(x,y)\right| > Seuil$$



Direction du contour
(perpendiculaire à la normale)



Normale au contour

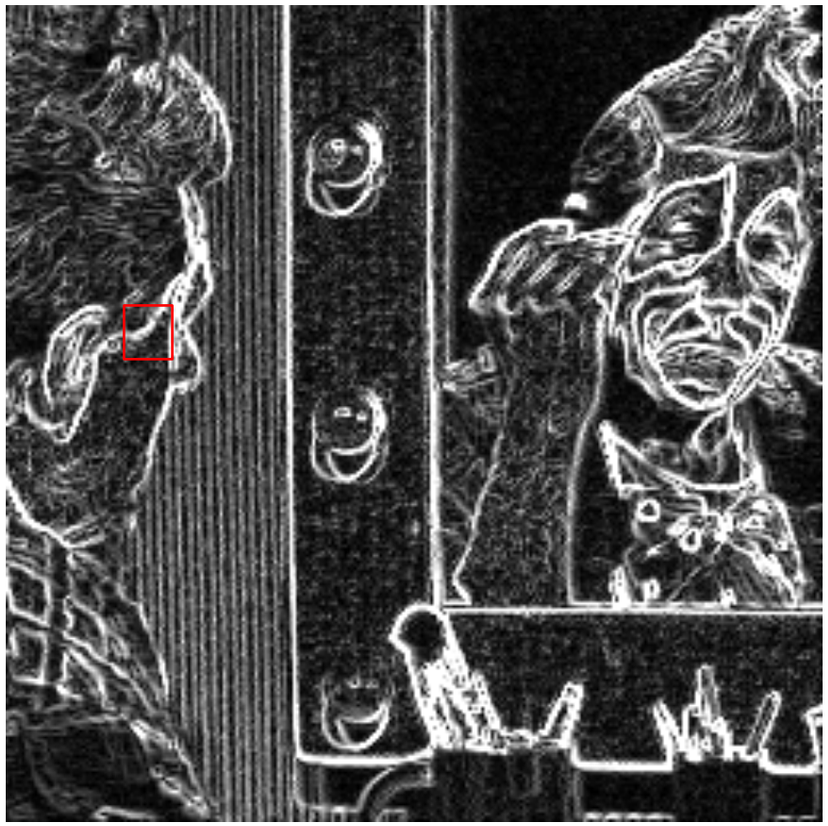
Contour ayant une épaisseur > 1 pixel

Pour éviter qu'un contour ait plus d'un pixel d'épais, on peut forcer à zéro tous les pixels dont $\left|\nabla f(x,y)\right|$ n'est **pas localement maximale** le long de la normale.

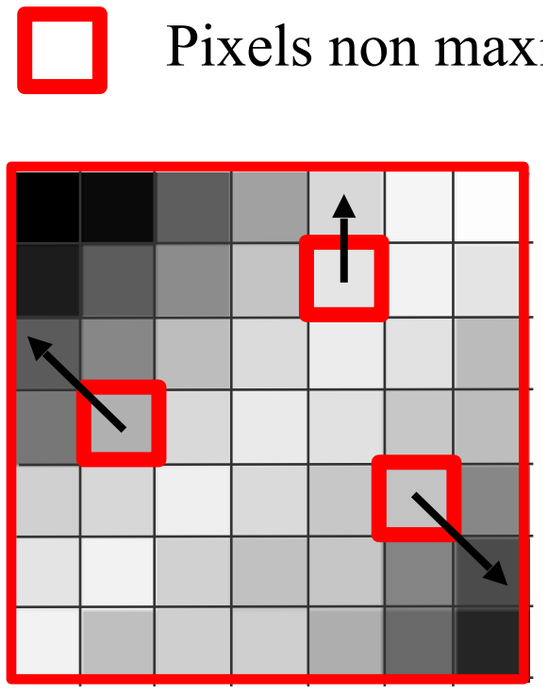
Suppression des non-max (*thinning*)

Rendre les contours filiformes

$$|\nabla f(x,y)|$$



Exemple de trois pixels **non-maximums**



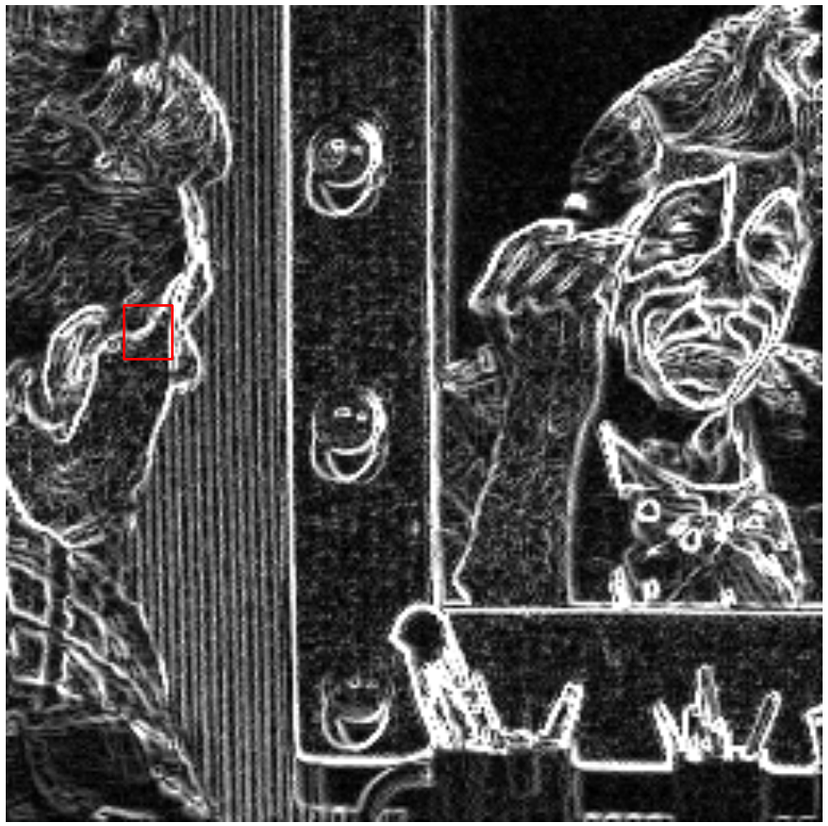
En général, on localise les deux voisins d'un pixel à l'aide **<u>l'angle</u>** de la normale

$$\theta(x,y) = \arctan\left(\frac{\partial f(x,y)}{\partial y} \Big/ \frac{\partial f(x,y)}{\partial x}\right)$$

Suppression des non-max (*thinning*)

Rendre les contours filiformes

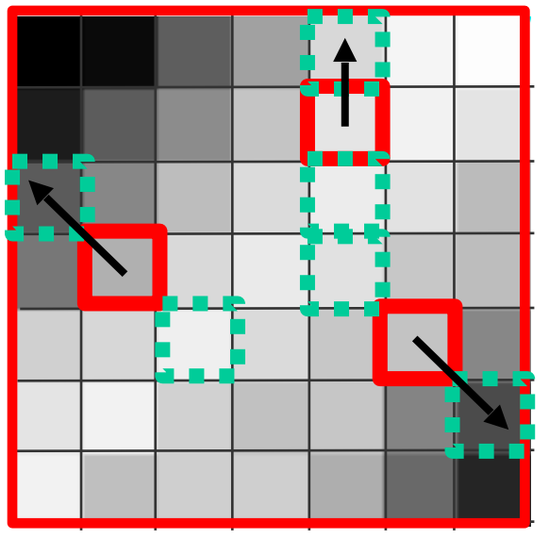
$$|\nabla f(x,y)|$$



Exemple de trois pixels **non-maximums**

Pixels voisins le long de la normale

Pixels non maximums



Normale

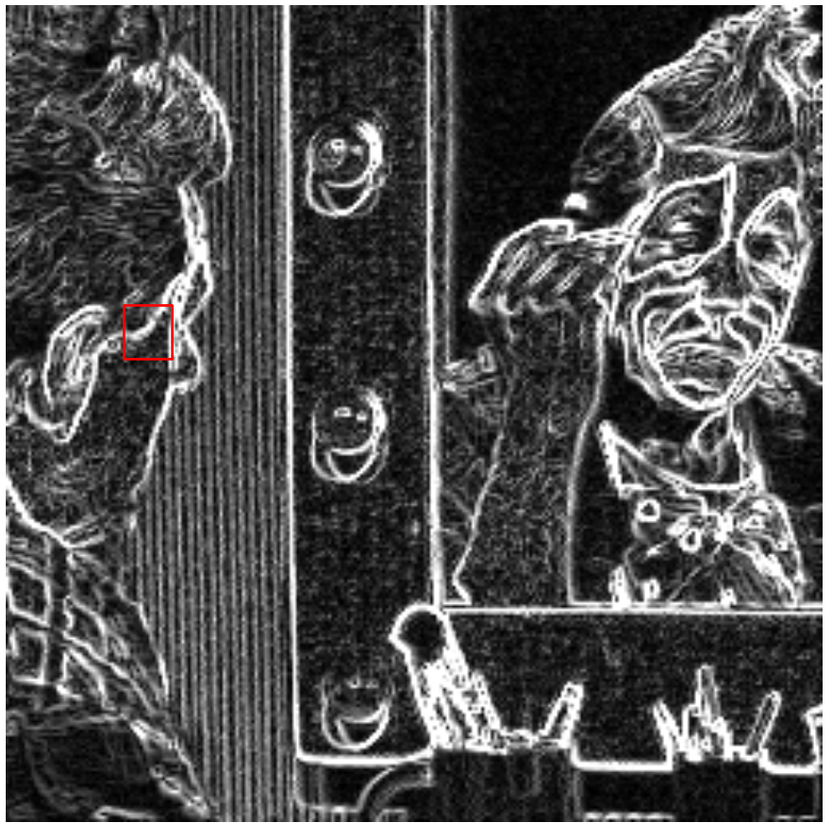
En général, on localise les deux voisins d'un pixel à l'aide **<u>l'angle</u>** de la normale

$$\theta(x,y) = \arctan\left(\frac{\partial f(x,y)}{\partial y} \Big/ \frac{\partial f(x,y)}{\partial x}\right)$$

Suppression des non-max (*thinning*)

Rendre les contours filiformes

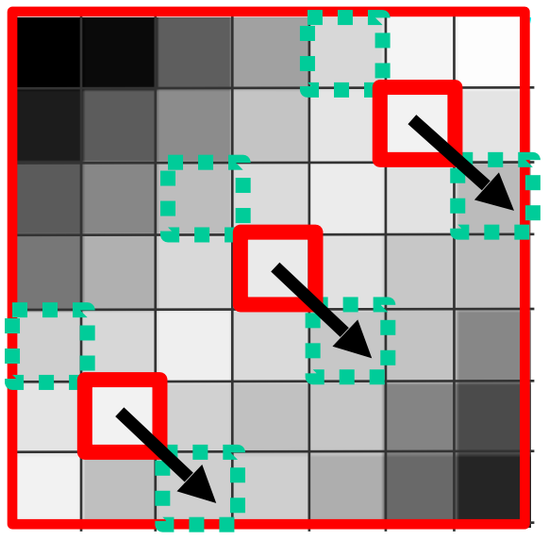
$$\left|\nabla f(x,y)\right|$$



Exemple de trois pixels **maximums**

Pixels voisins le long de la normale

Pixels correspondant à un maximum

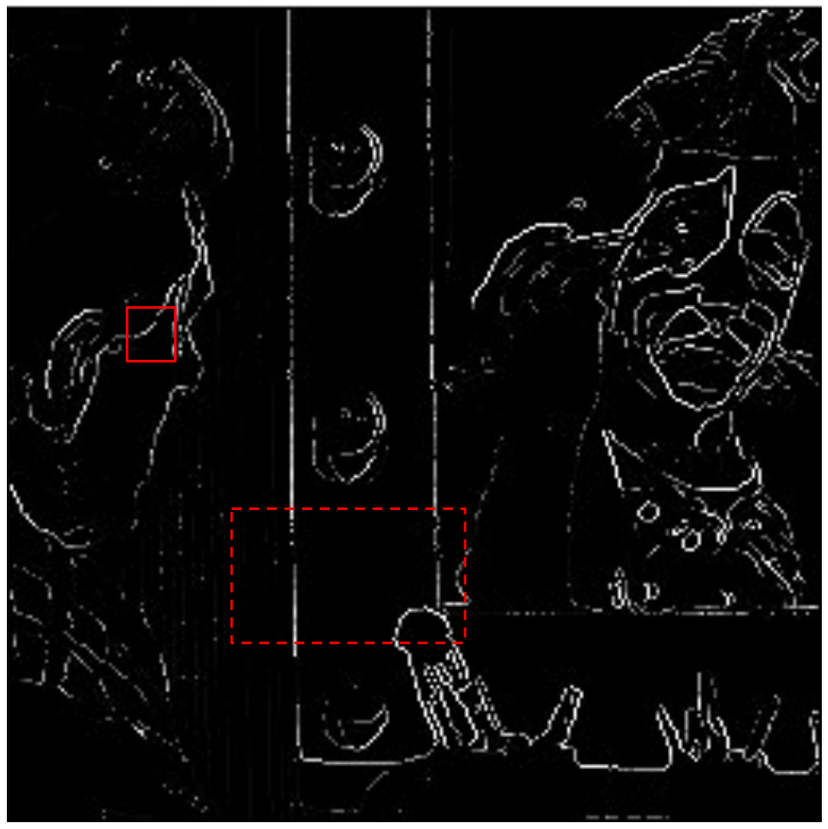


Normale

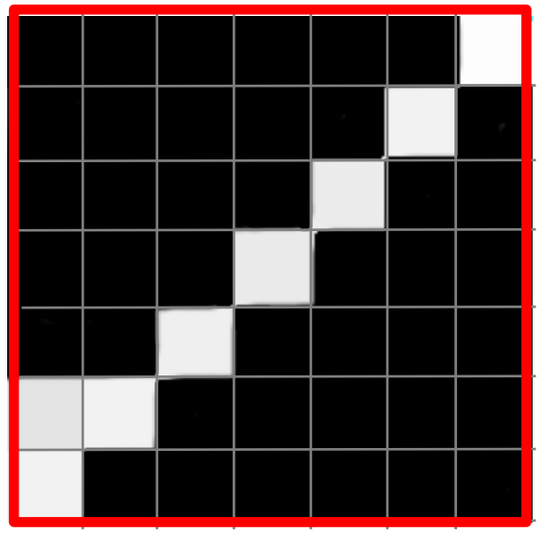
Suppression des non-max (*thinning*)

Rendre les contours filiformes

$|\nabla f(x,y)|$ avec Non - Max à zéro



Voici ce qui arrive lorsqu'on force à zéro les non-maximums

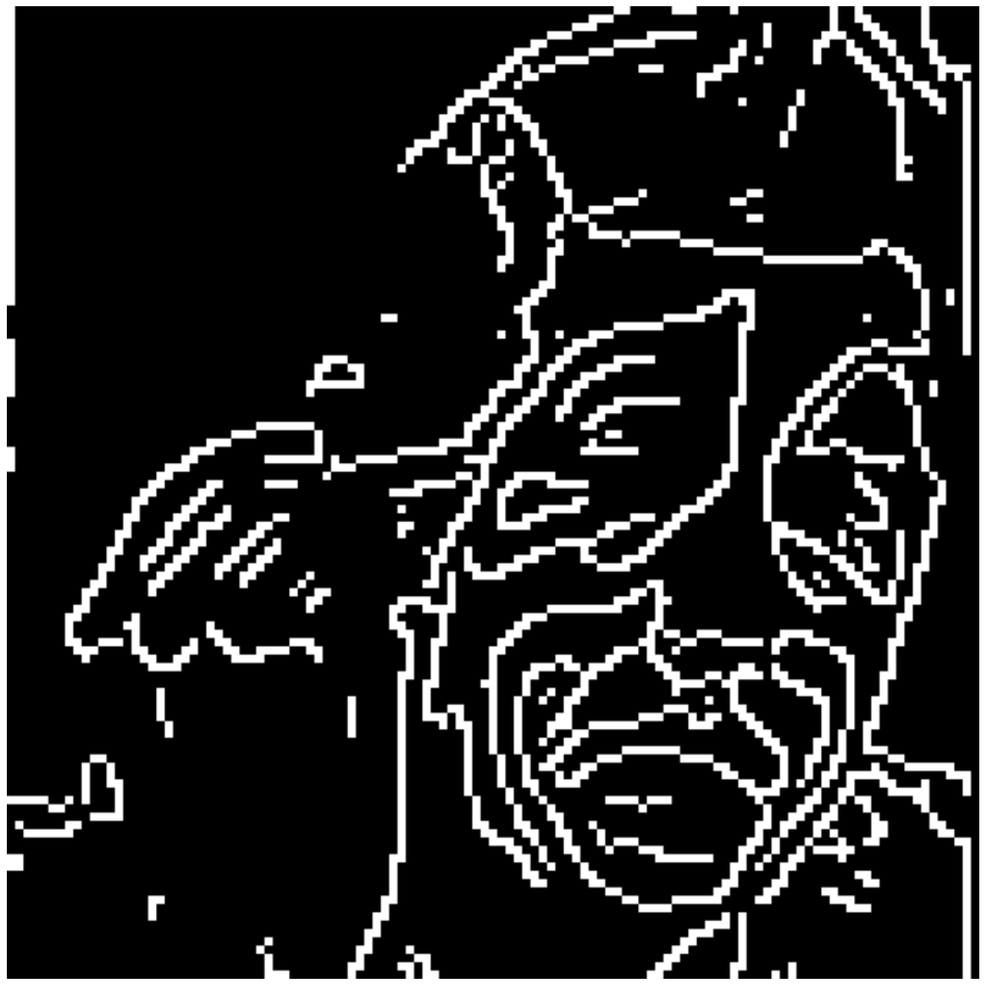


Suppression des non-max (*thinning*)

Rendre les contours filiformes

Voici ce qui arrive lorsqu'on seuil la norme du gradient dont les non-maximums ont été forcés à zéro.

Image binaire avec des contours d'épaisseur = 1 pixel



Algorithme de détection de contours par *thinning*

1: Convoluer l'image d'entrée à l'aide d'un filtre gaussien

$$g = G_\sigma * f$$

2: Calculer la magnitude du gradient en chaque point de l'image:

$$|\nabla g(x,y)| = \sqrt{\left(\frac{\partial g(x,y)}{\partial x}\right)^2 + \left(\frac{\partial g(x,y)}{\partial y}\right)^2}$$

3: Calculer l'orientation de la normale du gradient en chaque point de l'image

$$\theta(x,y) = \arctan\left(\frac{\partial g(x,y)}{\partial y} \Big/ \frac{\partial g(x,y)}{\partial x}\right)$$

4: À l'aide de $|\nabla g(x,y)|$ et $\theta(x,y)$ supprimer les non-maximums

$$h(x,y) = \text{NonMaxSupp}\left(|\nabla g(x,y)|, \theta(x,y)\right)$$

5: Appliquer un seuil

$$c(x,y) = \begin{cases} 1 & \text{si } h(x,y) > seuil \\ 0 & \text{sinon} \end{cases}$$

2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- Contours filiformes

Image originale : f

2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- Contours filiformes

Image originale : f

$|\nabla f|$: Sobel

2. CONTOURS

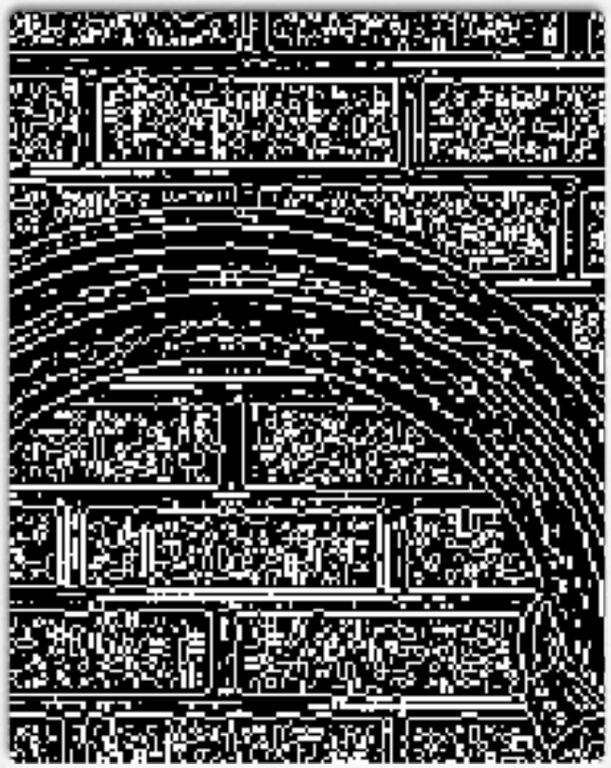
1. Détecteur gradient

b) Suppression des non-maxima

- Contours filiformes

Image originale : f

$|\nabla f|$: Sobel



Suppression des non-max

2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- Contours filiformes

Image originale : f

©Marie-Flavie Auclair-Fortier

2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- Contours filiformes

Image originale : f

Seuillage simple de $|\nabla f|$

calculé par le gradient de la Gaussienne ($\sigma = 1$, seuil = 50)

©Marie-Flavie Auclair-Fortier

2. CONTOURS

1. Détecteur gradient

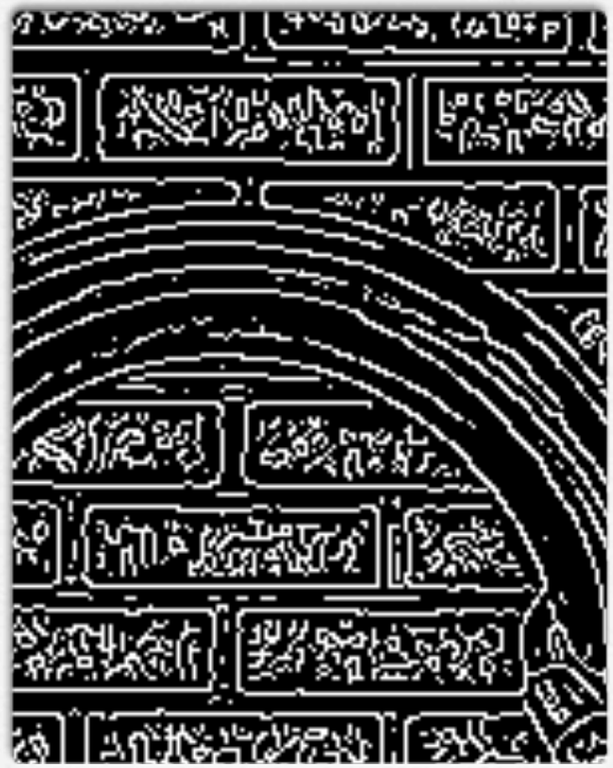
b) Suppression des non-maxima

- Contours filiformes

Image originale : f

Seuillage simple de $|\nabla f|$

calculé par le gradient de la Gaussienne ($\sigma = 1$, seuil = 50)



Suppression des non-max de l'amplitude du gradient de la Gaussienne

©Marie-Flavie Auclair-Fortier

2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- Contours filiformes
  - ✓ Gaussienne, effet de l'échelle σ

Image originale : *f*

2. CONTOURS

1. Détecteur gradient

b) Suppression des non-maxima

- Contours filiformes
  - ✓ Gaussienne, effet de l'échelle σ

Image originale : *f*

Suppression des non-max

$\sigma = 1$

2. CONTOURS

1. Détecteur gradient

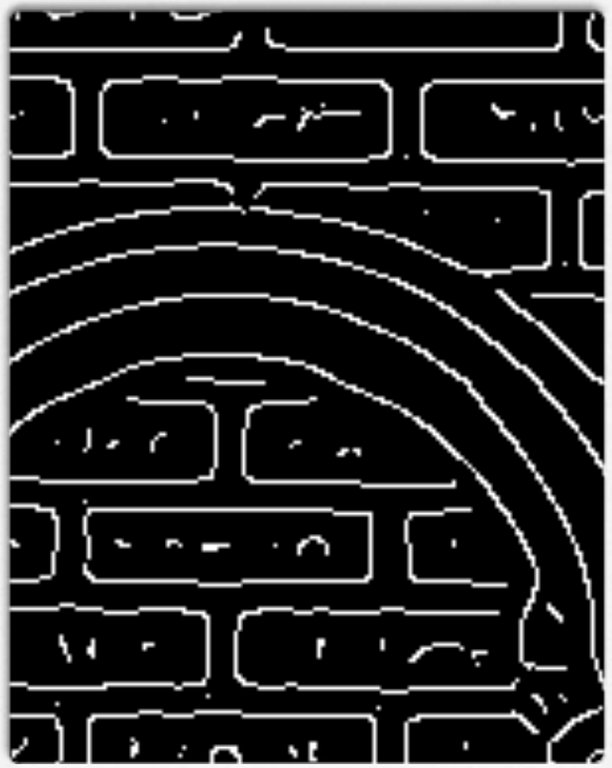
b) Suppression des non-maxima

- Contours filiformes
  - ✓ Gaussienne, effet de l'échelle σ

Image originale : *f*

Suppression des non-max

$\sigma = 1$



Suppression des non-max

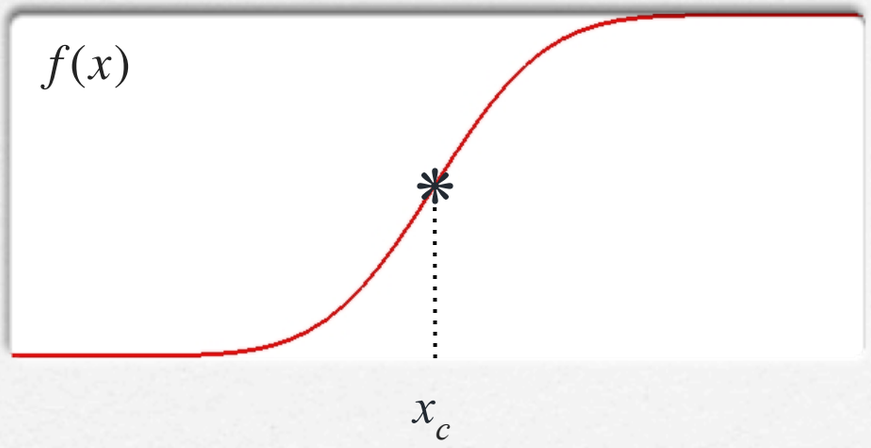
$\sigma = 3$

2. CONTOURS

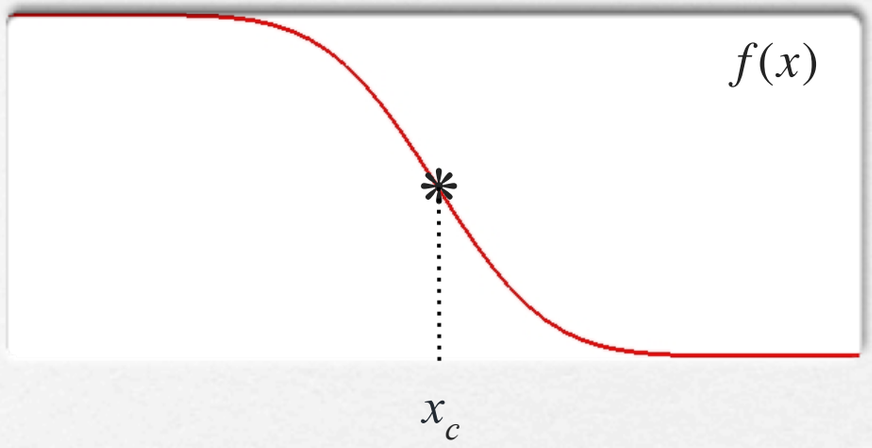
2. Détecteur Laplacien

- *Rappel*
  - ✓ En 1D, on peut localiser les contours par les passages par 0 de la 2^{e} dérivée

Marche montante



Marche descendante



Source :

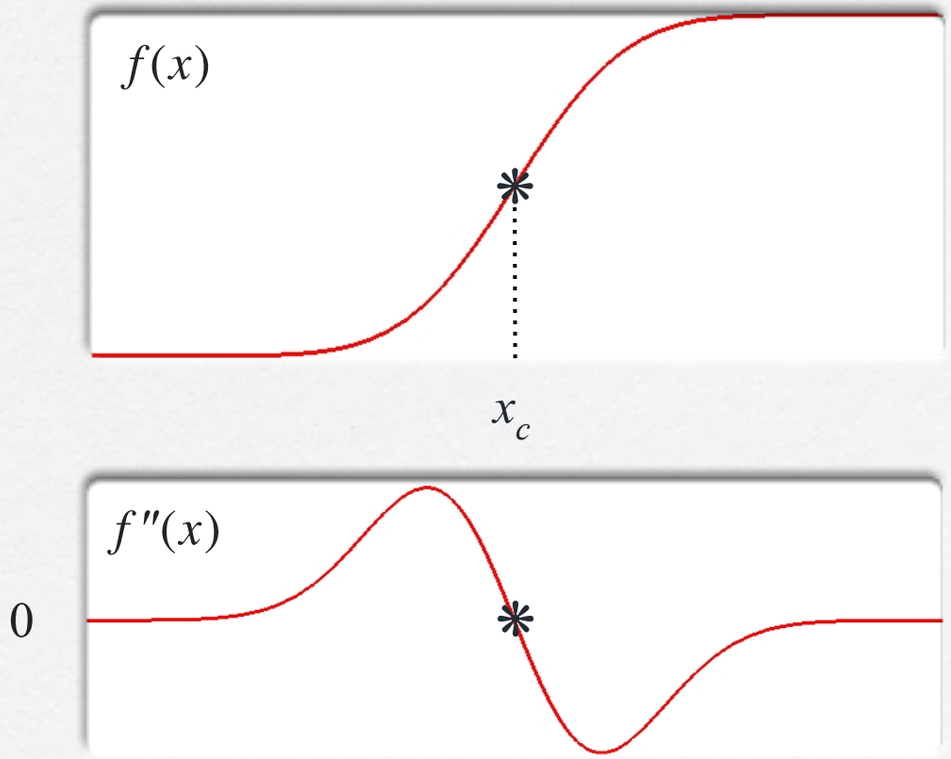
©Marie-Flavie Auclair-Fortier

2. CONTOURS

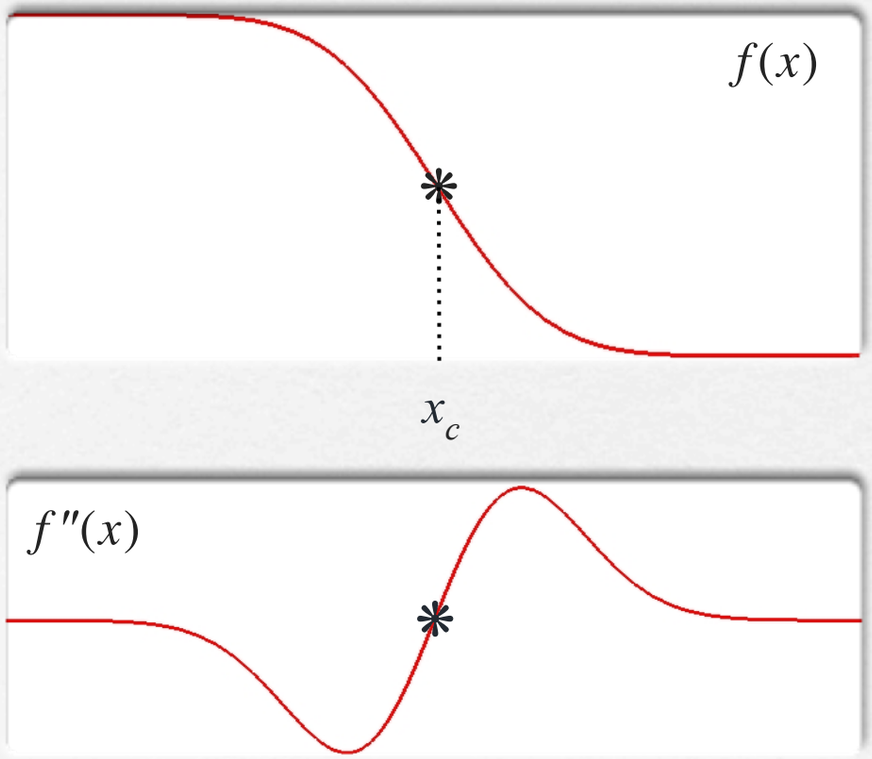
2. Détecteur Laplacien

- *Rappel*
  - ✓ En 1D, on peut localiser les contours par les passages par 0 de la 2^{e} dérivée

Marche montante



Marche descendante



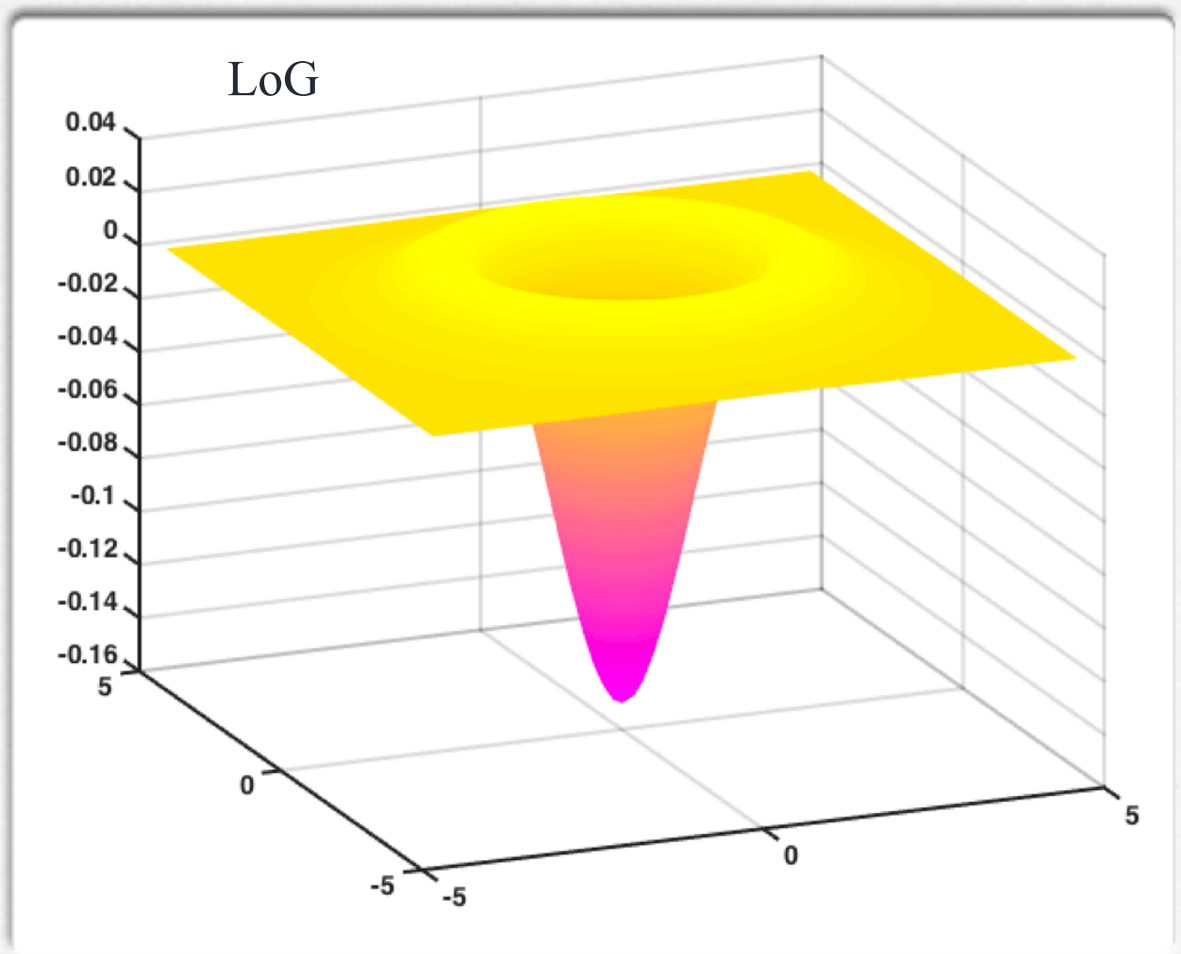
Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

- La forme particulière du LoG d'échelle σ permet de faire ressortir les éléments isolé d'une certaine grosseur (dictée par σ)
  - ✓ Une tache claire (réponse foncée) de rayon $\cong \sqrt{2\sigma}$ sera détectée comme un min (réponse max) local de $\nabla^2_\sigma f$



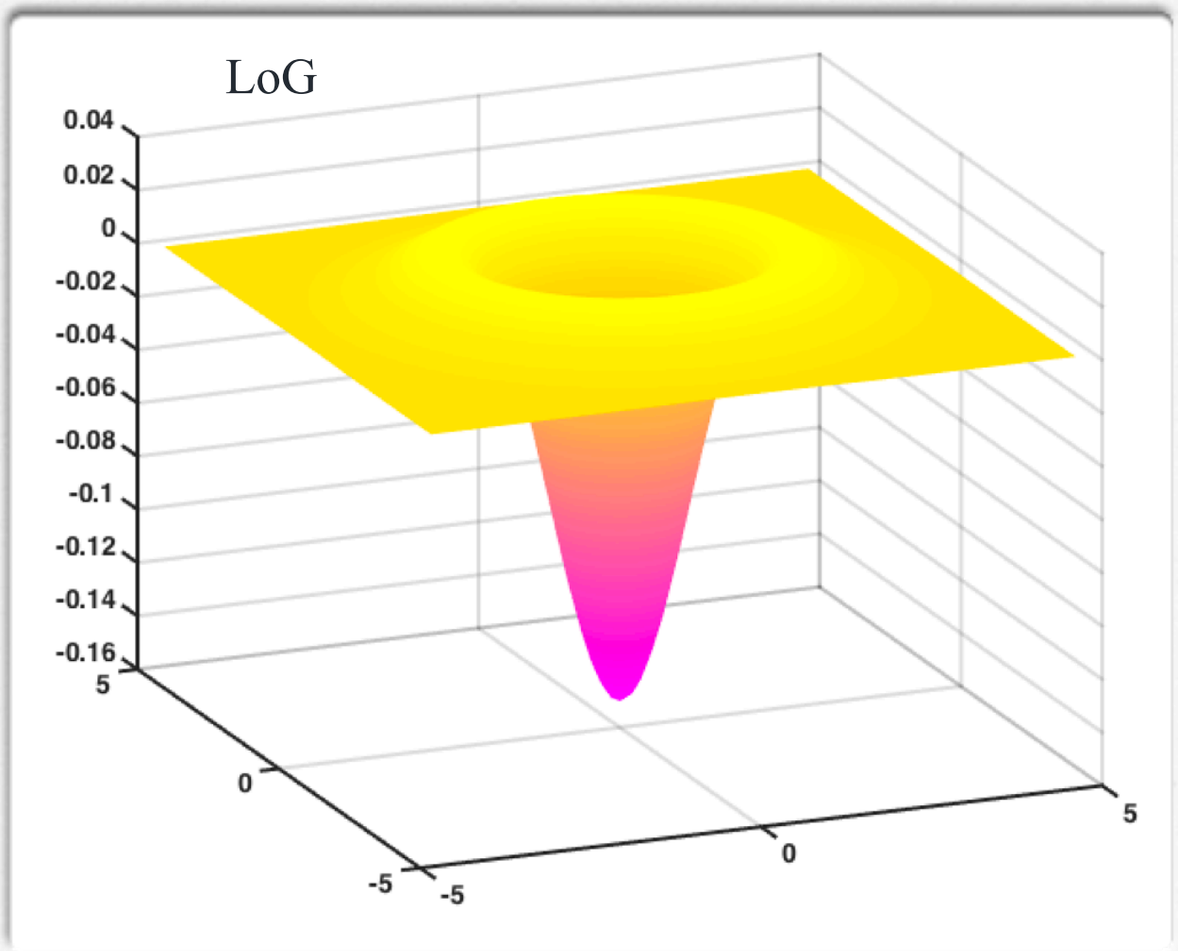
Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

- La forme particulière du LoG d'échelle σ permet de faire ressortir les éléments isolé d'une certaine grosseur (dictée par σ)
  - ✓ Une tache claire (réponse foncée) de rayon $\cong \sqrt{2\sigma}$ sera détectée comme un min (réponse max) local de $\nabla^2_\sigma f$



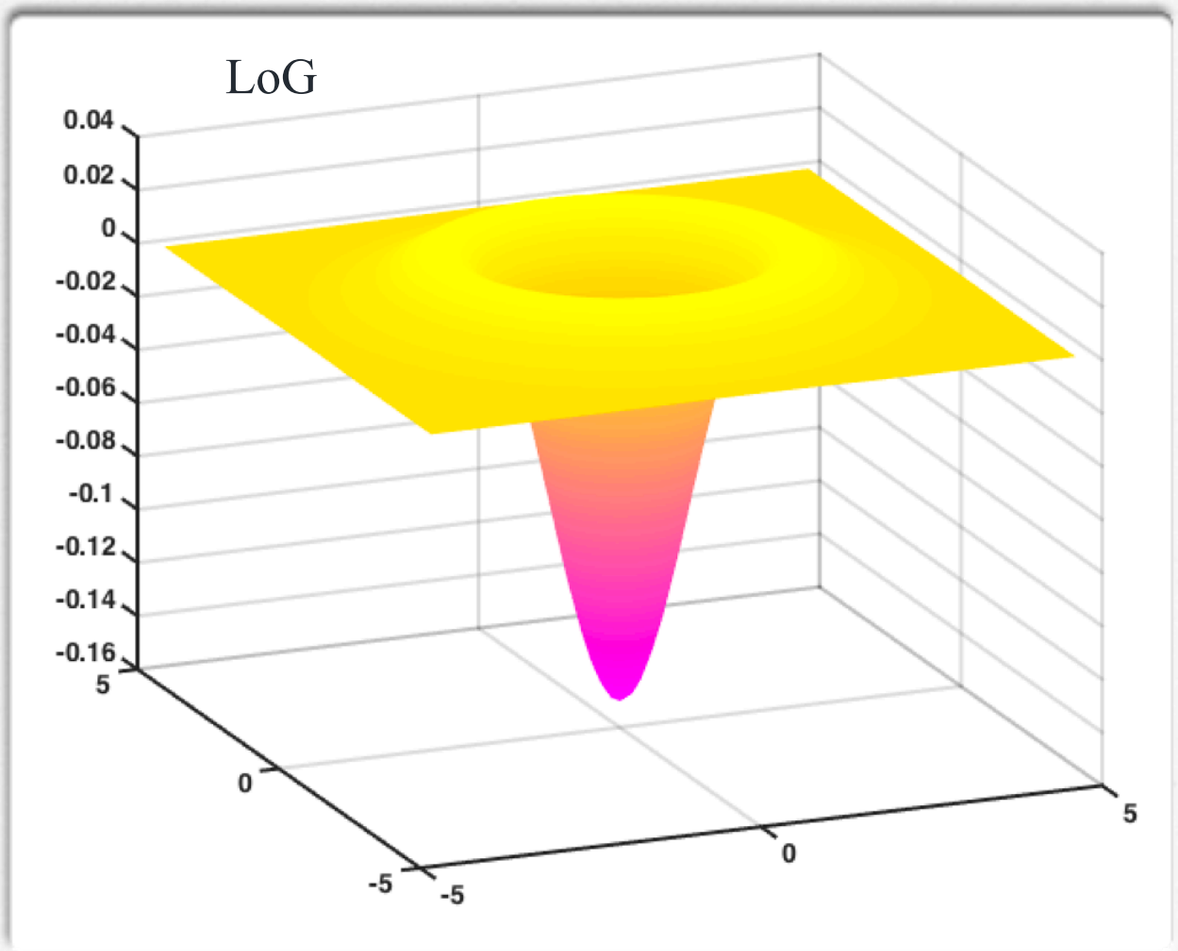
Source :

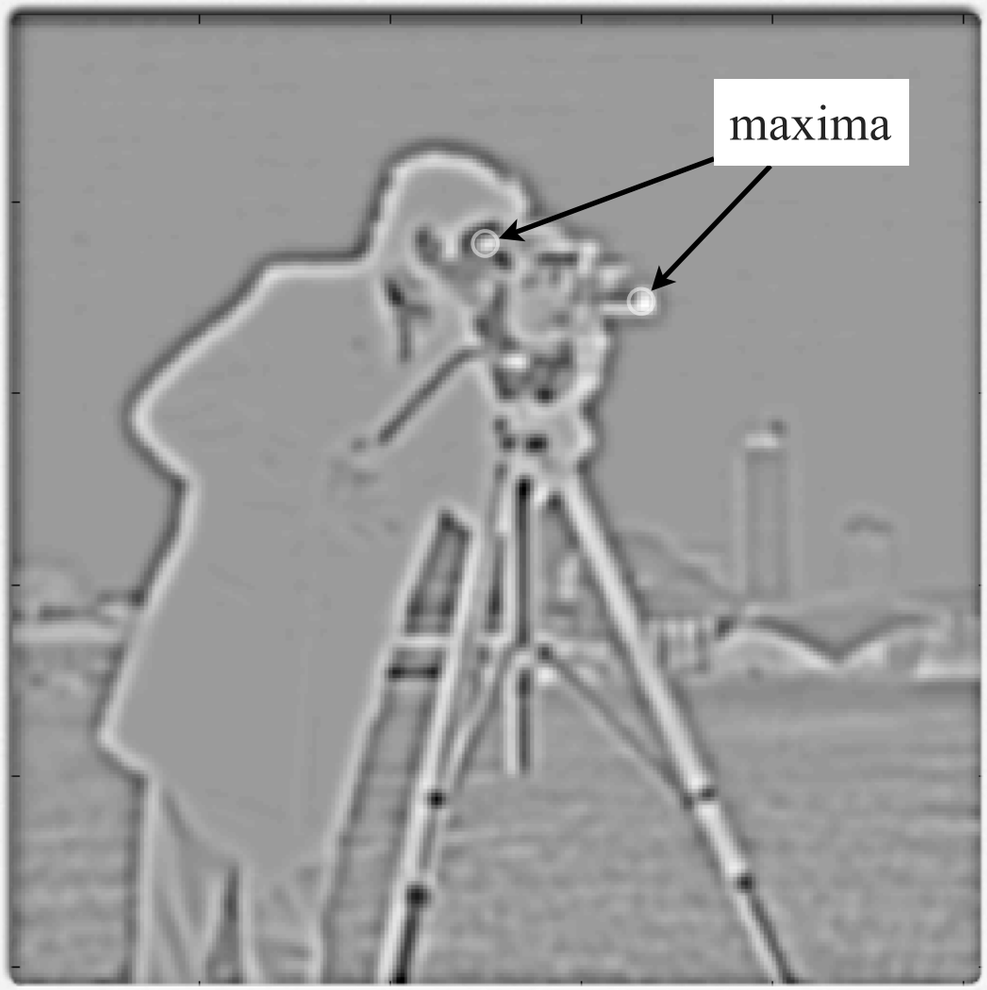
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

- La forme particulière du LoG d'échelle σ permet de faire ressortir les éléments isolé d'une certaine grosseur (dictée par σ)
  - ✓ Une tache claire (réponse foncée) de rayon $\cong \sqrt{2\sigma}$ sera détectée comme un min (réponse max) local de $\nabla^2_\sigma f$





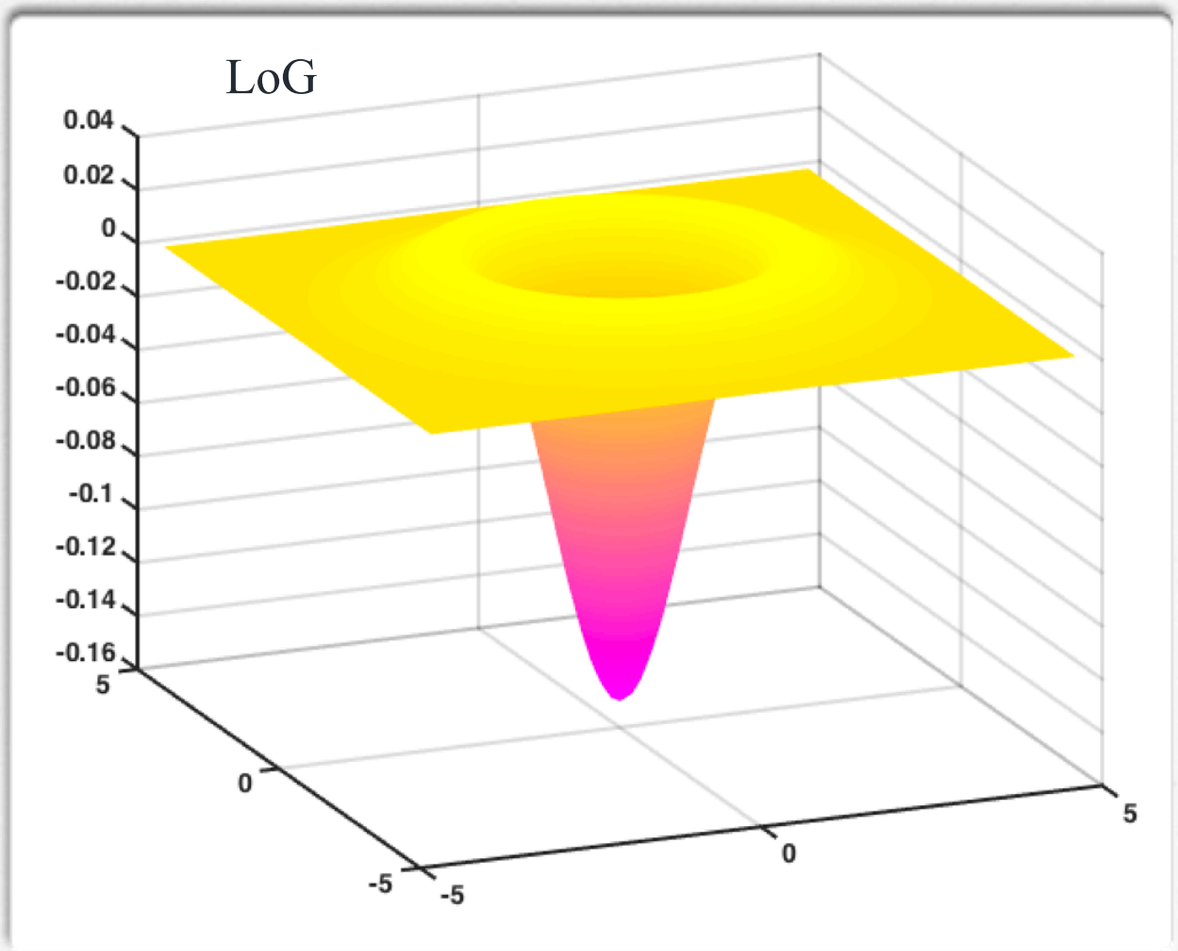
Source :

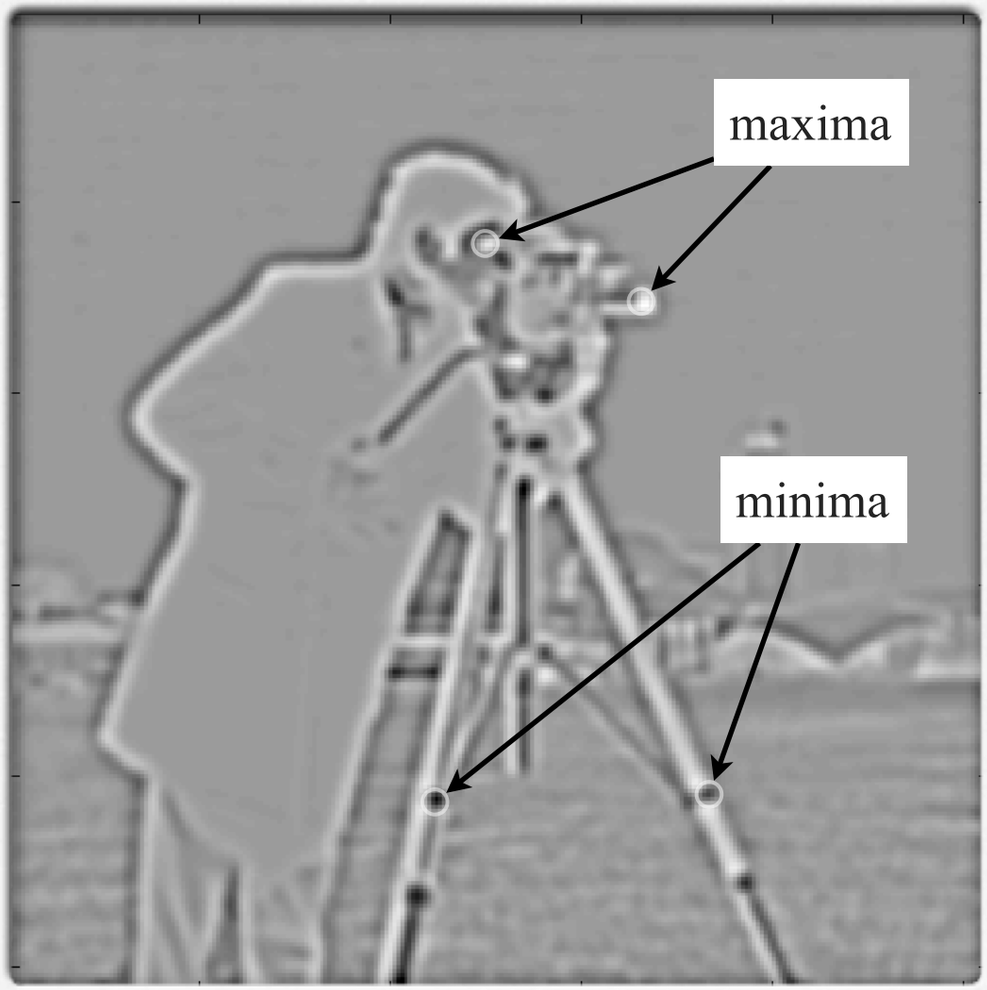
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

- La forme particulière du LoG d'échelle σ permet de faire ressortir les éléments isolé d'une certaine grosseur (dictée par σ)
  - ✓ Une tache claire (réponse foncée) de rayon $\cong \sqrt{2\sigma}$ sera détectée comme un min (réponse max) local de $\nabla^2_\sigma f$





Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

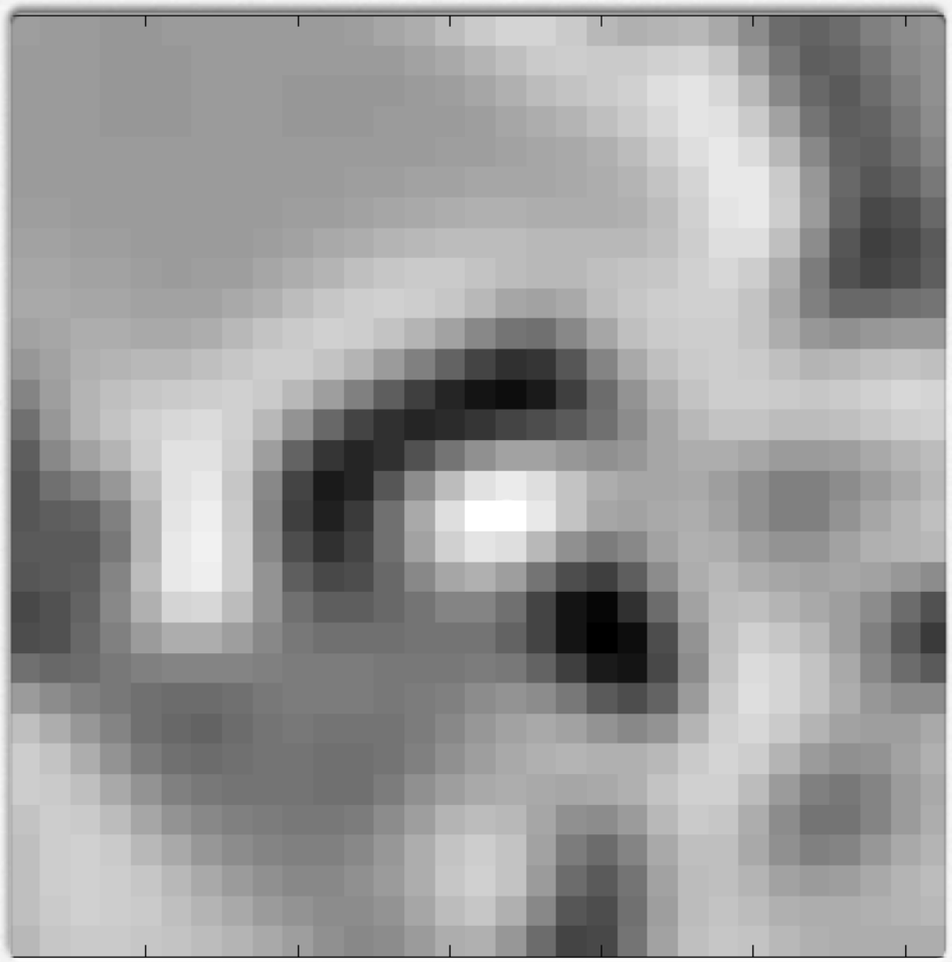
2. Laplacien

Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien



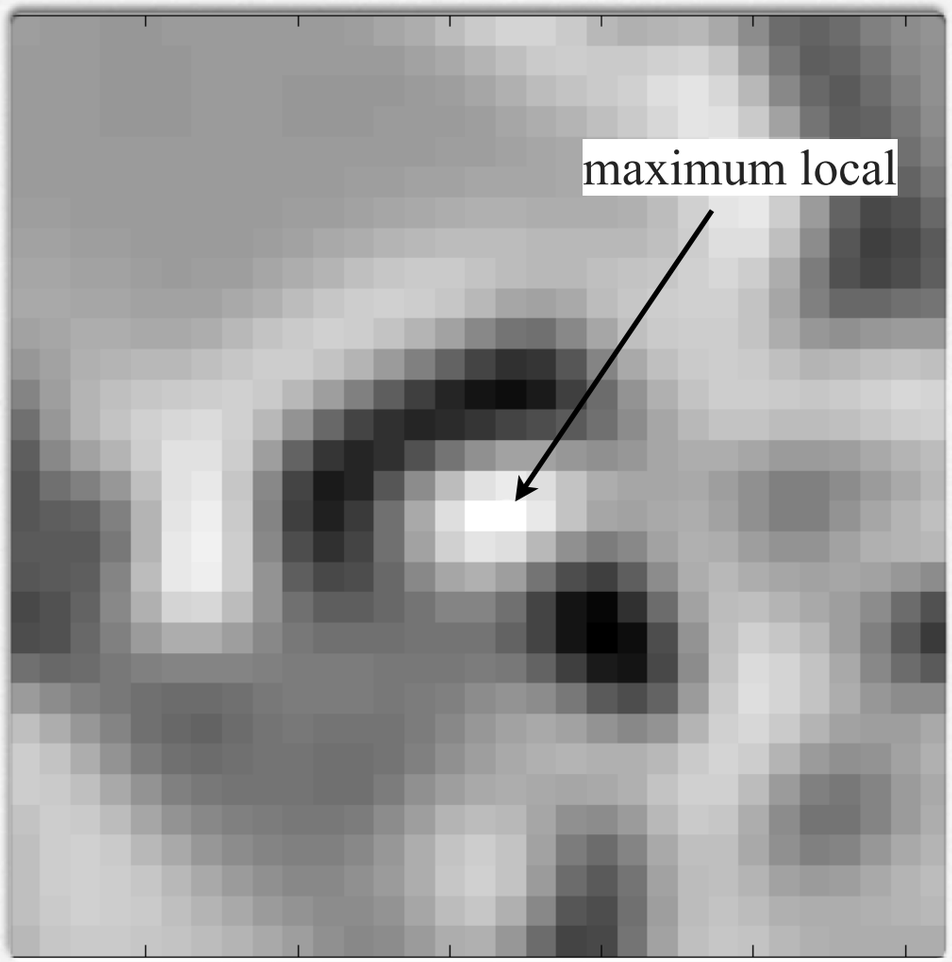
LoG $\sigma = 2$

Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien



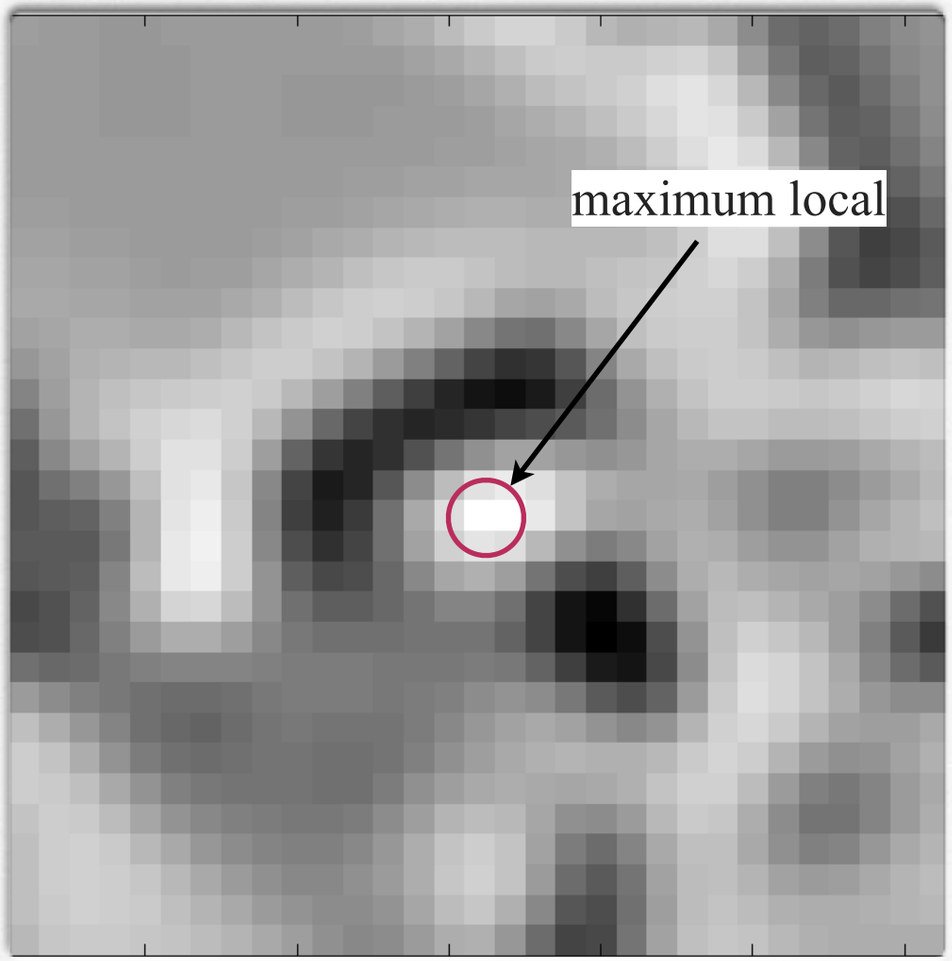
LoG $\sigma = 2$

Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien



LoG $\sigma = 2$

Source :

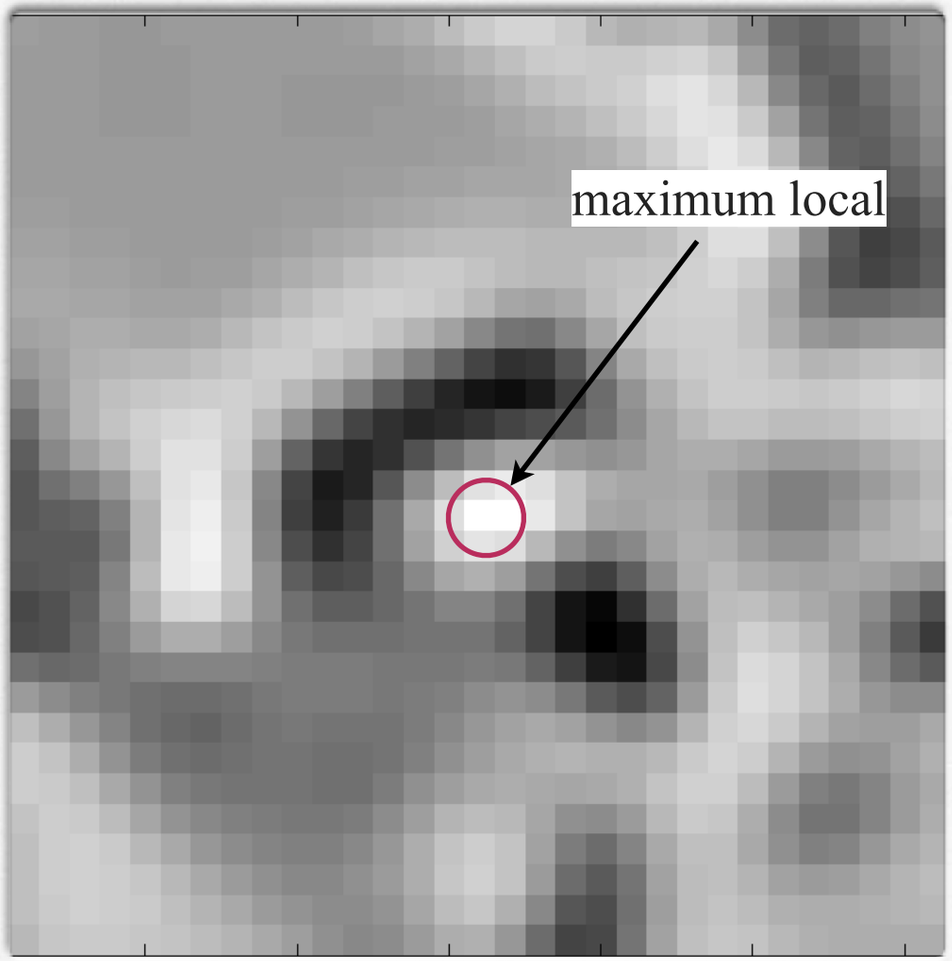
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

Remarque :

- en variant l'échelle, on peut aller chercher des zones de différentes grosseurs



LoG $\sigma = 2$

Source :

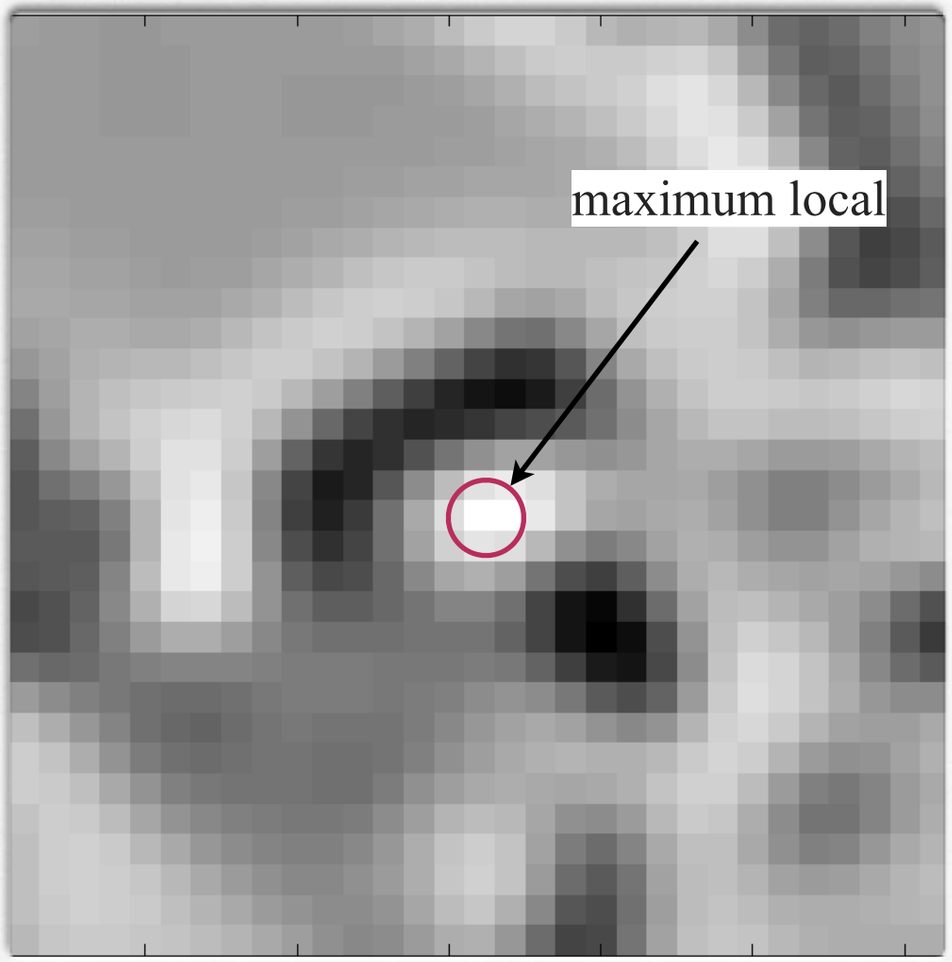
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

Remarque :

- en variant l'échelle, on peut aller chercher des zones de différentes grosseurs



LoG $\sigma = 2$

LoG $\sigma = 10$

Source :

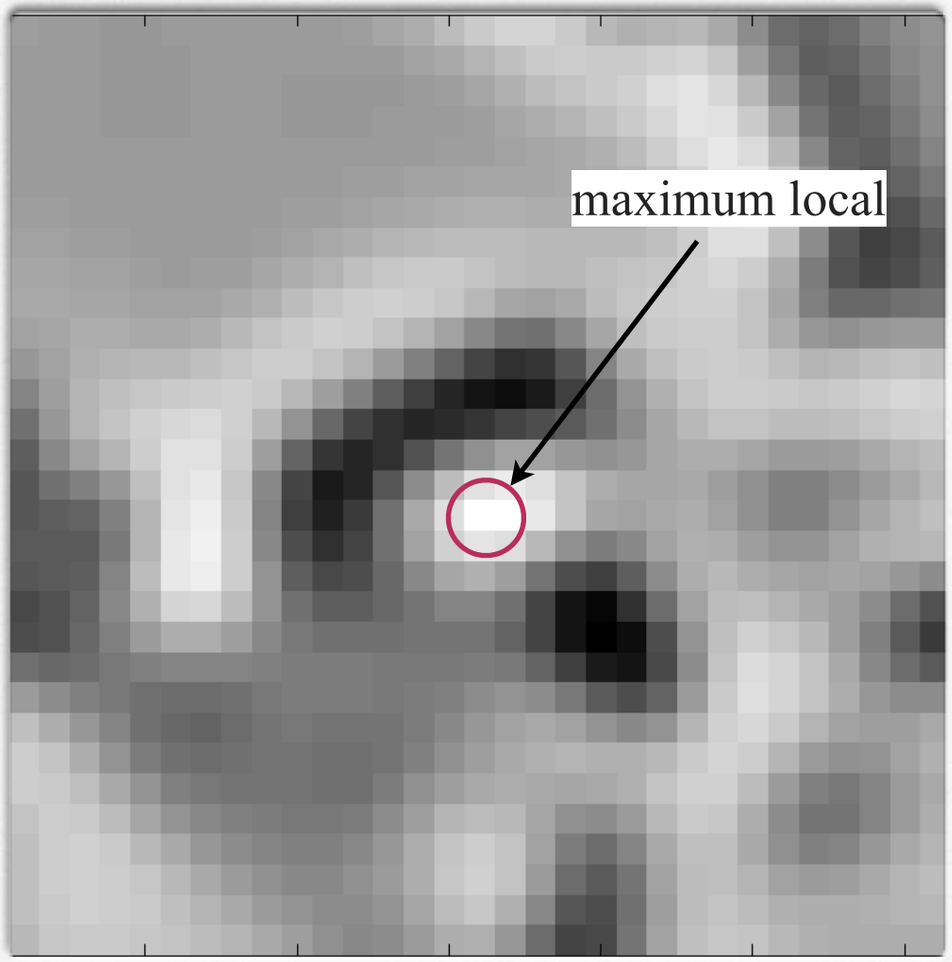
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

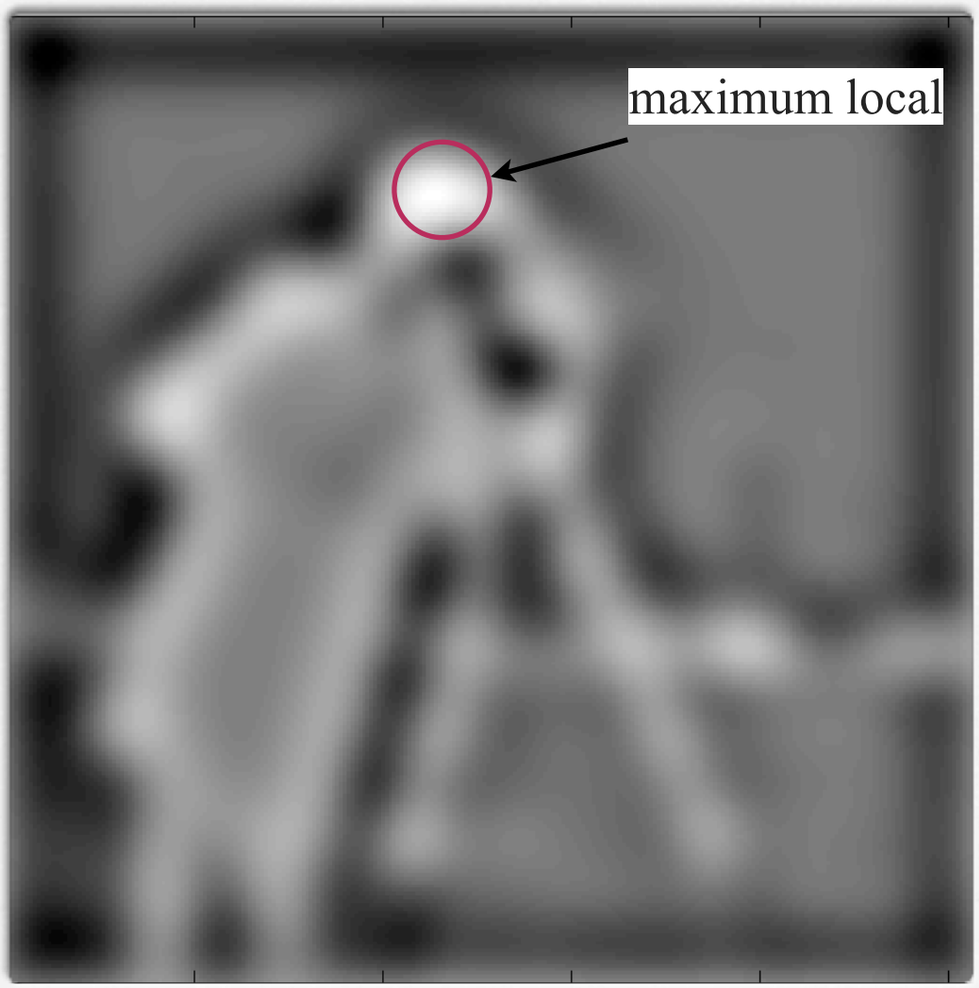
2. Laplacien

Remarque :

- en variant l'échelle, on peut aller chercher des zones de différentes grosseurs



LoG $\sigma = 2$



LoG $\sigma = 10$

Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

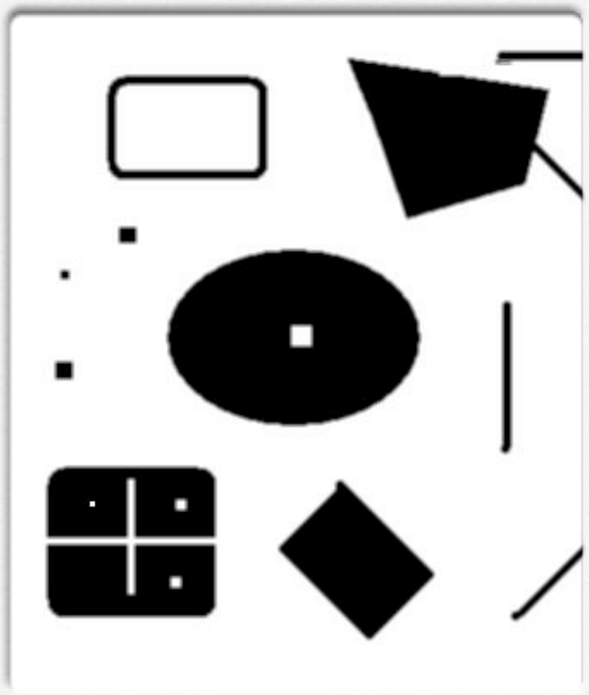


Image originale

Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

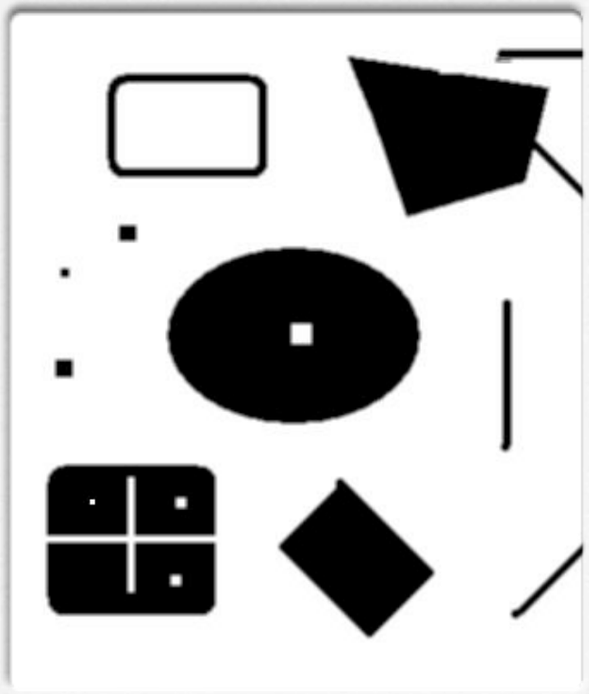


Image originale

$\nabla^2_\sigma f[m,n]$ avec $\sigma = 1.5$

3. Caractéristiques ponctuelles

2. Laplacien

$\nabla^2_\sigma f[m,n]$ avec $\sigma = 3$

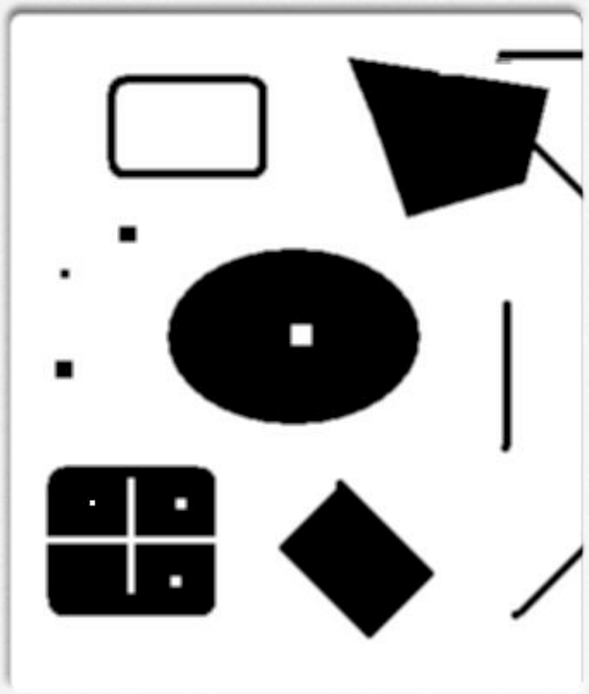


Image originale

$\nabla^2_\sigma f[m,n]$ avec $\sigma = 1.5$

Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

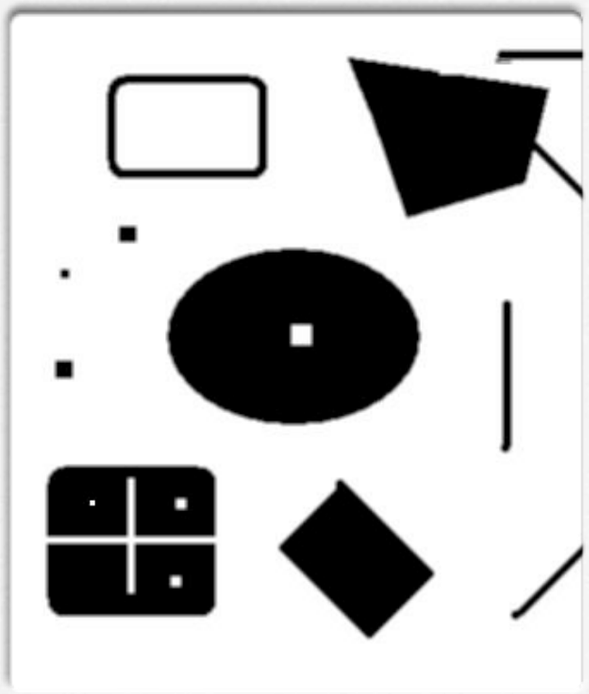


Image originale

$\nabla^2_\sigma f[m,n]$ avec $\sigma = 1.5$

$\nabla^2_\sigma f[m,n]$ avec $\sigma = 3$

$\nabla^2_\sigma f[m,n]$ avec $\sigma = 4.5$

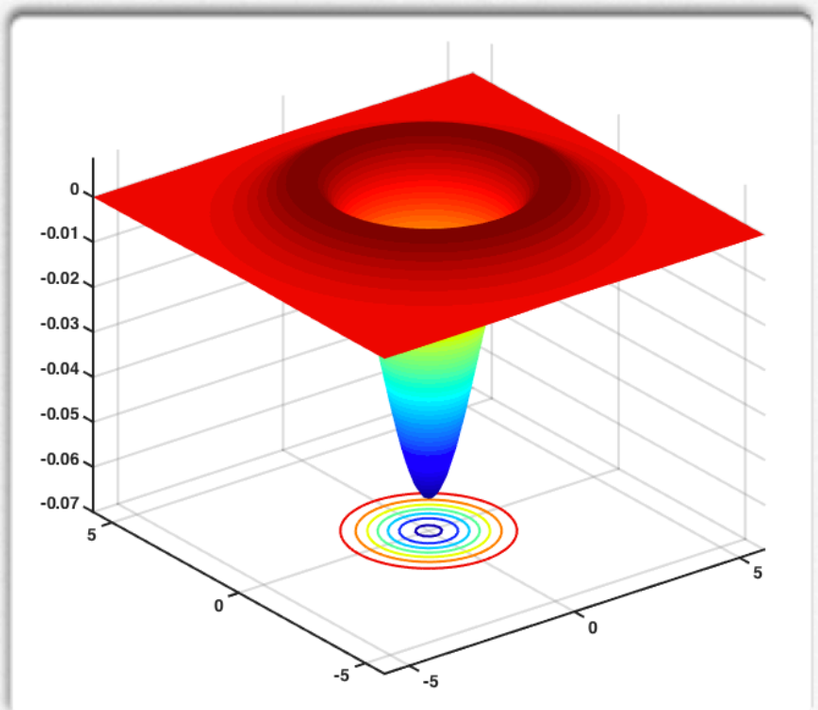
Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

2. Laplacien

$\nabla^2 g_\sigma[m,n]$ avec $\sigma = 1.5$



min = -0.0629

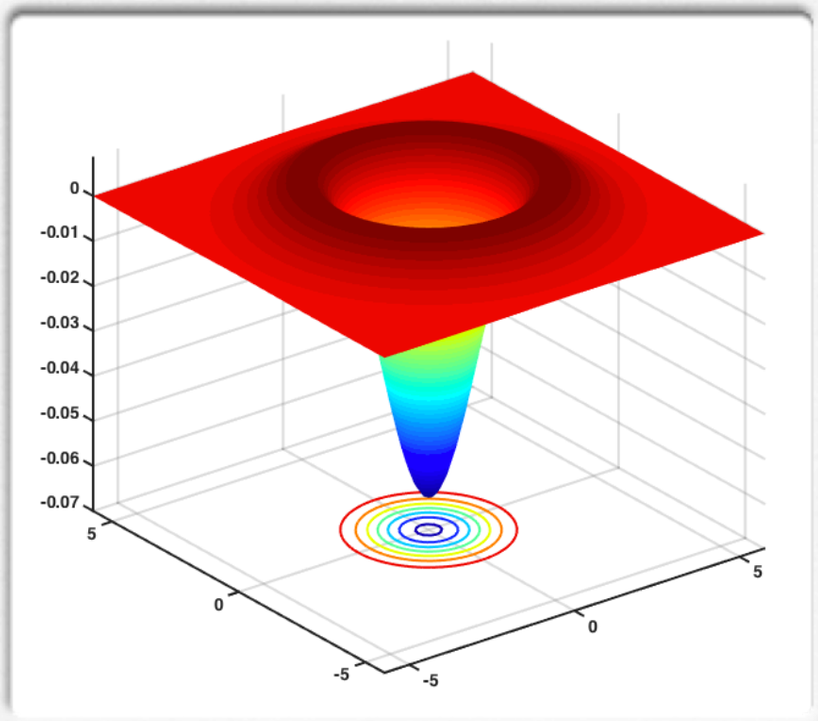
Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

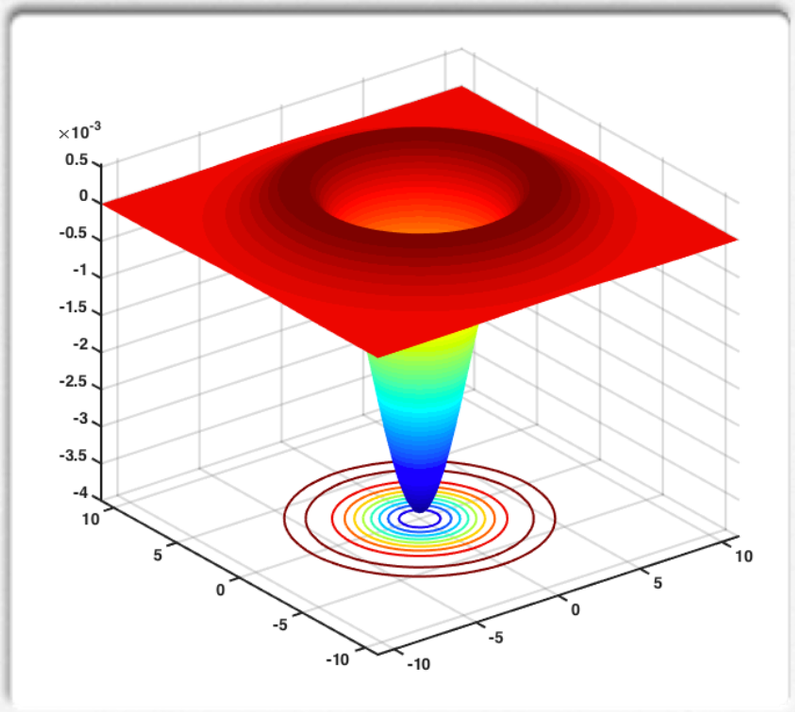
2. Laplacien

$\nabla^2 g_\sigma[m,n]$ avec $\sigma = 1.5$



min = -0.0629

$\nabla^2 g_\sigma [m, n]$ avec $\sigma = 3$



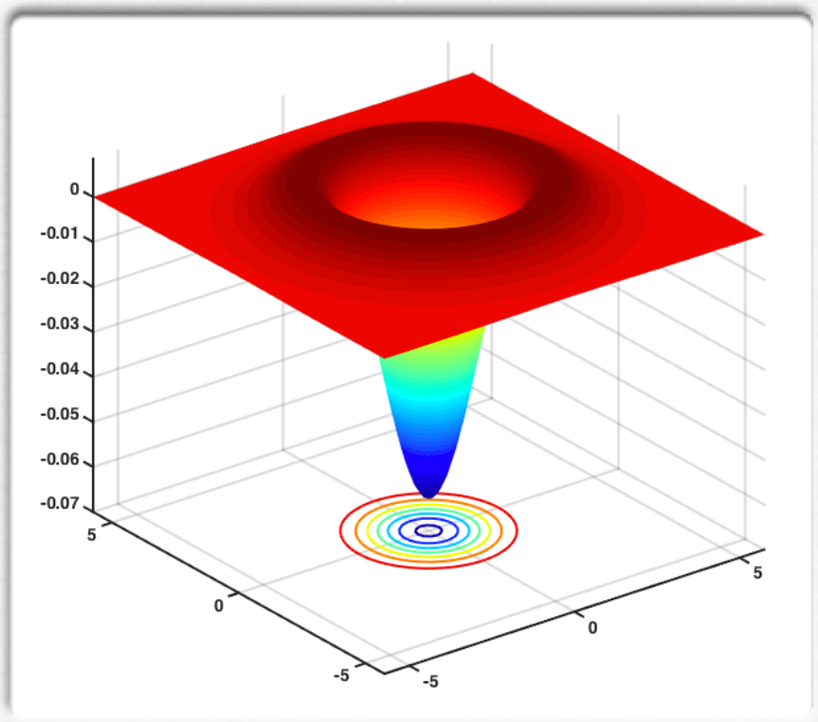
min =-0.0039

Source :

3. Caractéristiques ponctuelles

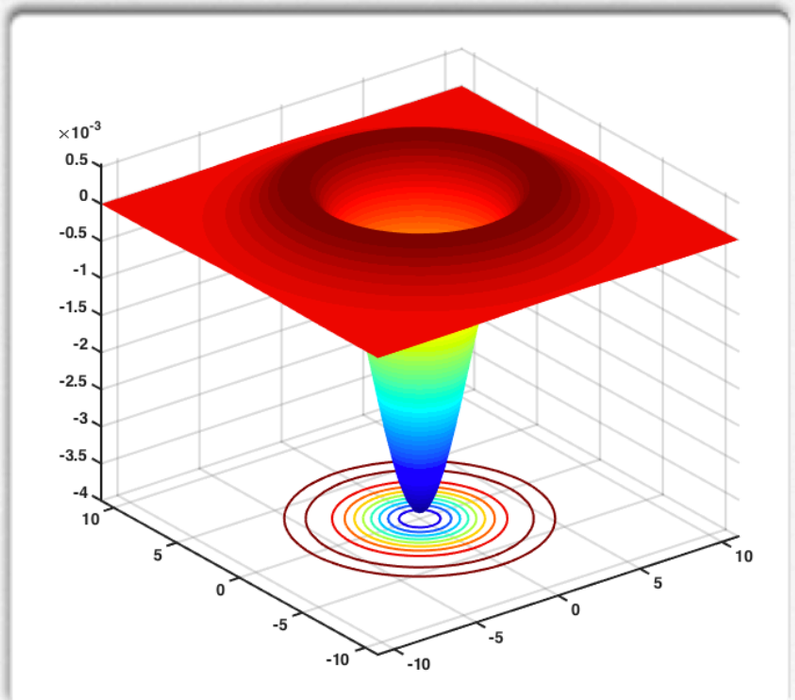
2. Laplacien

$\nabla^2 g_\sigma[m,n]$ avec $\sigma = 1.5$



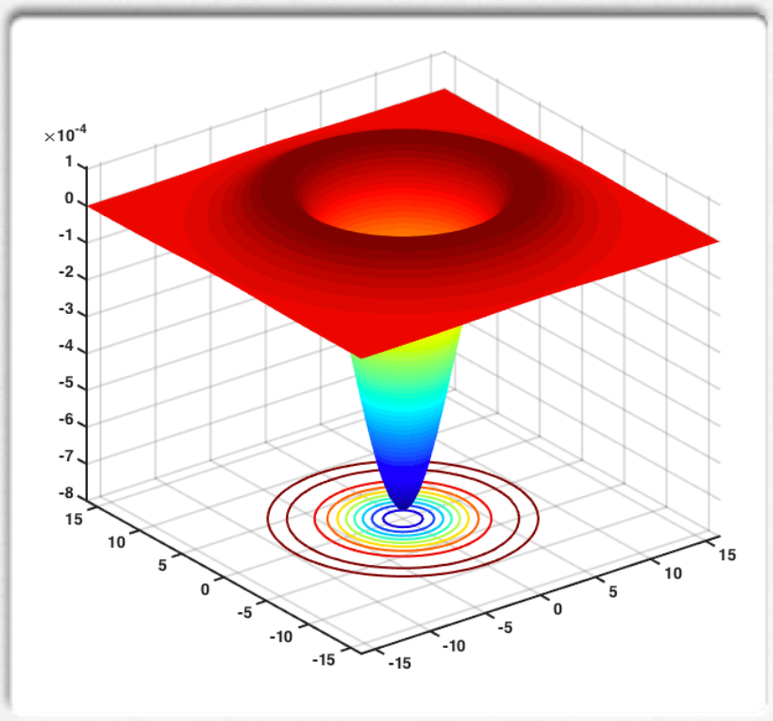
min = -0.0629

$\nabla^2 g_\sigma [m, n]$ avec $\sigma = 3$



min =-0.0039

$\nabla^2 g_\sigma [m, n]$ avec $\sigma = 4.5$



min = -7.7625e-04

Source :

©Marie-Flavie Auclair-Fortier

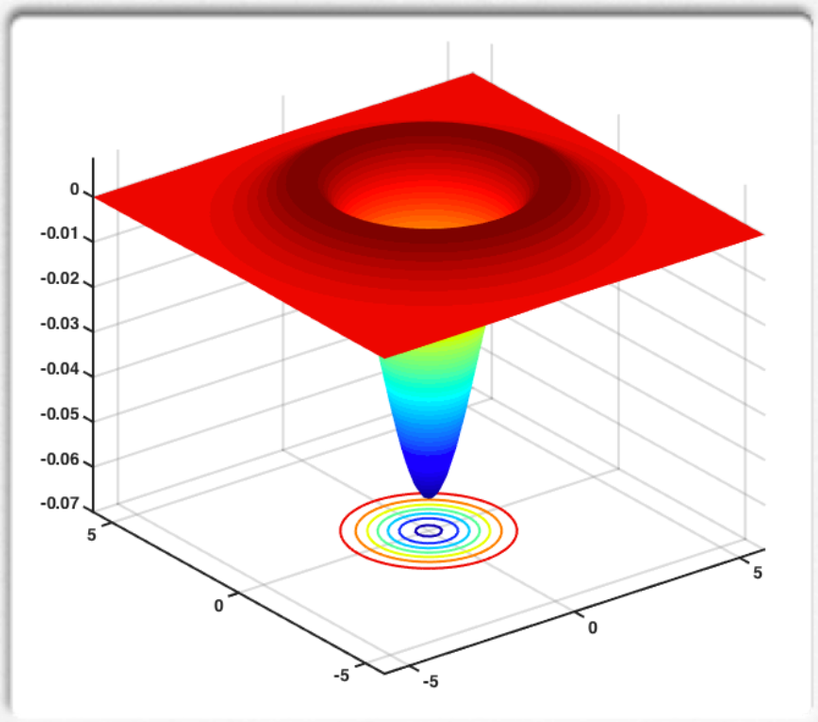
3. CARACTÉRISTIQUES PONCTUELLES

2. Laplacien

Remarque :

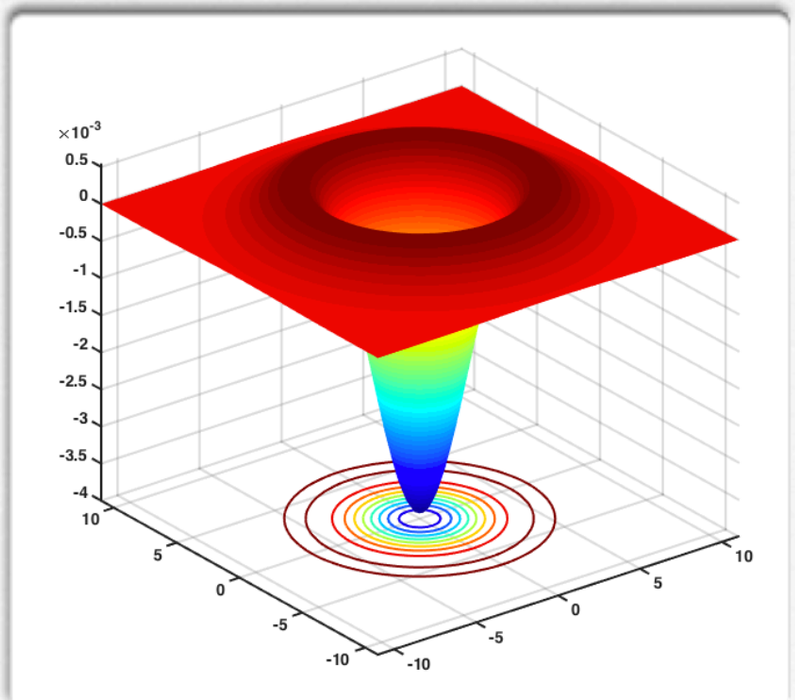
- en variant l'échelle, le rayon du masque augmente mais l'intensité diminue

$\nabla^2 g_\sigma[m,n]$ avec $\sigma = 1.5$



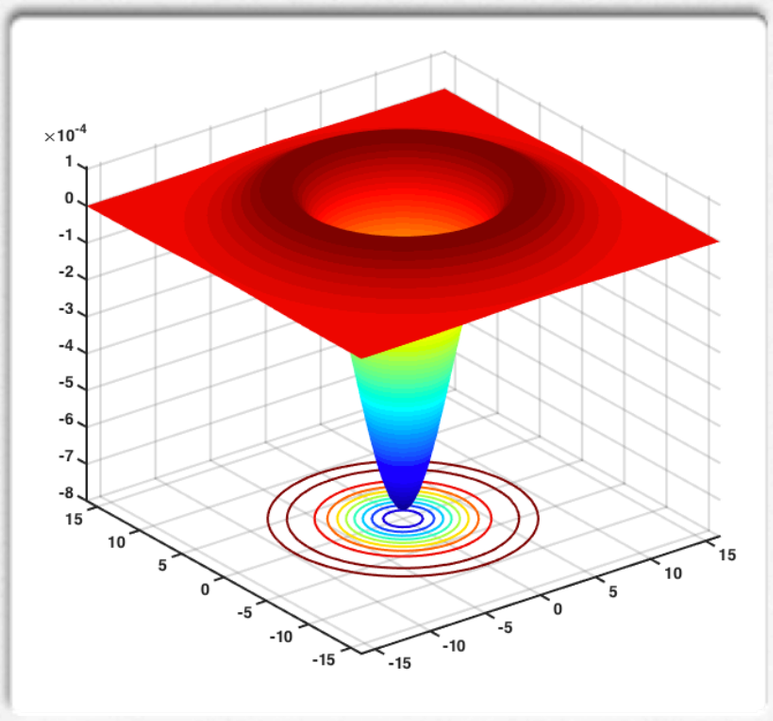
min = -0.0629

$\nabla^2 g_\sigma[m,n]$ avec $\sigma = 3$



min =-0.0039

$\nabla^2 g_\sigma[m,n]$ avec $\sigma = 4.5$



min = -7.7625e-04

Source :

©Marie-Flavie Auclair-Fortier

Zero-Crossing du Laplacien

Algorithme du *zero-crossing*

1: Calculer le laplacien de l'image d'entrée.

$$L(x,y) = \frac{\partial^2 f(x,y)}{\partial^2 x} + \frac{\partial^2 f(x,y)}{\partial^2 y}$$

2: POUR CHAQUE pixel (x,y) de l'image FAIRE

2a.

SI il existe un voisin (i, j) du pixel (x, y) telque

$L(i,j) > 0$ ET $L(x,y) \leq 0$ ALORS

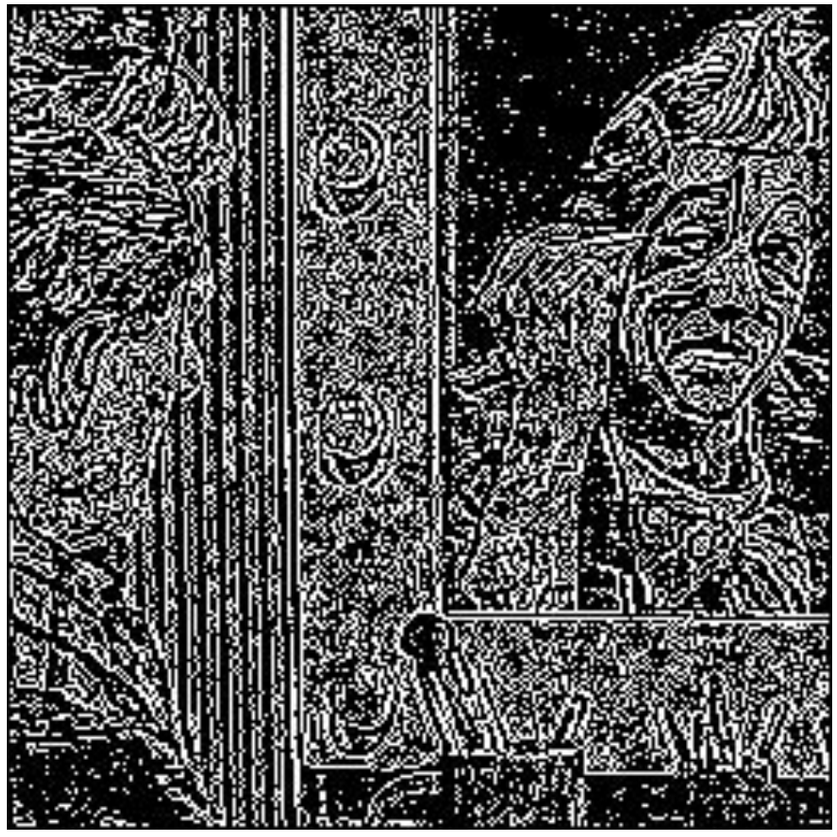
$c(x,y) = 1$

SINON

$c(x,y) = 0$

Zero-Crossing du Laplacien

Le zero-crossing est *TRÈS* sensible au bruit.



Zero-Crossing avec LoG (Laplacian of Gaussian)

Au lieu de prendre le laplacien de l'image d'entrée :

$$L(x,y) = \frac{\partial^2 f(x,y)}{\partial^2 x} + \frac{\partial^2 f(x,y)}{\partial^2 y}$$

On calcule le laplacien de l'image filtrée par une gaussienne

$$g(x,y) = G_\sigma * f$$

$$L_G(x,y) = \frac{\partial^2 g(x,y)}{\partial^2 x} + \frac{\partial^2 g(x,y)}{\partial^2 y}$$

Zero-Crossing du Laplacien

Algorithme du zero-crossing

1. Convoluer l'image d'entrée à l'aide d'une gaussienne

$$g = G_\sigma * f$$

2. Calculer le laplacien de l'image convoluée.

$$L(x,y) = \frac{\partial^2 g(x,y)}{\partial^2 x} + \frac{\partial^2 g(x,y)}{\partial^2 y}$$

3. POUR CHAQUE pixel (x,y) de l'image FAIRE

3a.

SI il existe un voisin (i, j) du pixel (x, y) telque

(*) $L(i,j) < 0$ ET $L(x,y) \geq 0$ ALORS

$c(x,y) = 1$

SINON

$c(x,y) = 0$

Note : pour rendre l'algorithme encore plus robuste au bruit, on peut remplacer la ligne (*) par

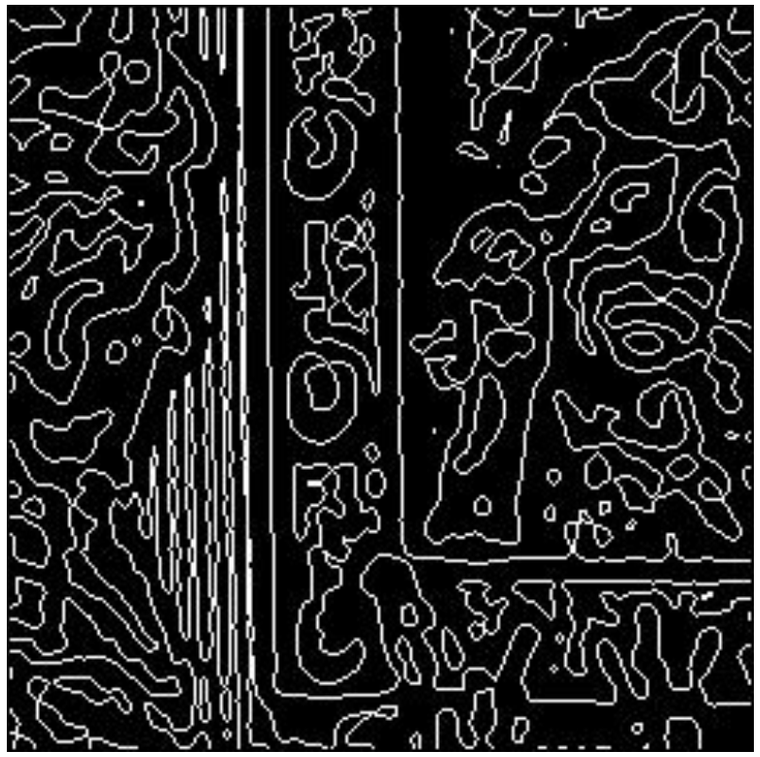
$$L(i,j) < -Seuil \text{ ET } L(x,y) \geq Seuil$$

Zero-Crossing avec LoG (Laplacian of Gaussian)

σ = *1*

σ = *2*

σ = *3*



Il est aussi possible de mettre un **<u>seuil supérieur à zéro</u>** pour ne conserver que les transitions franches

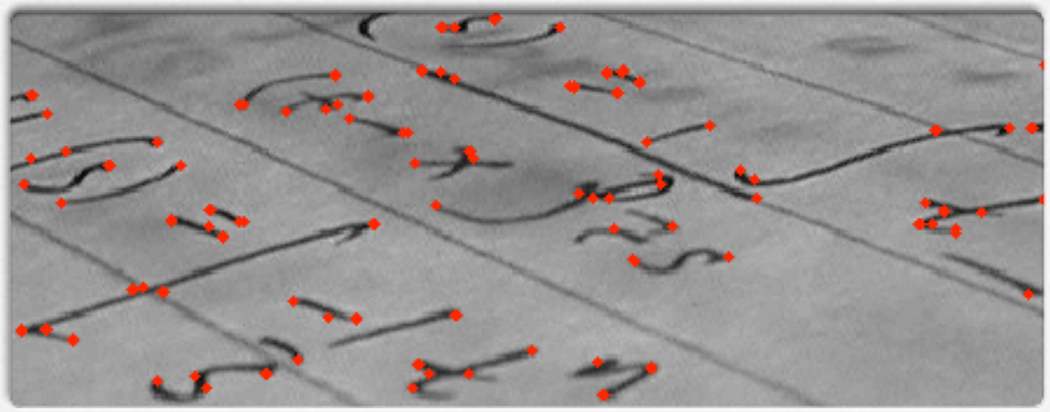
Plan

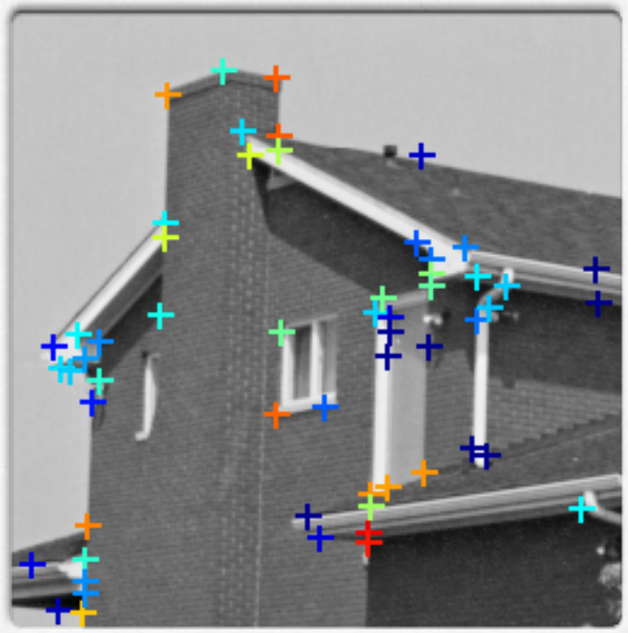
1. Calcul des dérivées d'une image
2. Contours
3. Coins
   - 3.1. Tenseur de structure
   - 3.2. Laplacien
4. Segmentation

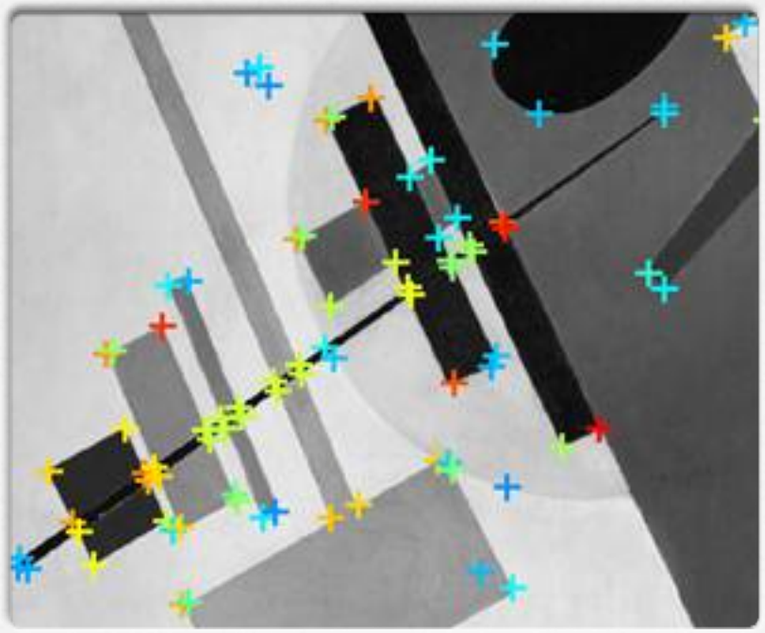
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

- Des points d'intérêts
  - ✓ Ne changent pas d'aspect même si rotation, translation, changements d'échelle, perspective



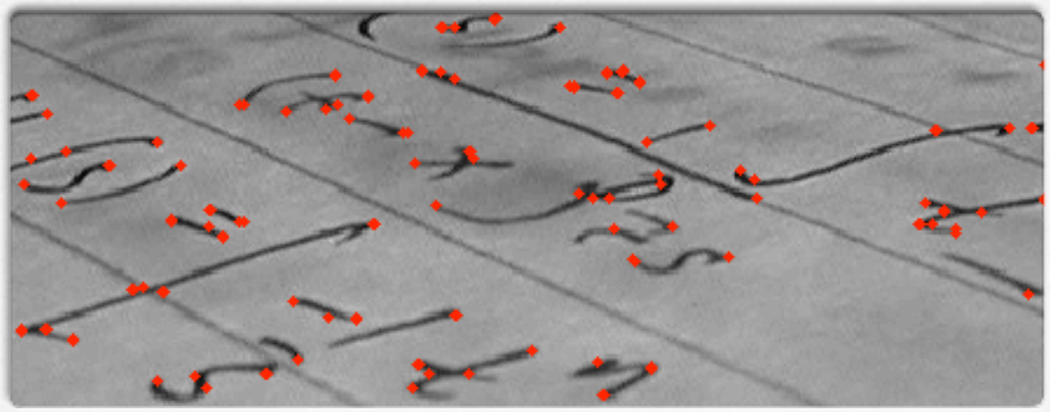


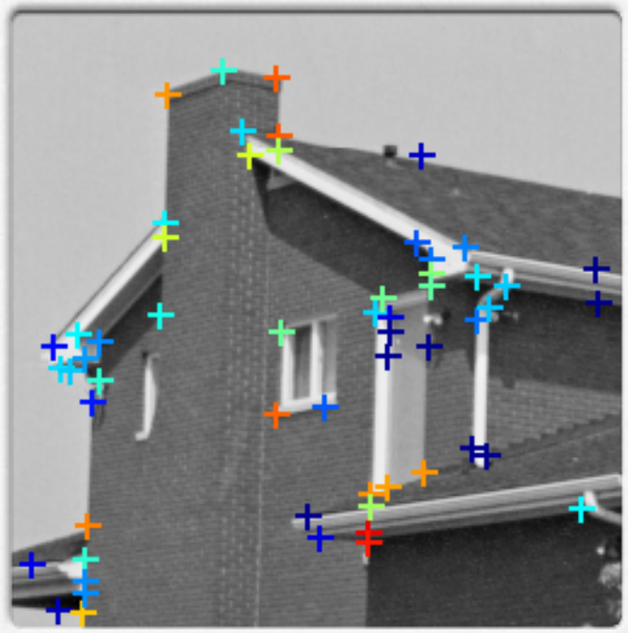


Source : G Xia, J. Delon et Y. Gousseau. Accurate and contrast invariant junction detection, CNRS Paris, France

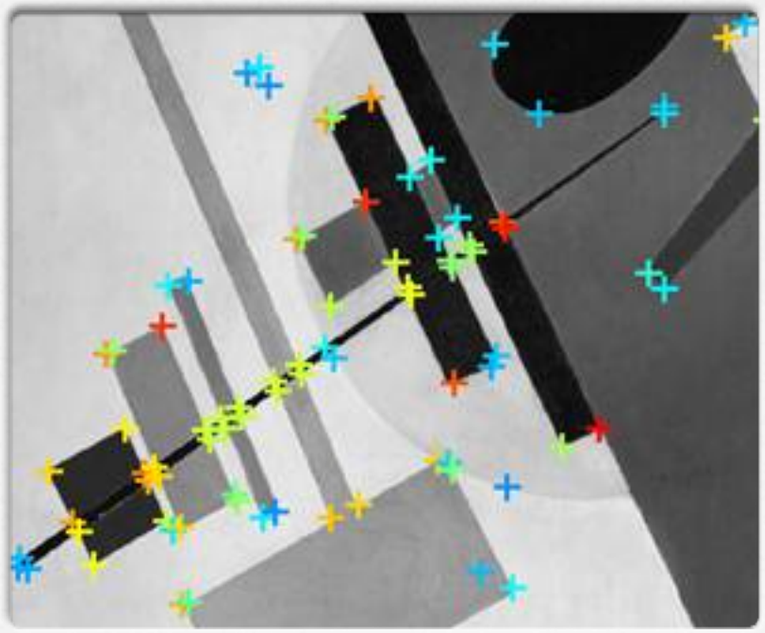
3. Caractéristiques ponctuelles

- Des points d'intérêts
  - ✓ Ne changent pas d'aspect même si rotation, translation, changements d'échelle, perspective





Coins ou jonctions en L

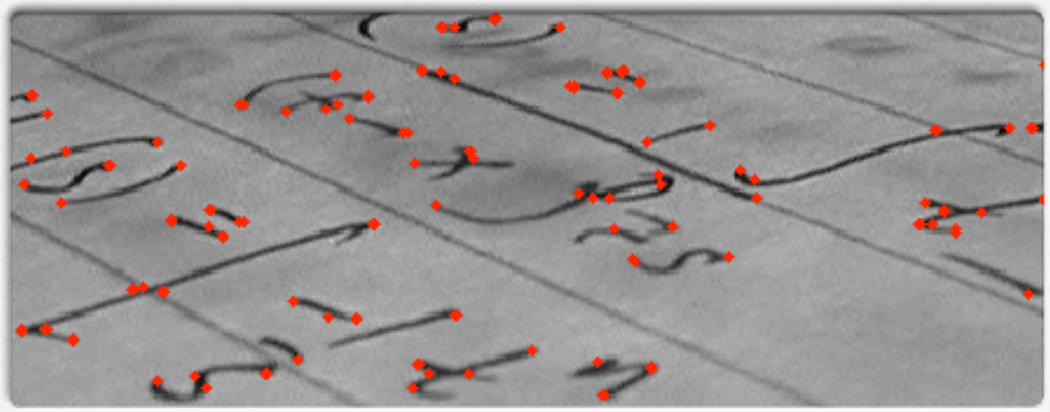


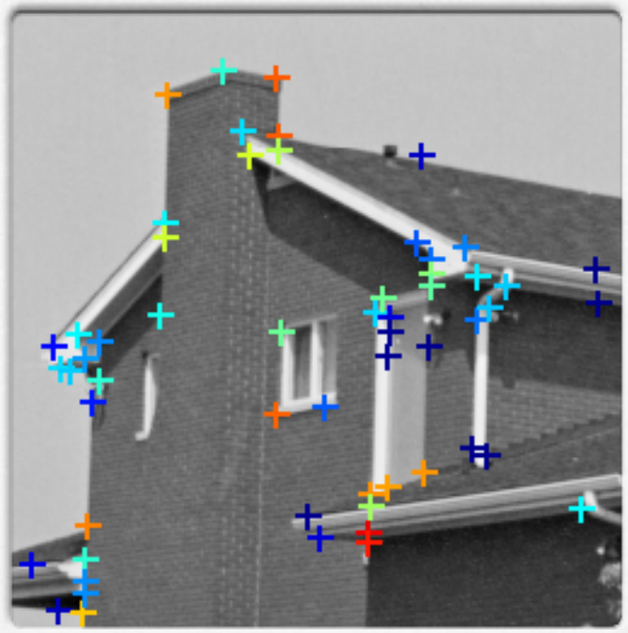
Source : G Xia, J. Delon et Y. Gousseau. Accurate and contrast invariant junction detection, CNRS Paris, France

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

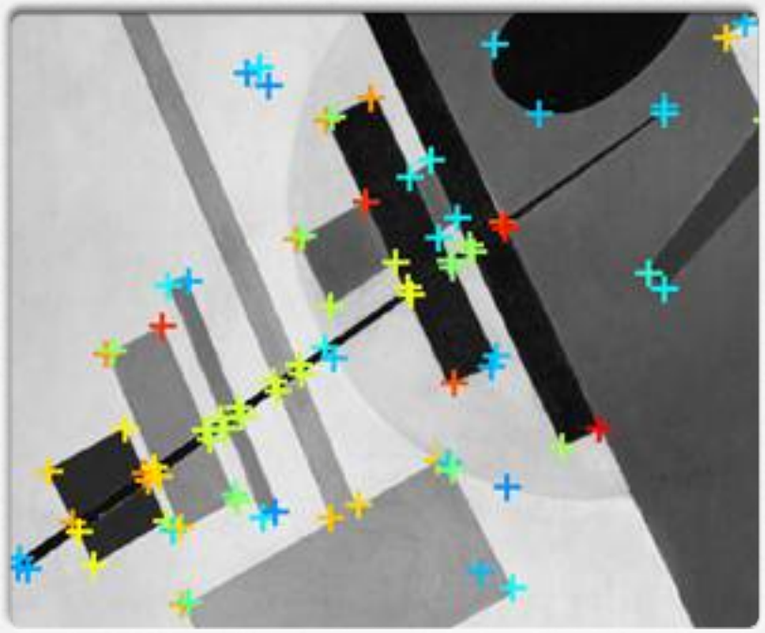
- Des points d'intérêts
  - ✓ Ne changent pas d'aspect même si rotation, translation, changements d'échelle, perspective





Coins ou jonctions en L

Jonction en T

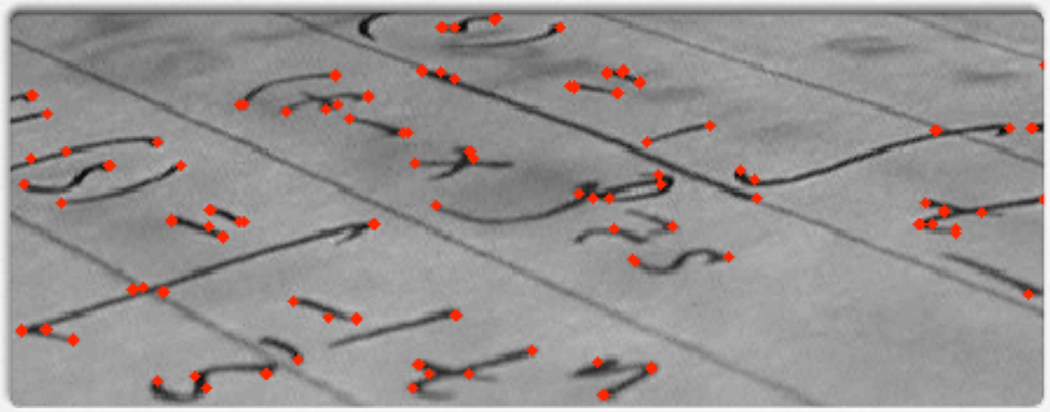


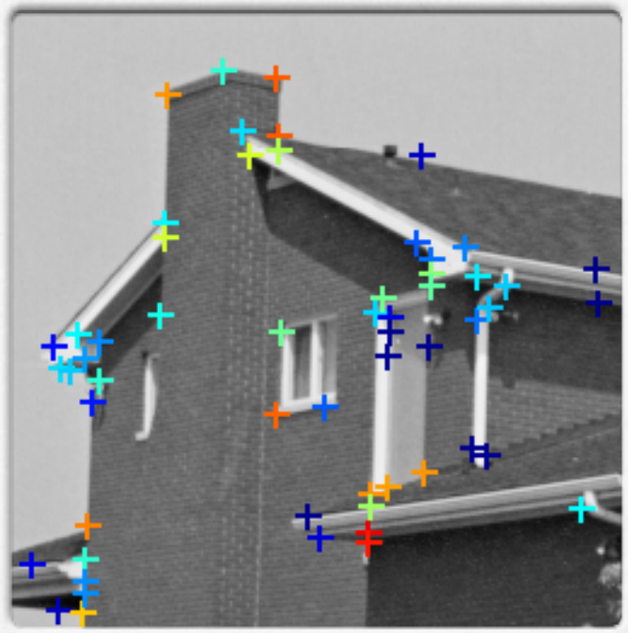
Source : G Xia, J. Delon et Y. Gousseau. Accurate and contrast invariant junction detection, CNRS Paris, France

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

- Des points d'intérêts
  - ✓ Ne changent pas d'aspect même si rotation, translation, changements d'échelle, perspective

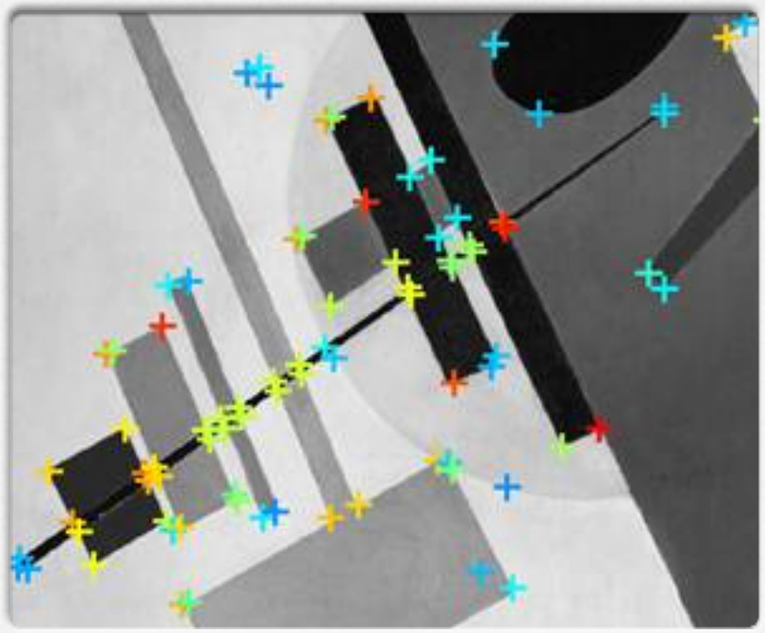




Coins ou jonctions en L

Jonction en T

Jonction en Y



Source : G Xia, J. Delon et Y. Gousseau. Accurate and contrast invariant junction detection, CNRS Paris, France

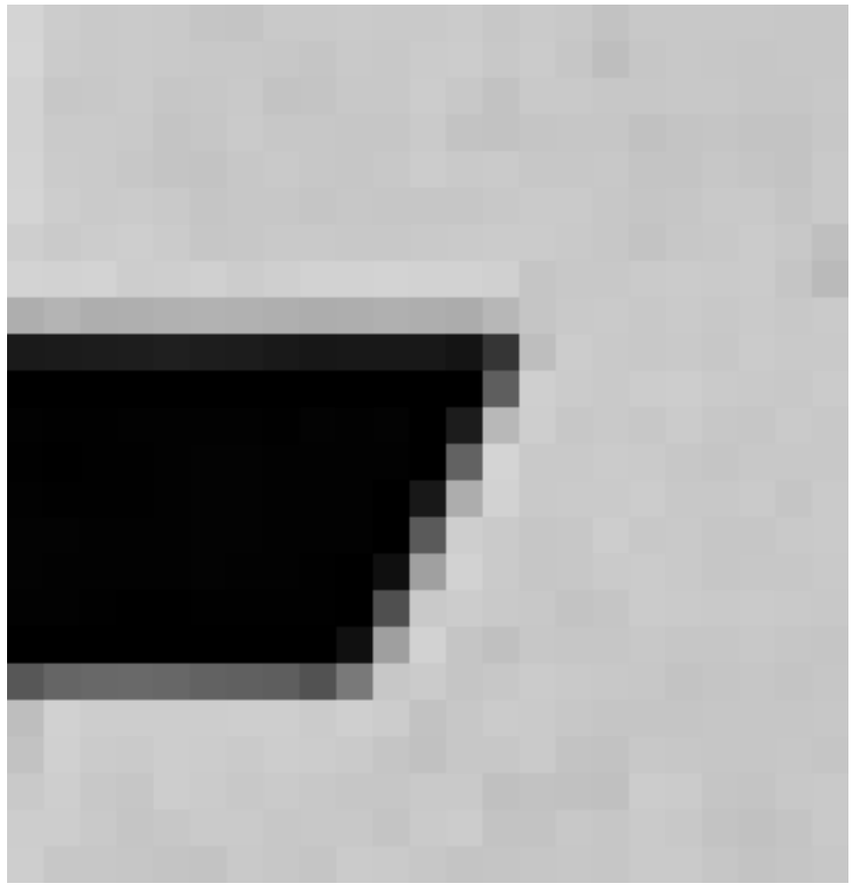
©Marie-Flavie Auclair-Fortier

Détecteur de coins

Qu'est-ce qu'un coin? Une région de l'image comprenant une discontinuité ponctuelle.

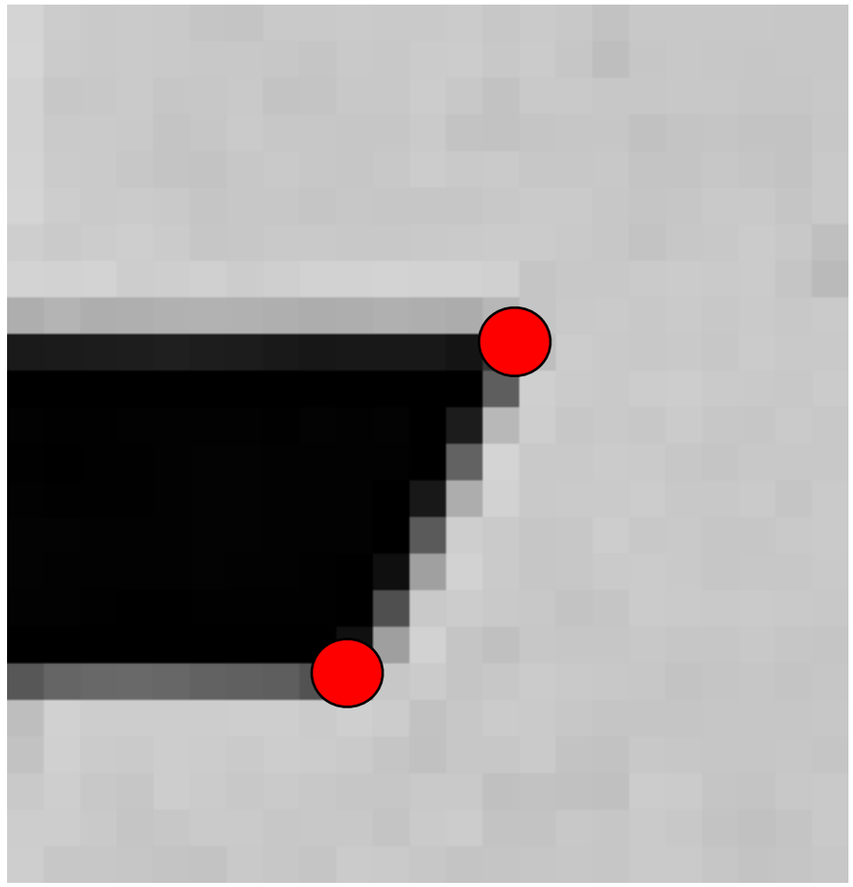
Détecteur de coins

Qu'est-ce qu'un coin? Une région de l'image comprenant une discontinuité ponctuelle.



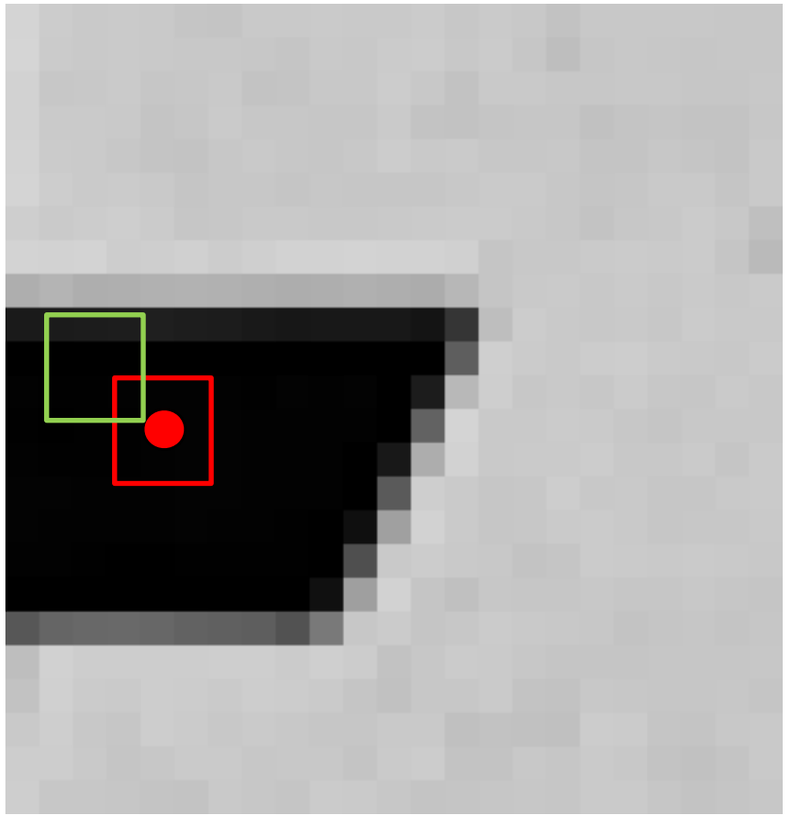
Détecteur de coins

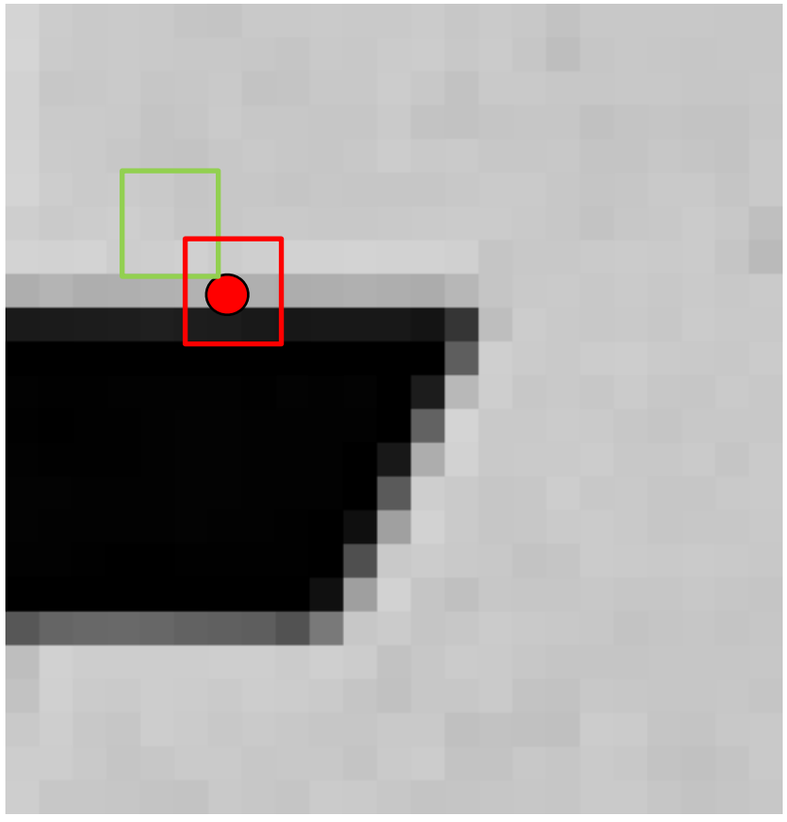
Qu'est-ce qu'un coin? Une région de l'image comprenant une discontinuité ponctuelle.

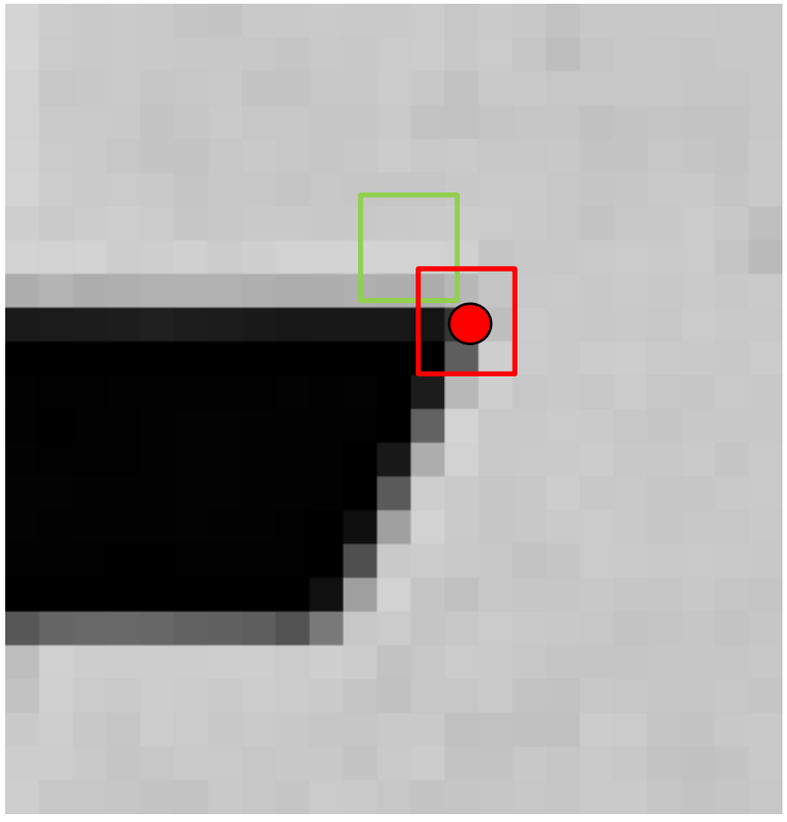


Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

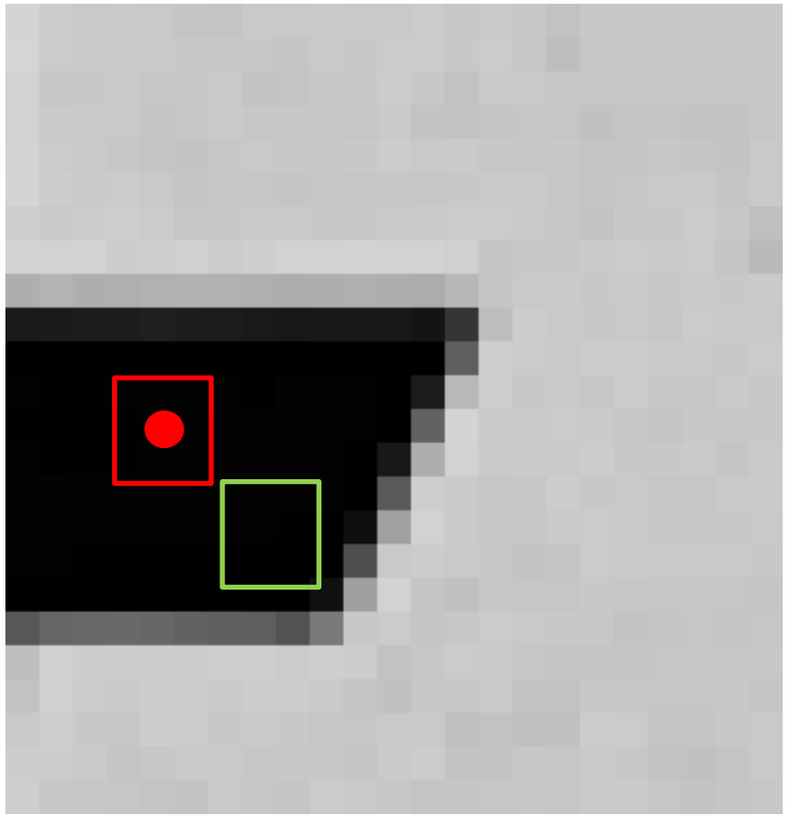


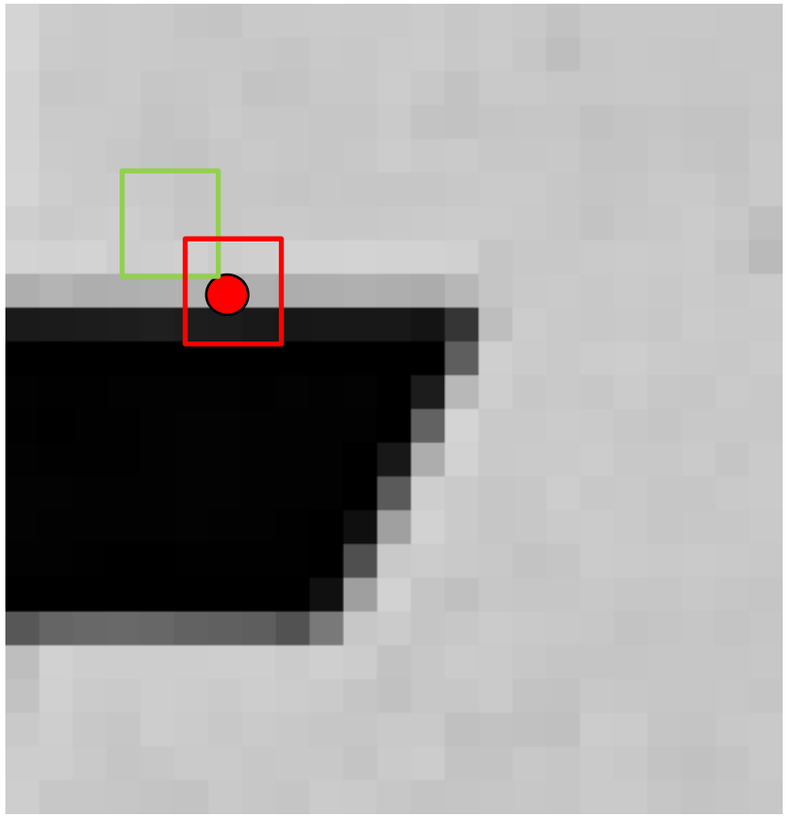


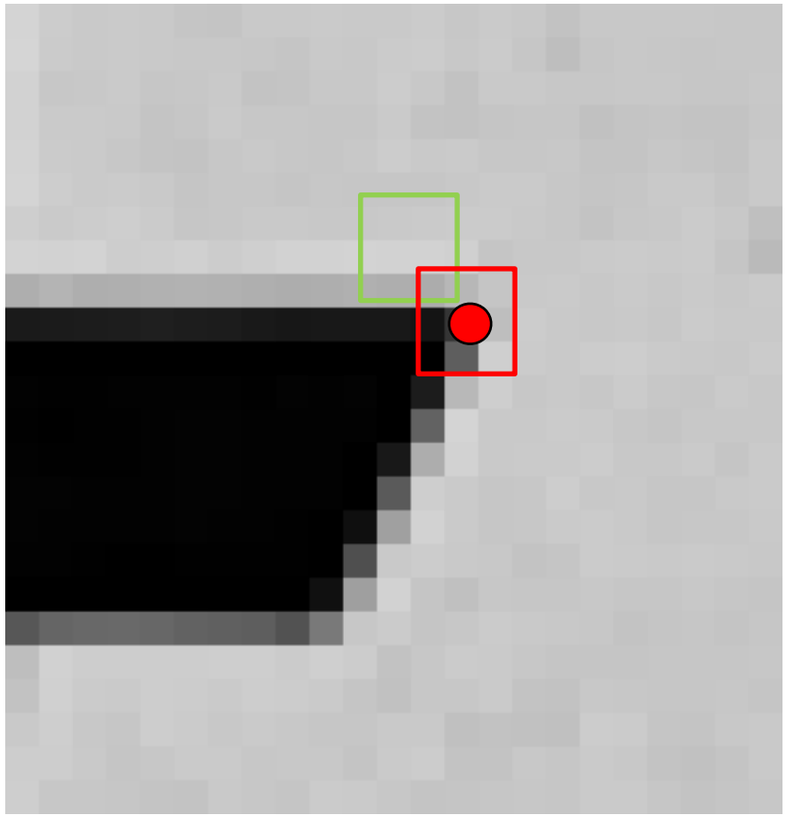


Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

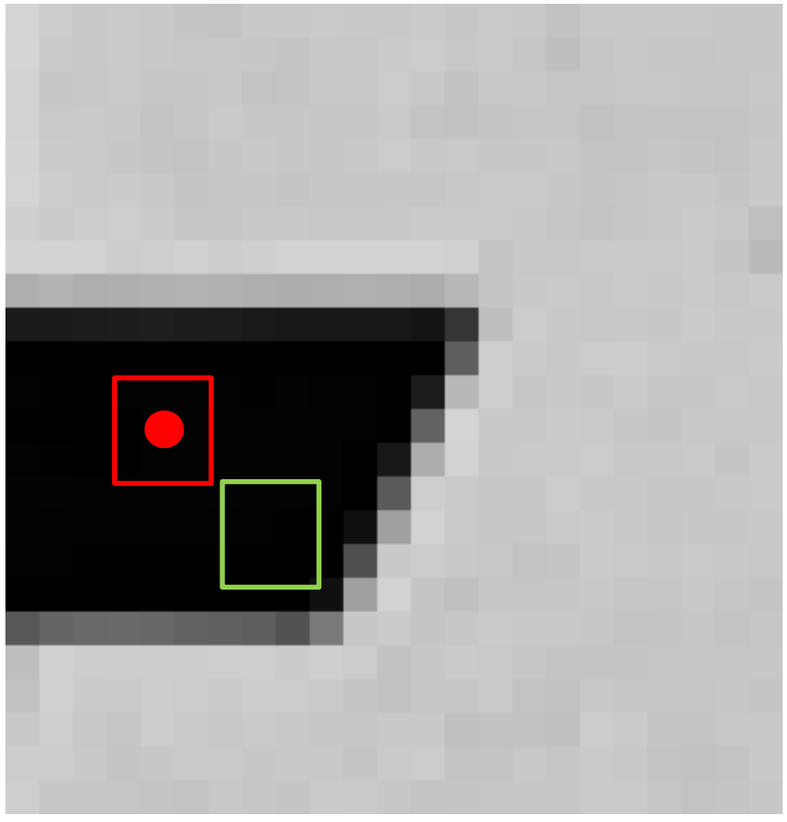


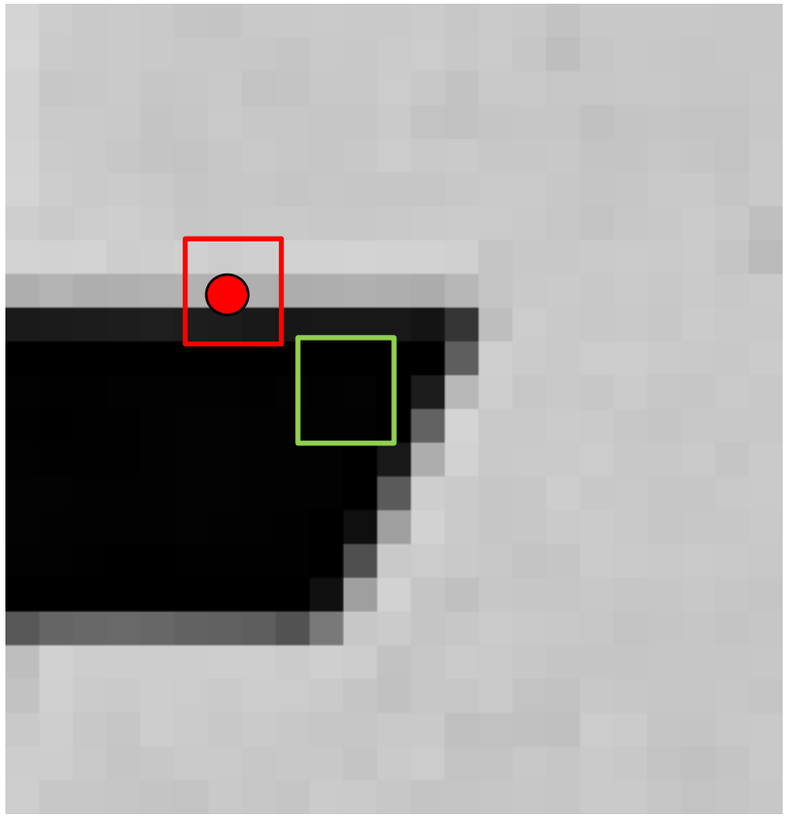


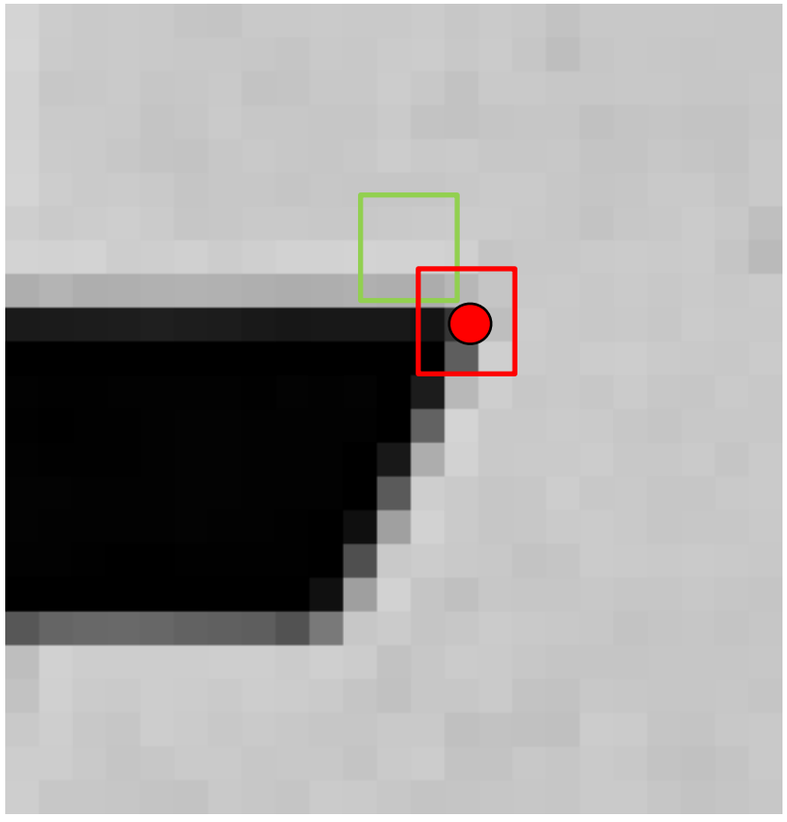


Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

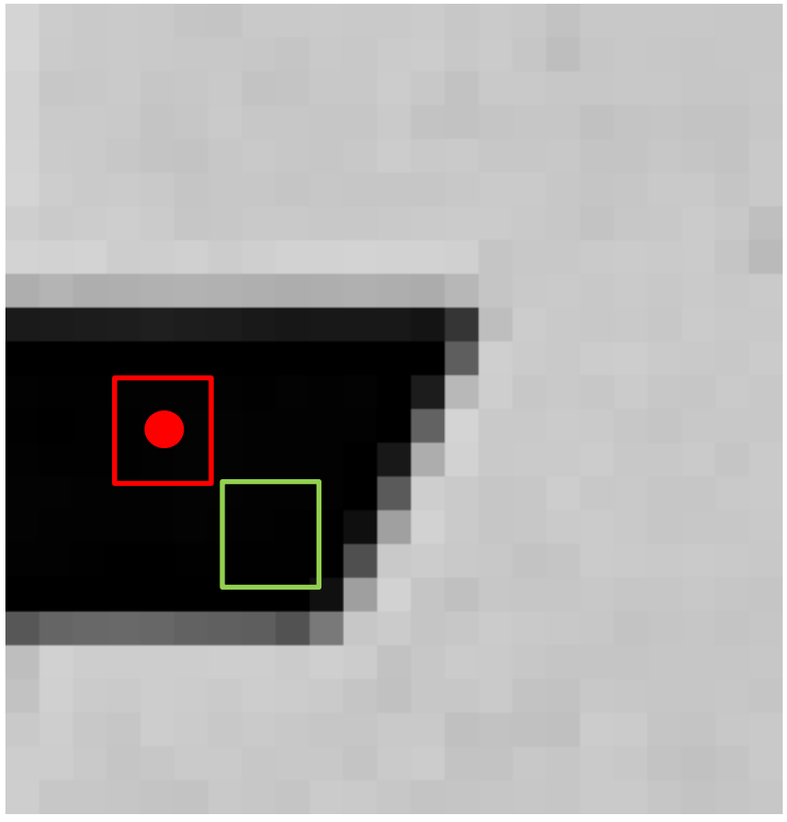


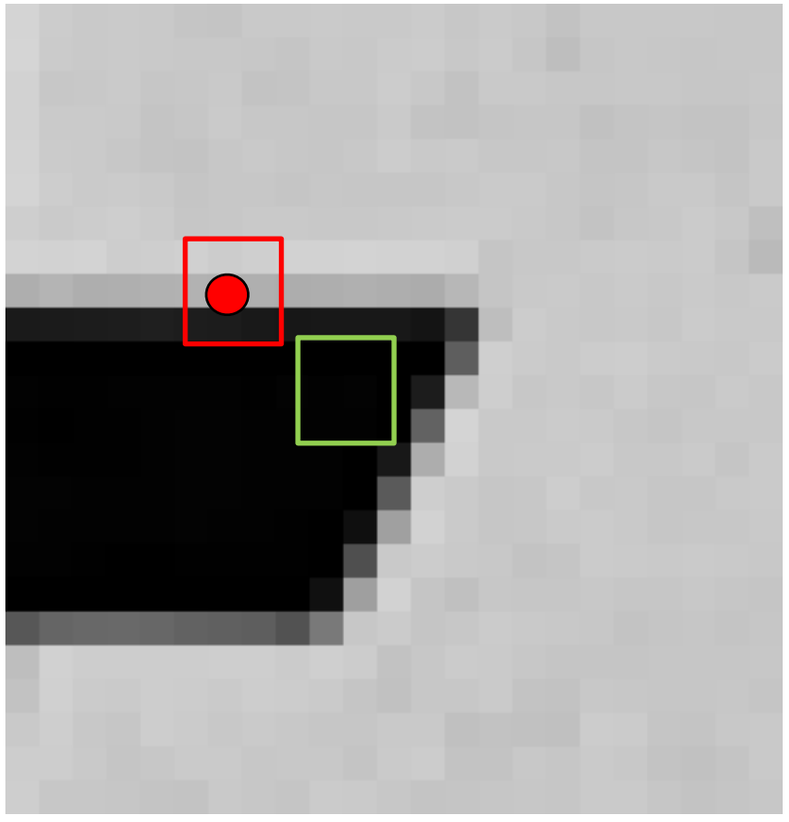


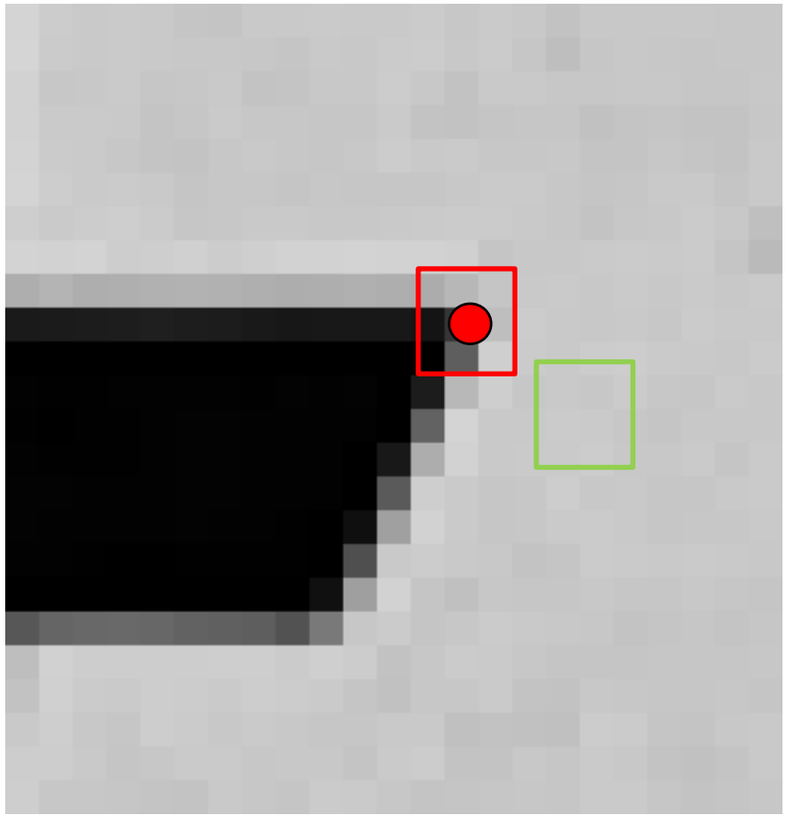


Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

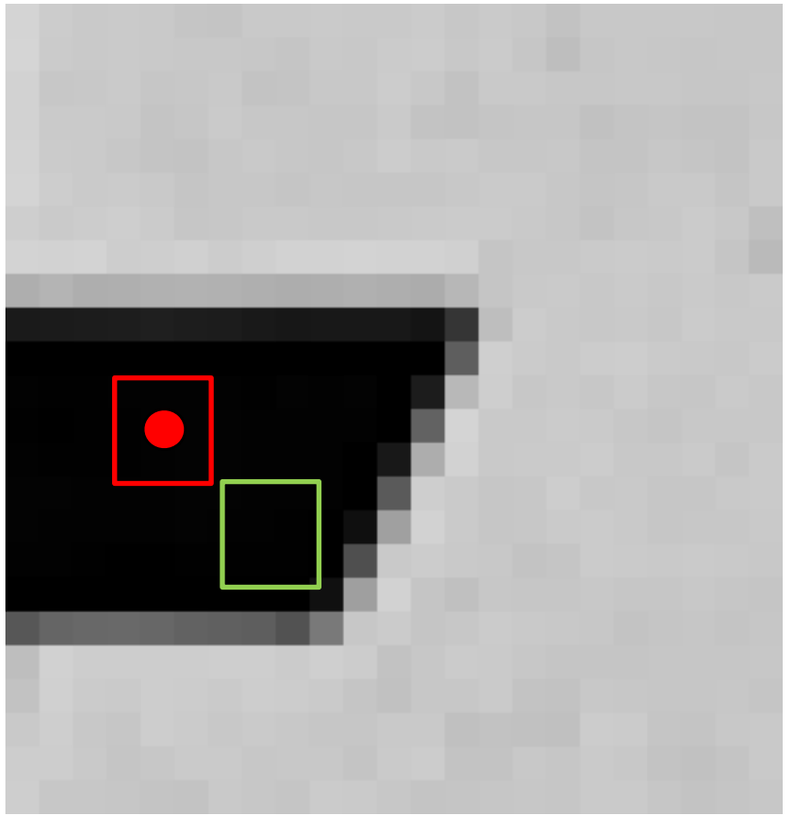


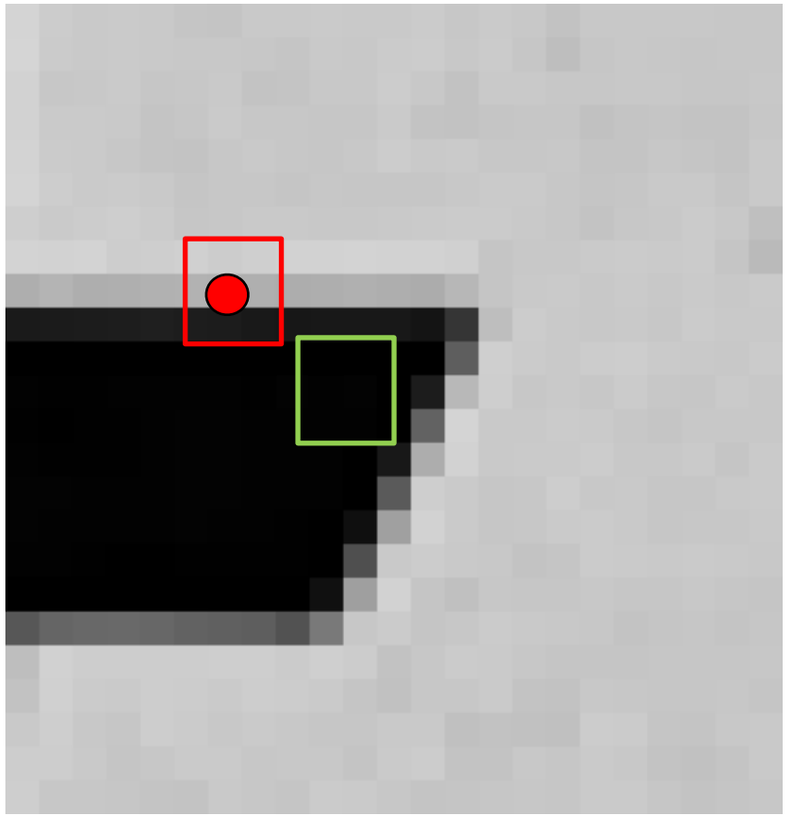


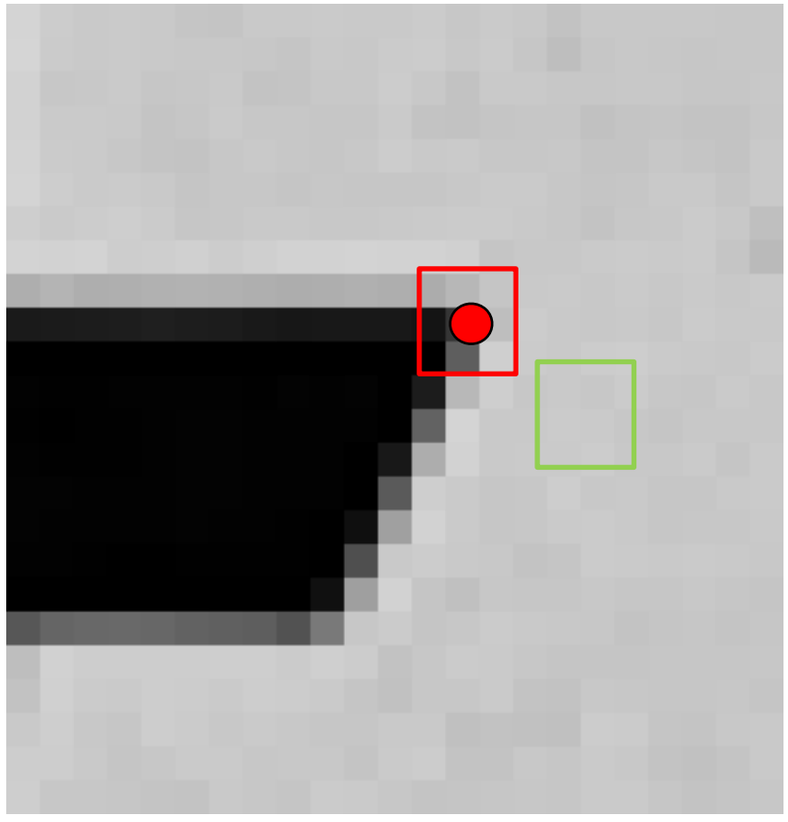


Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

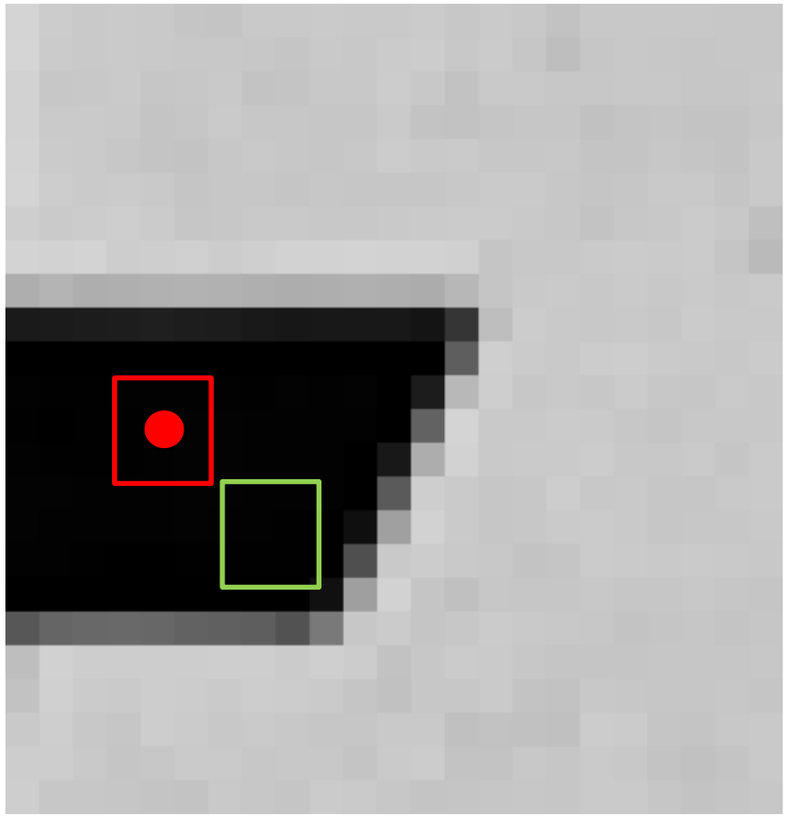


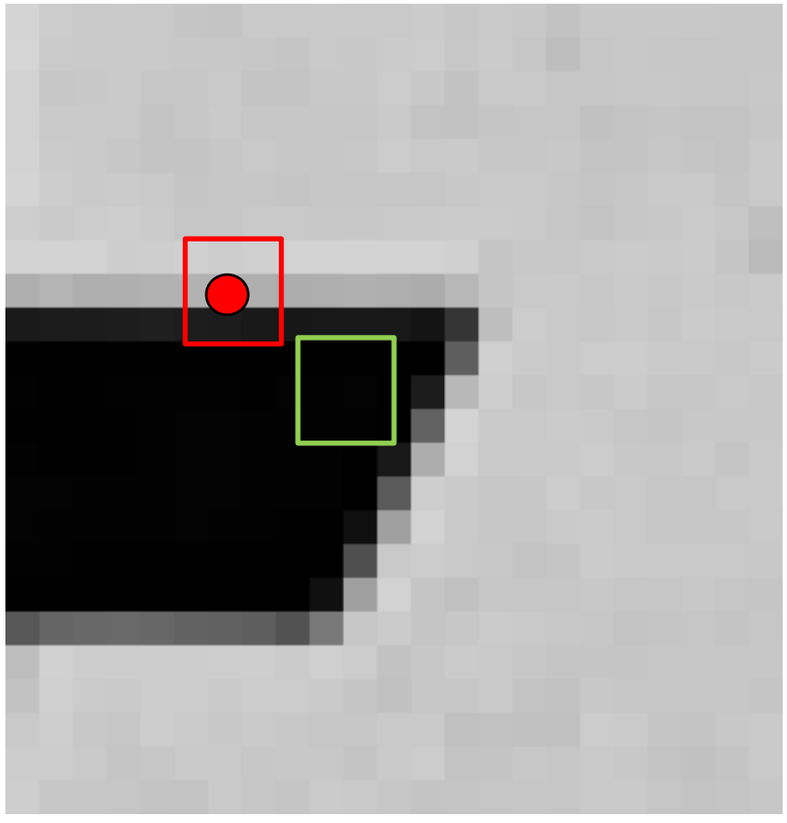


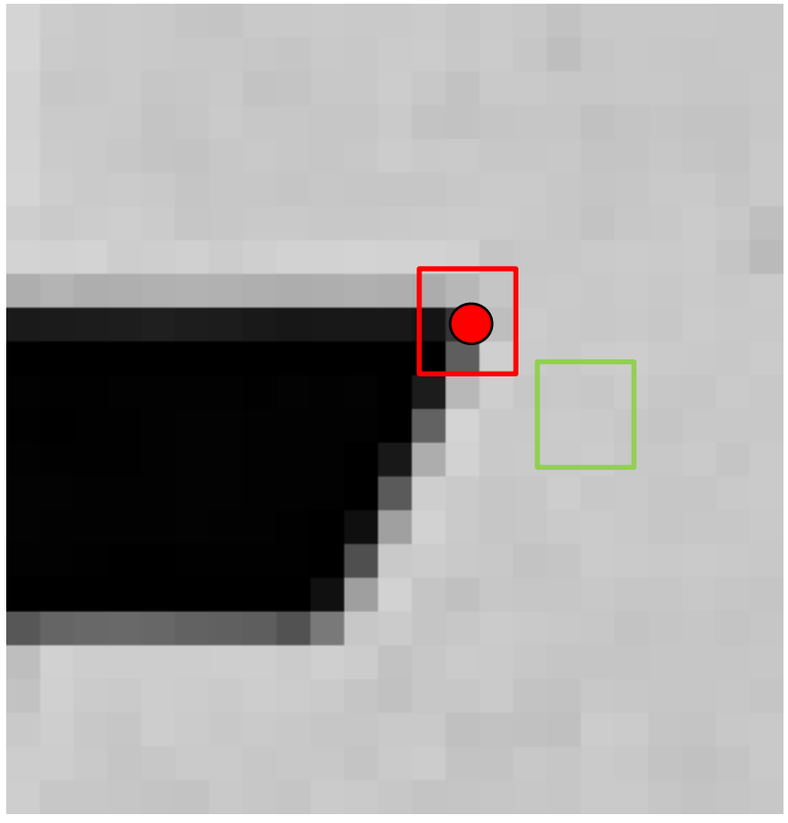


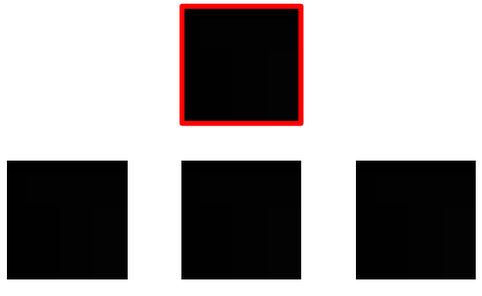
Détecteur de coins

Un coin = tout pixel dont la région est localement unique.



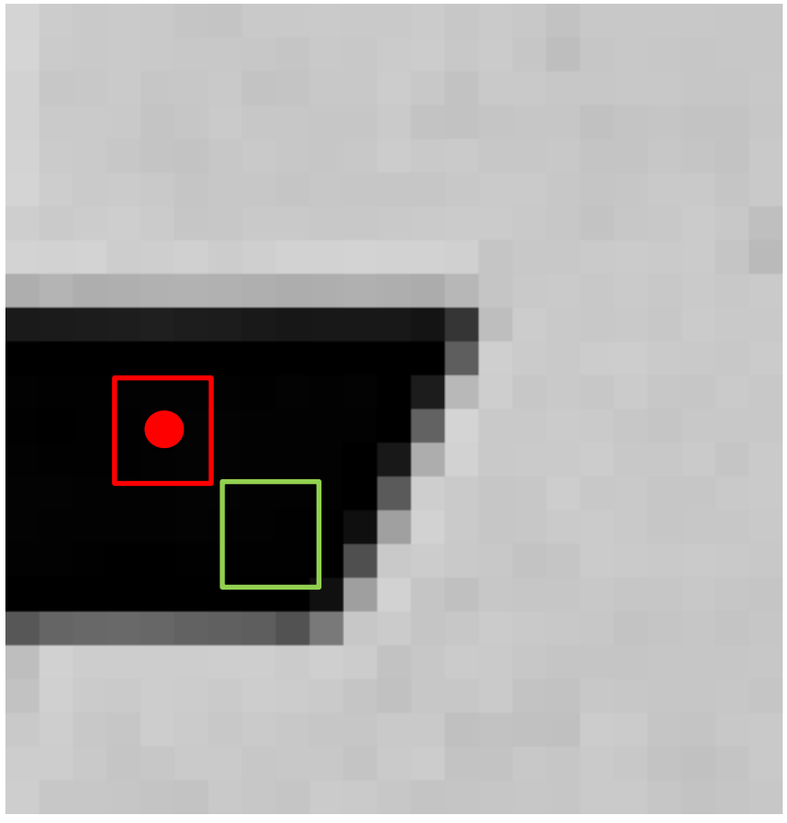


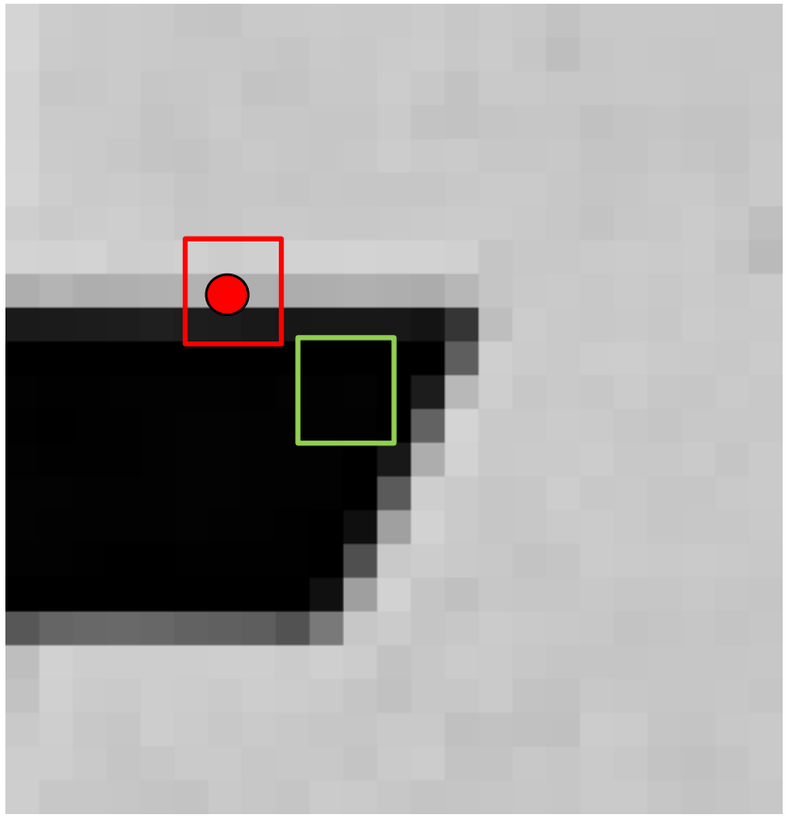


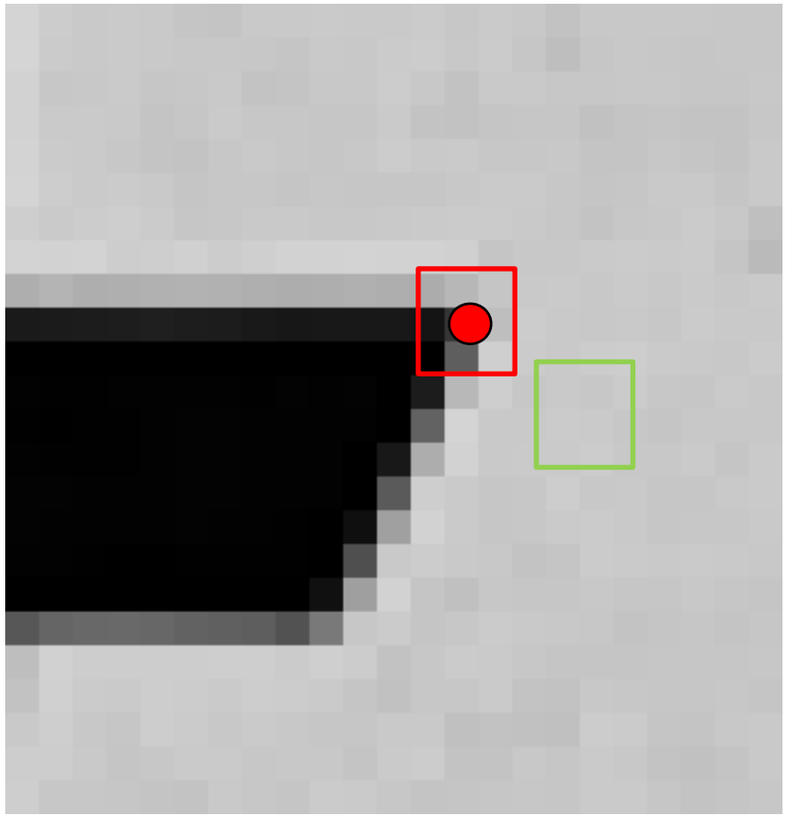


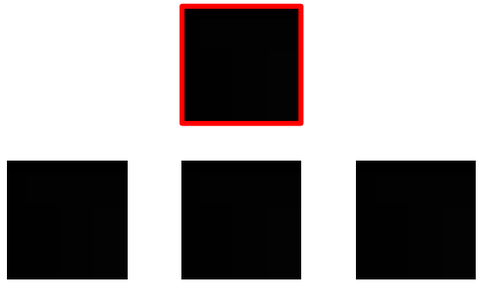
Détecteur de coins

Un coin = tout pixel dont la région est localement unique.





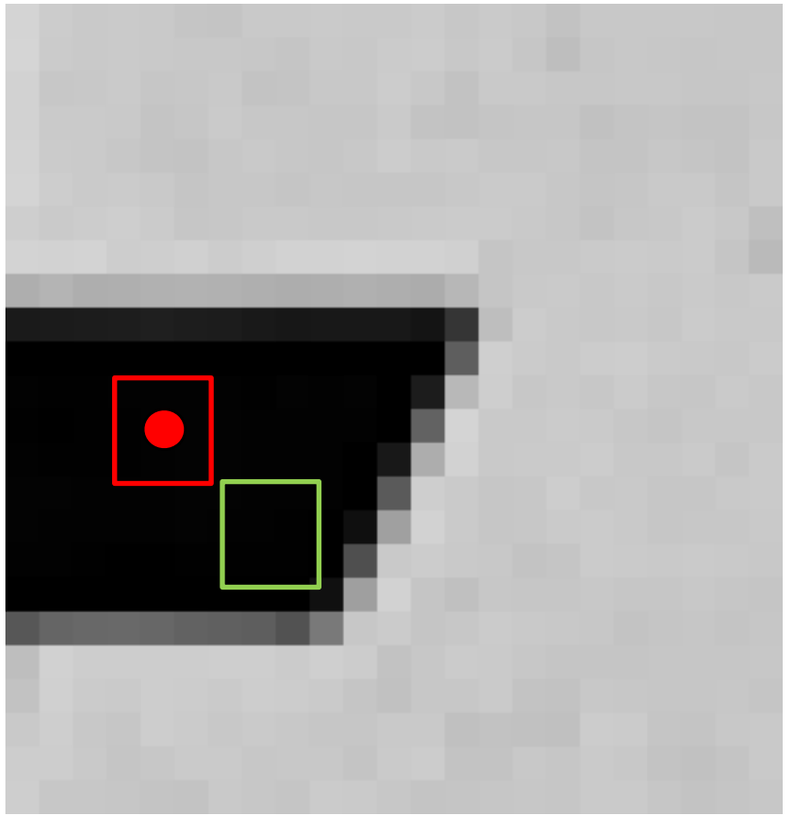


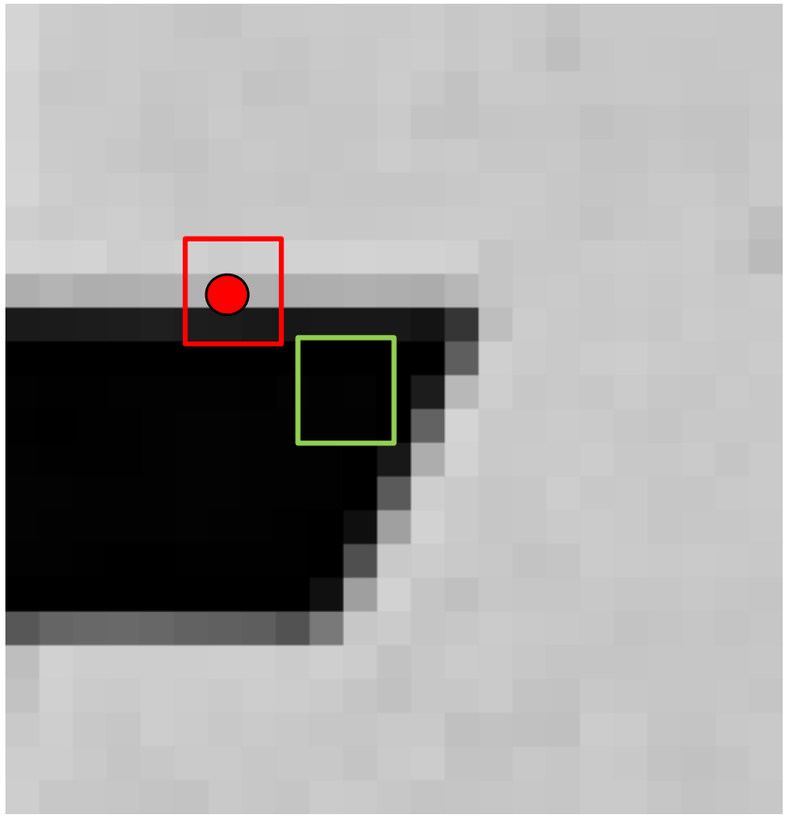


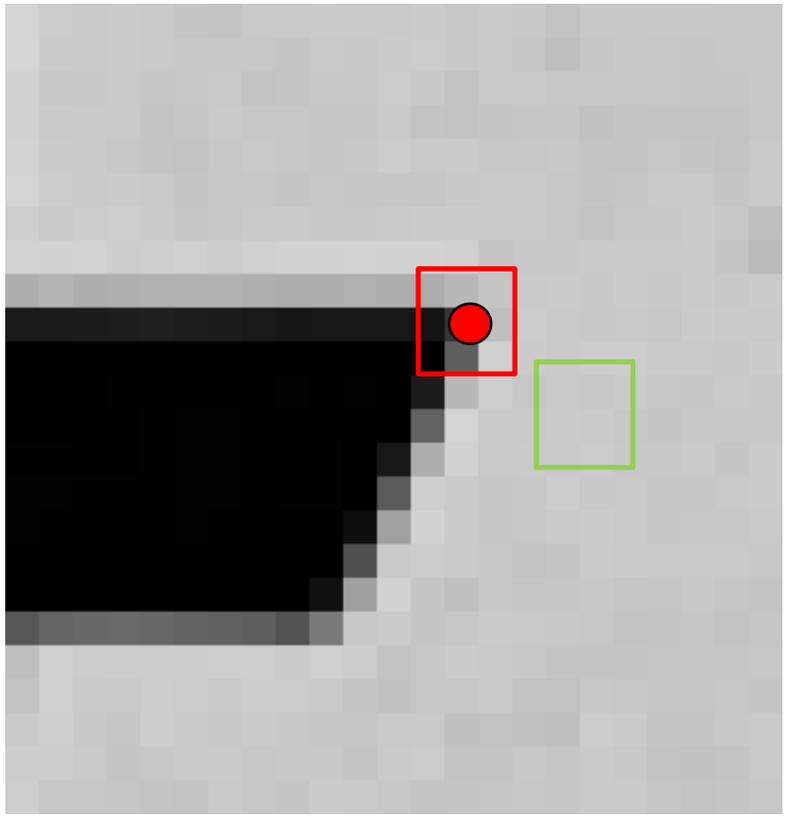
Plusieurs regions ont le même contenu que la fenêtre de référence

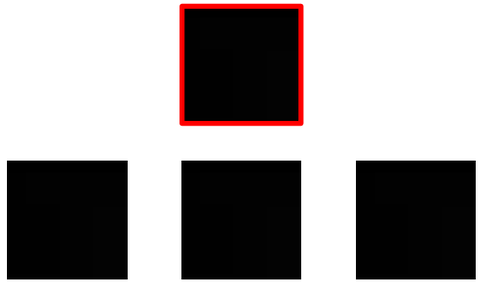
Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

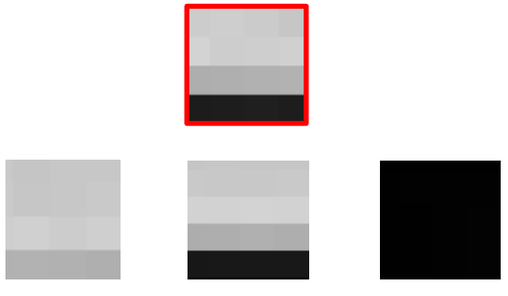






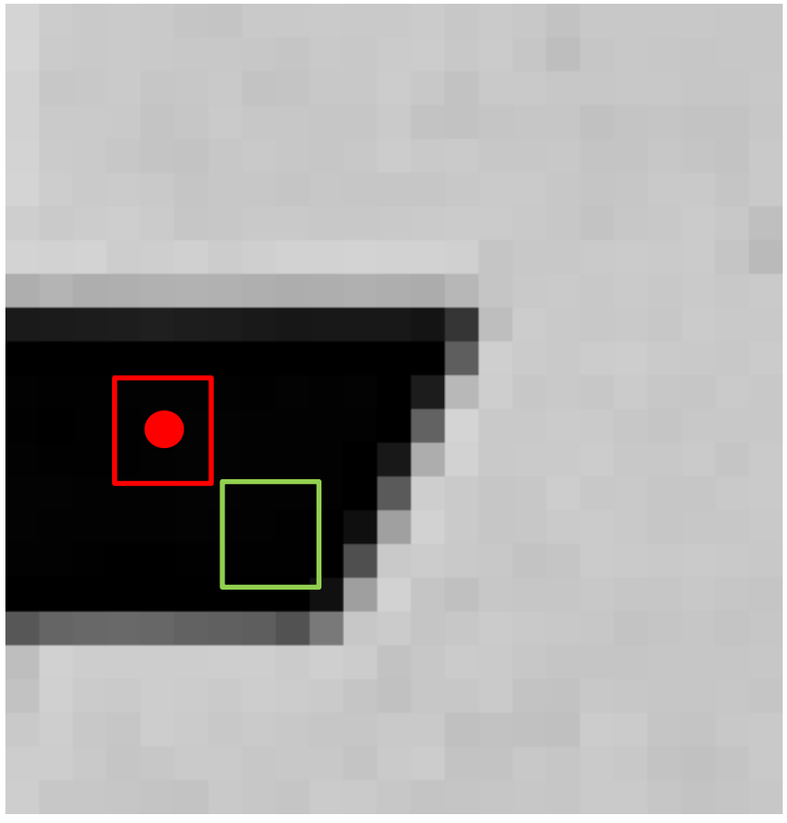


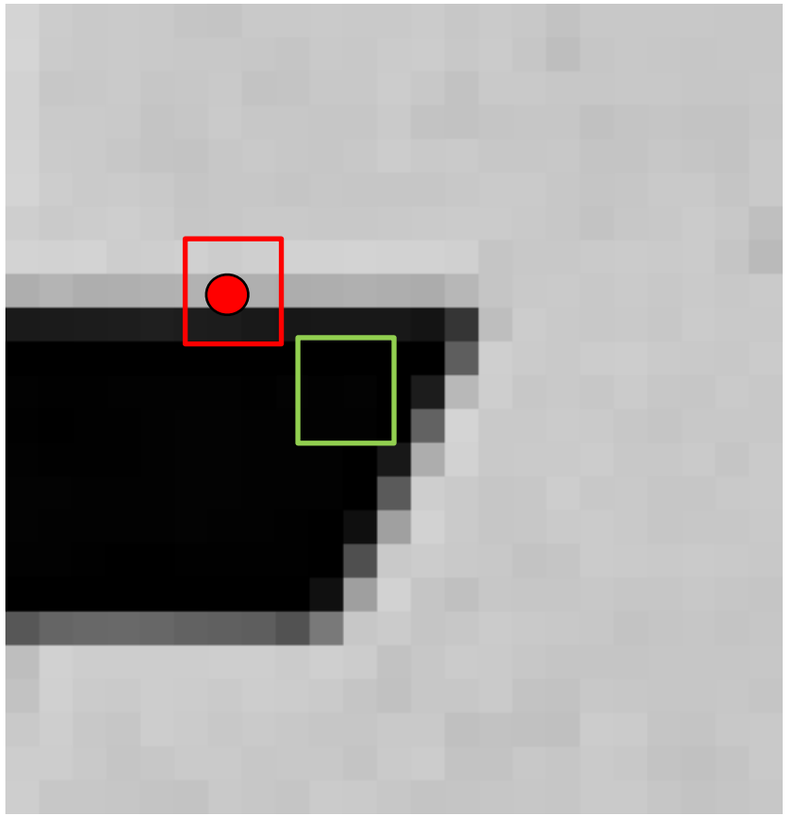
Plusieurs regions ont le même contenu que la fenêtre de référence

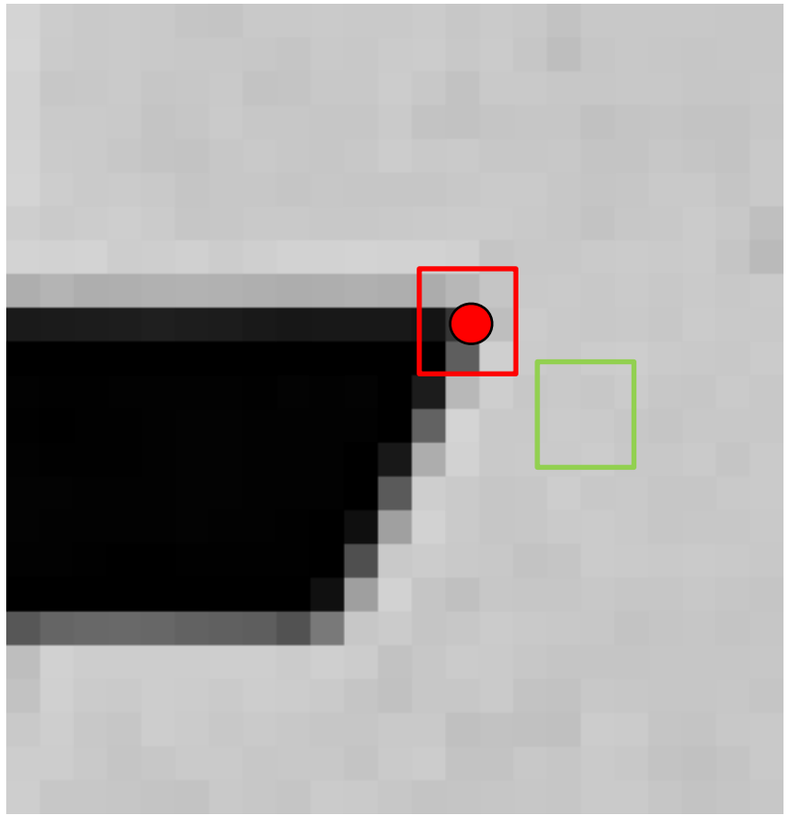


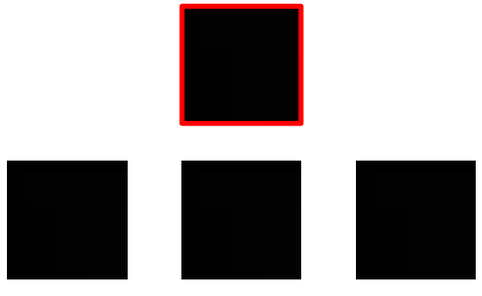
Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

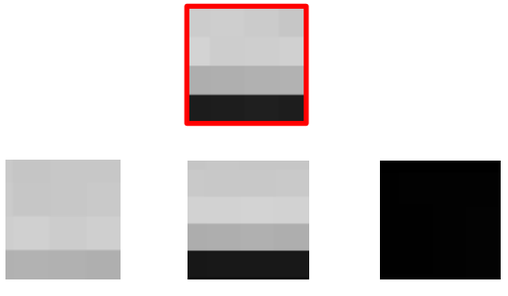








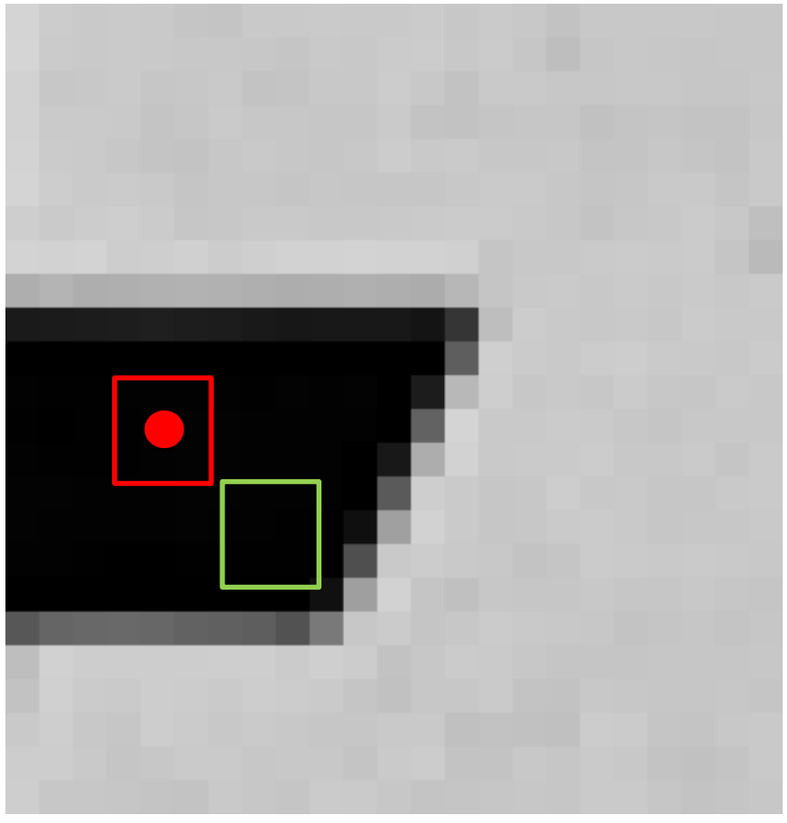
Plusieurs regions ont le même contenu que la fenêtre de référence

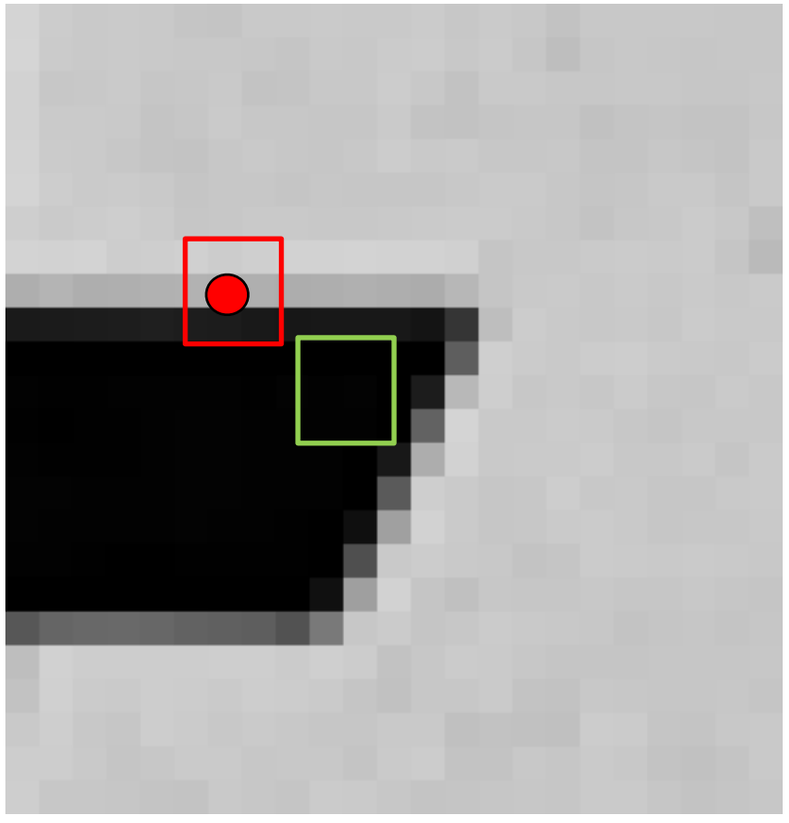


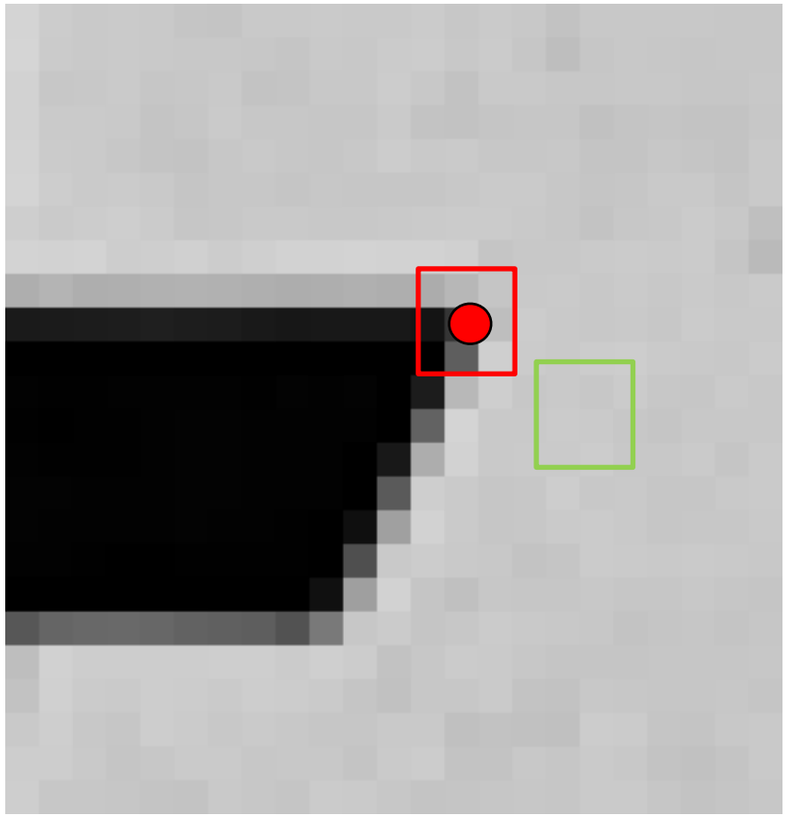
Au moins une region a le même contenu que la fenêtre de référence

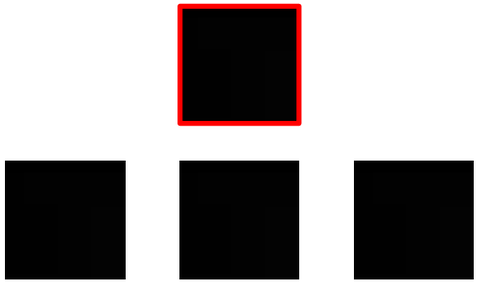
Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

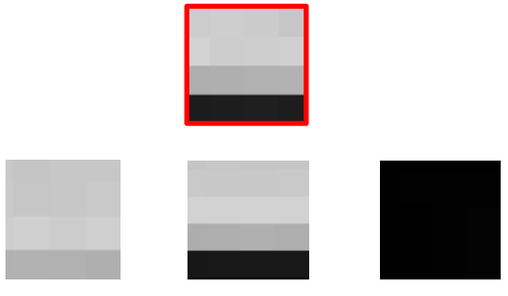




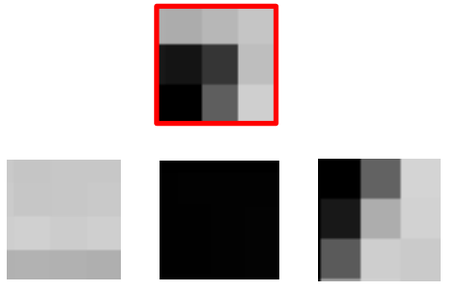




Plusieurs regions ont le même contenu que la fenêtre de référence

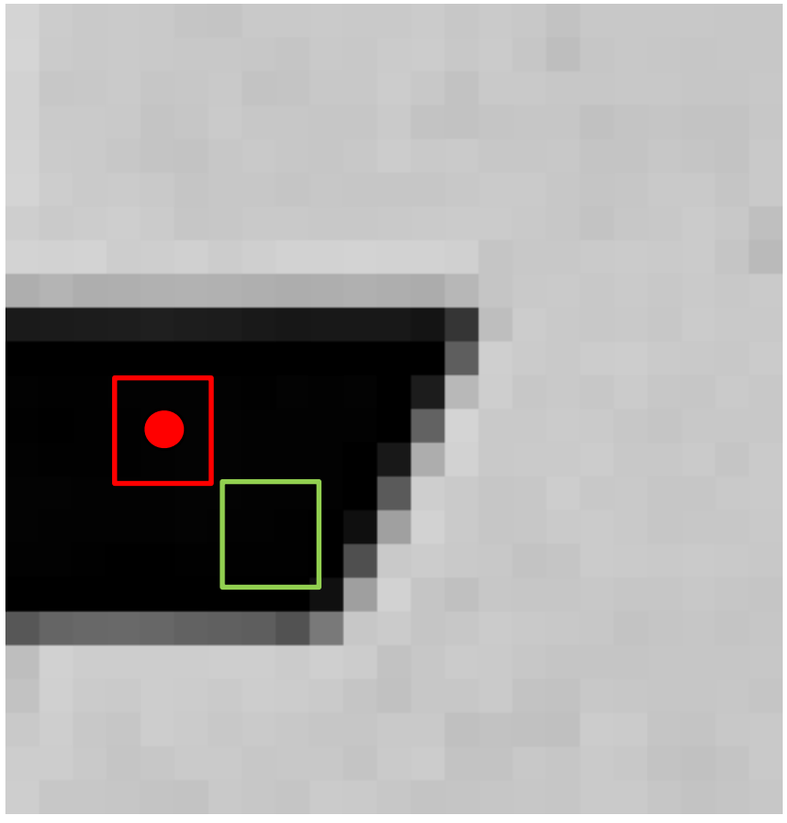


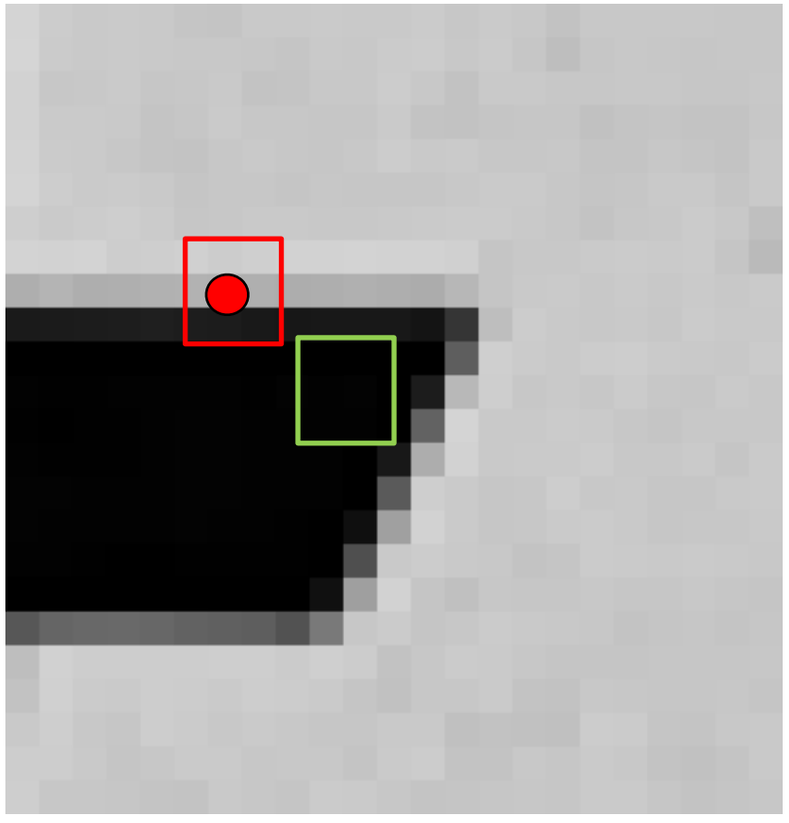
Au moins une region a le même contenu que la fenêtre de référence

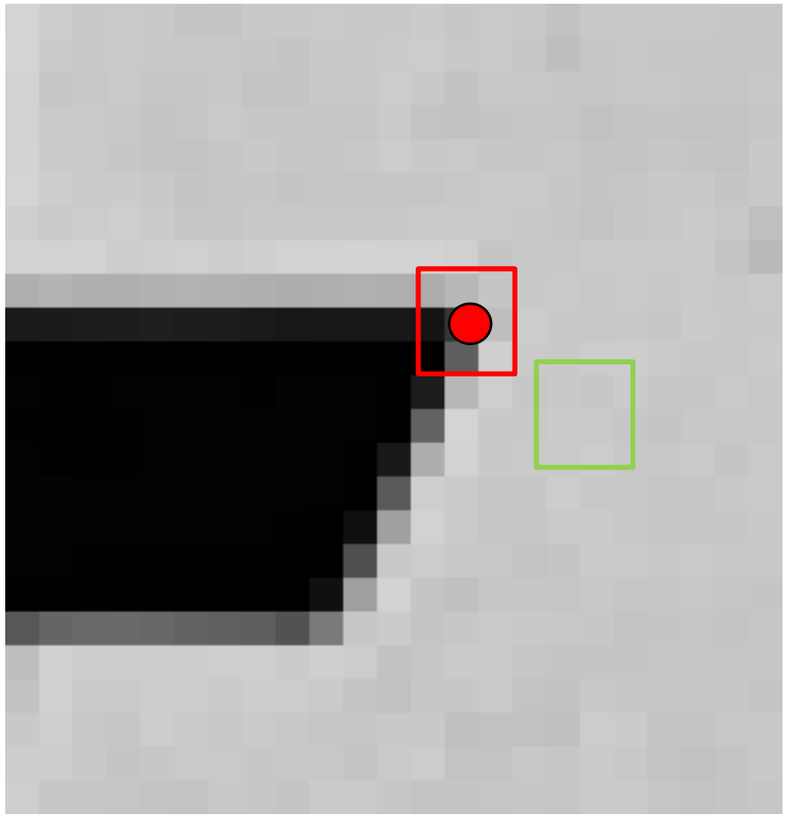


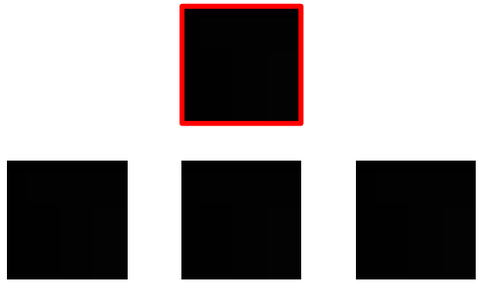
Détecteur de coins

Un coin = tout pixel dont la région est localement unique.

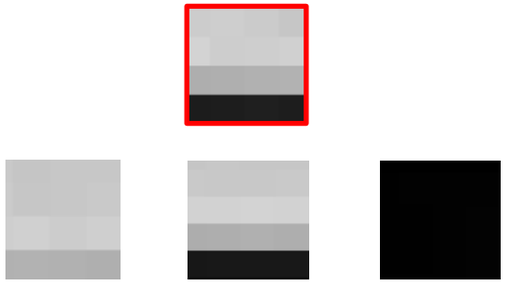




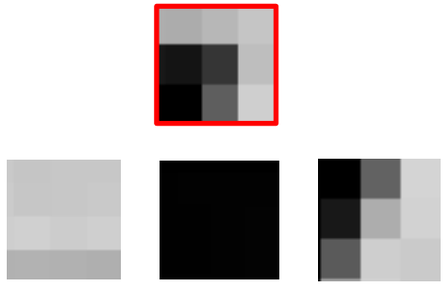




Plusieurs regions ont le même contenu que la fenêtre de référence



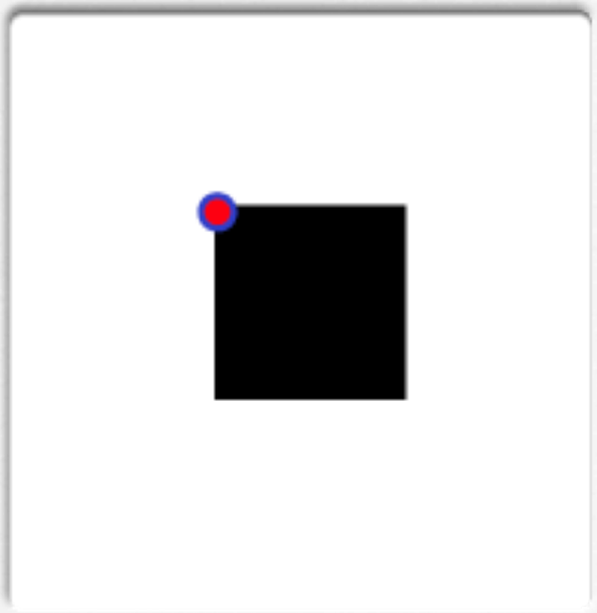
Au moins une region a le même contenu que la fenêtre de référence



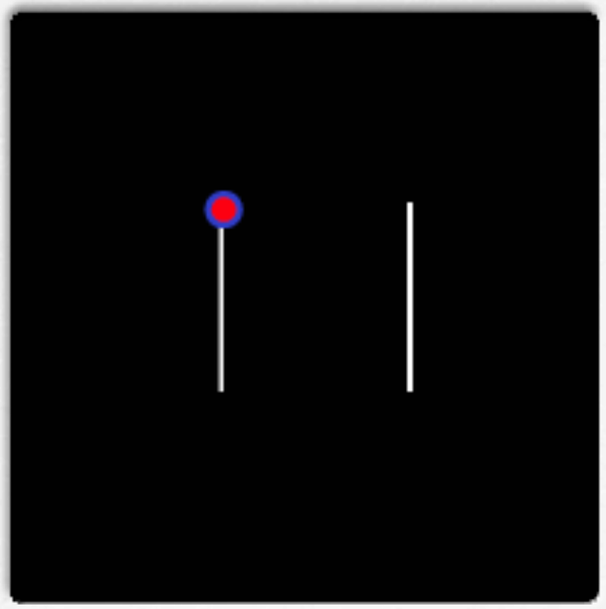
Aucune region n'a le même contenu que la fenêtre de référence

3. Caractéristiques ponctuelles

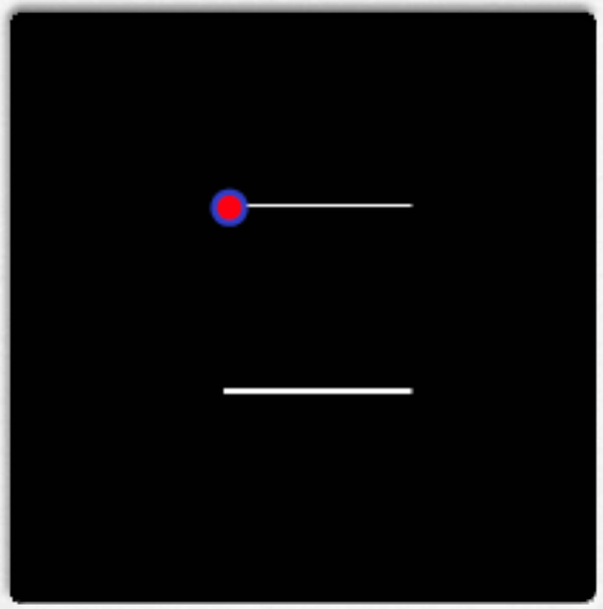
1. Tenseur de structure



$f[m,n]$



$\left(\frac{\partial f}{\partial x}[m,n]\right)^2$



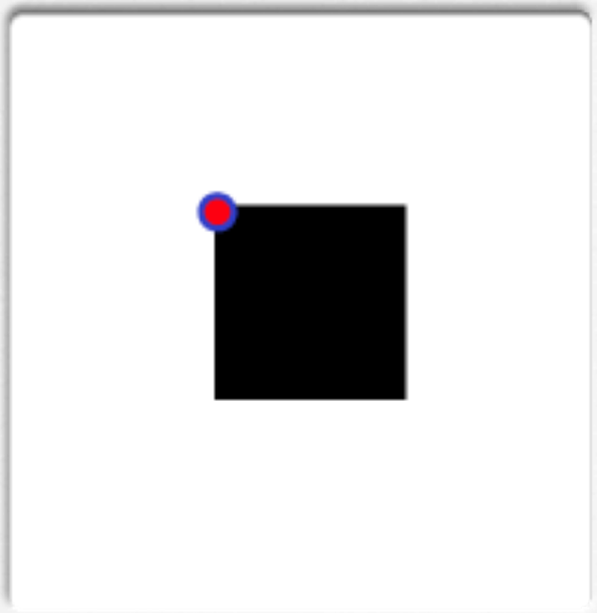
$\left(\frac{\partial f}{\partial y}[m,n]\right)^2$

Source : PM Jodoin

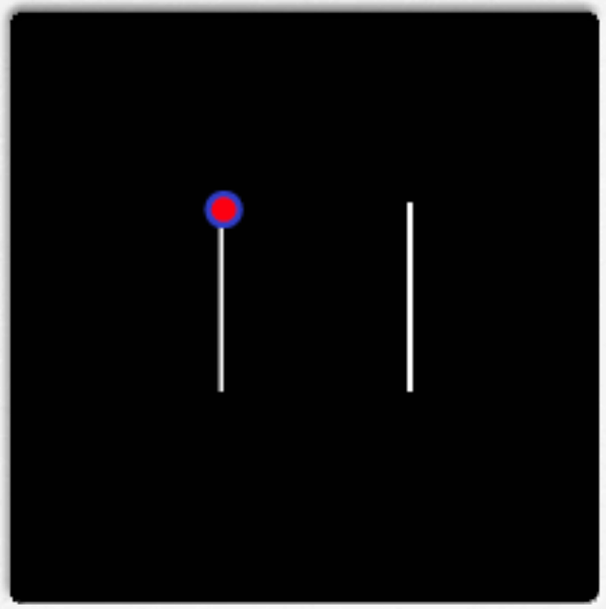
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

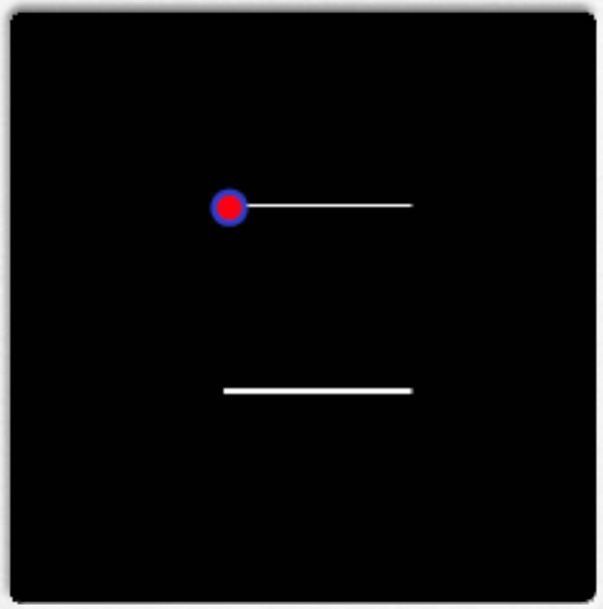
1. Tenseur de structure



$f[m,n]$



$\left(\frac{\partial f}{\partial x}[m,n]\right)^2$



$\left(\frac{\partial f}{\partial y}[m,n]\right)^2$

Remarque :

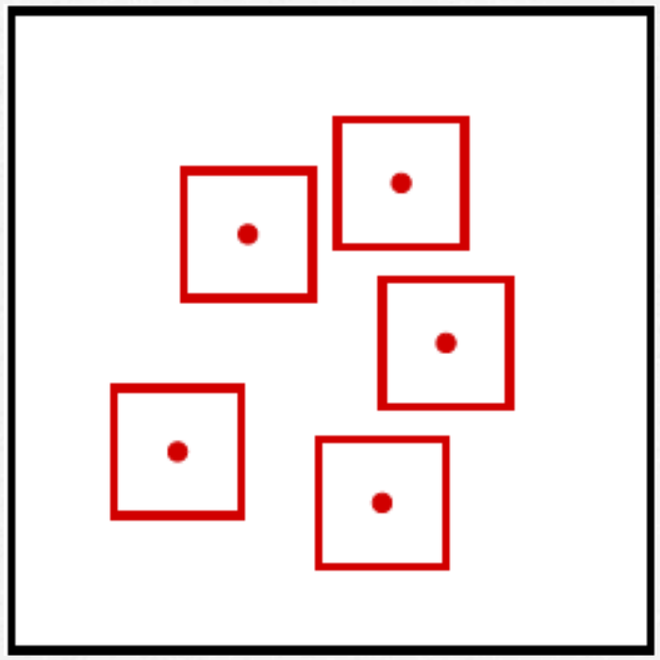
- l'exposant 2 est là juste pour éviter les négatifs

Source : PM Jodoin

3. Caractéristiques ponctuelles

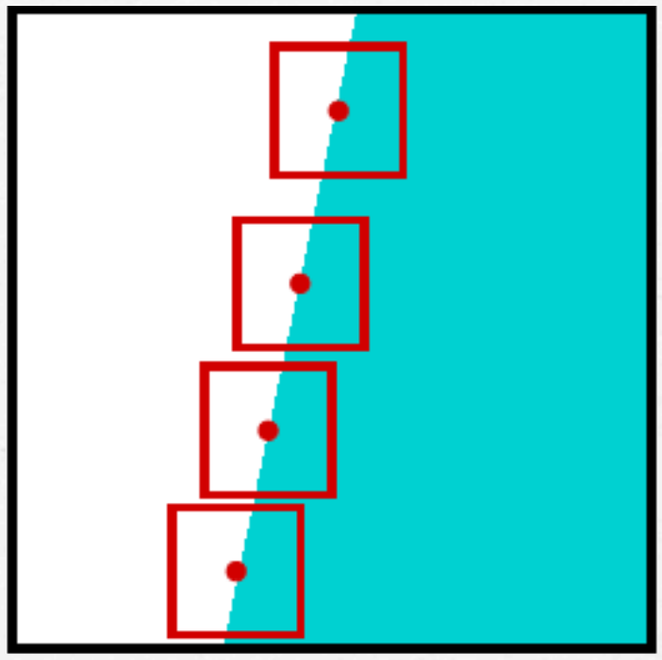
1. Tenseur de structure

Région constante :



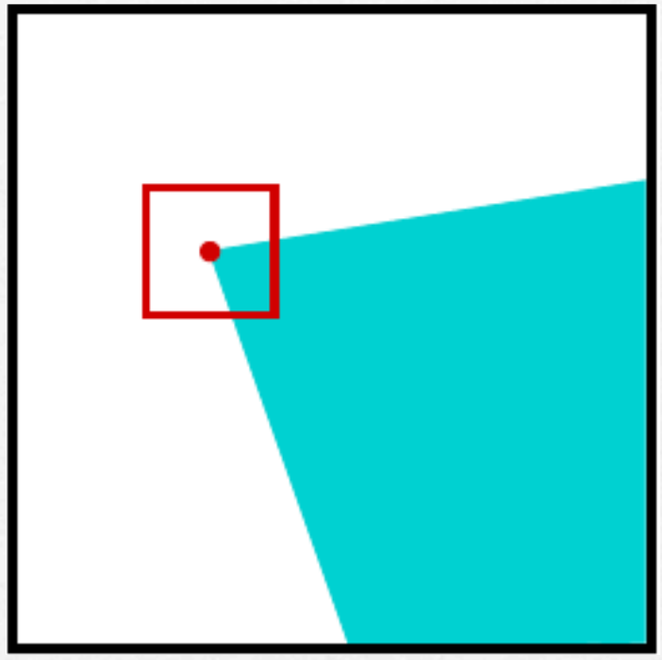
Pas de direction

Contour :



Une direction très forte

Jonction :



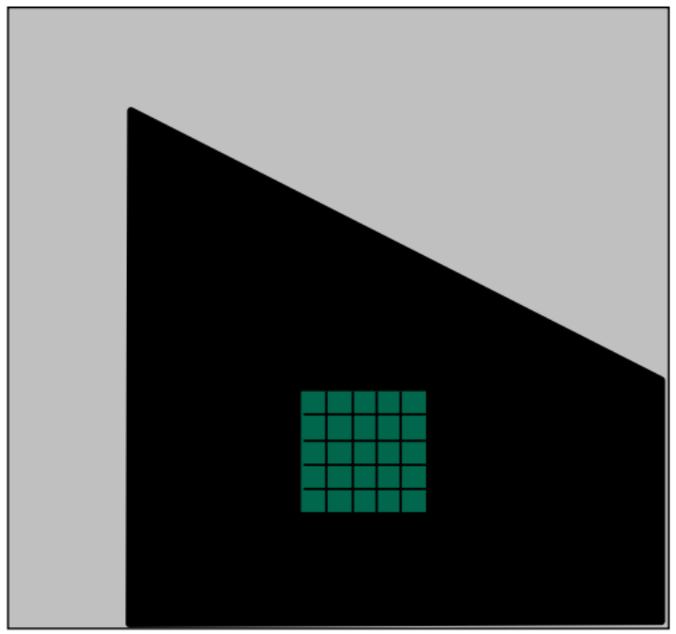
Deux directions très fortes

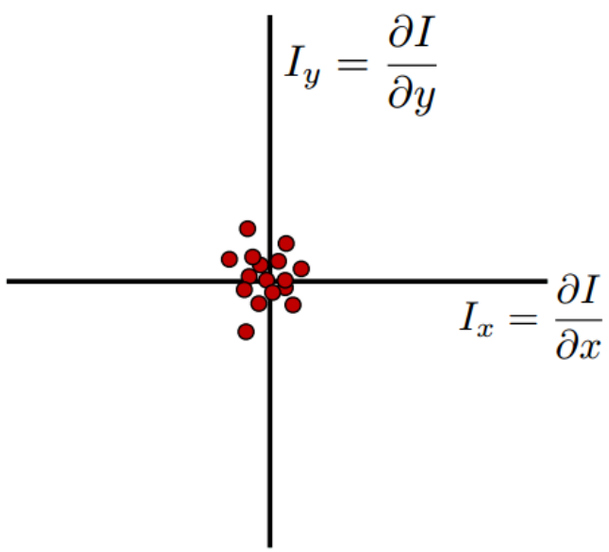
Source : B. Goldlücke “Introduction to Image Processing on the GPU” Short Course, 2012

©Marie-Flavie Auclair-Fortier

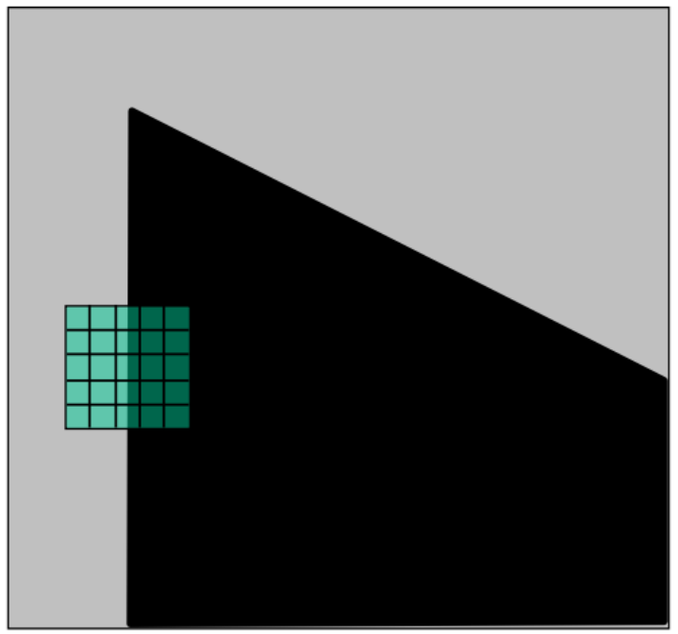
Détecteur de Harris (1989)

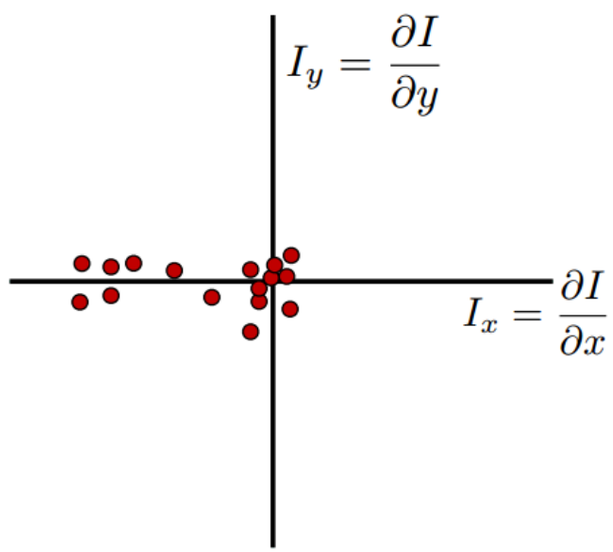
Régions uniformes
2 dérivées faibles



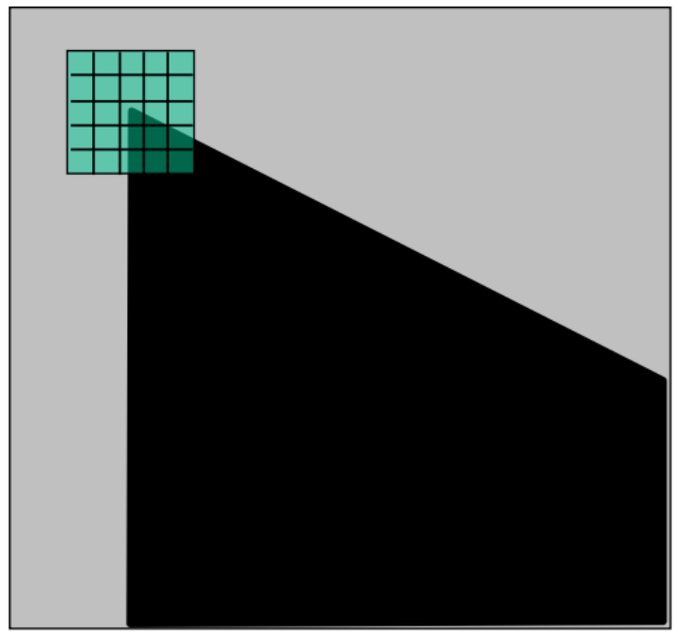


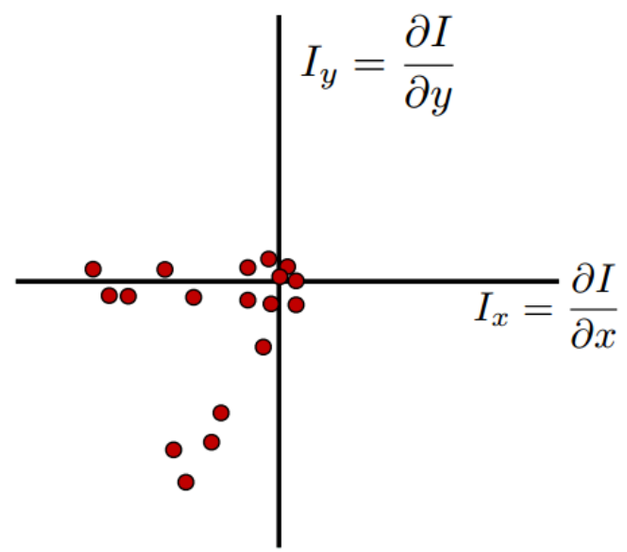
Près d'un contours:
1 dérivée faible et
1 dérivée élevée





Près d'un coin:
2 dérivées élevées





3. Caractéristiques ponctuelles

1. Tenseur de structure

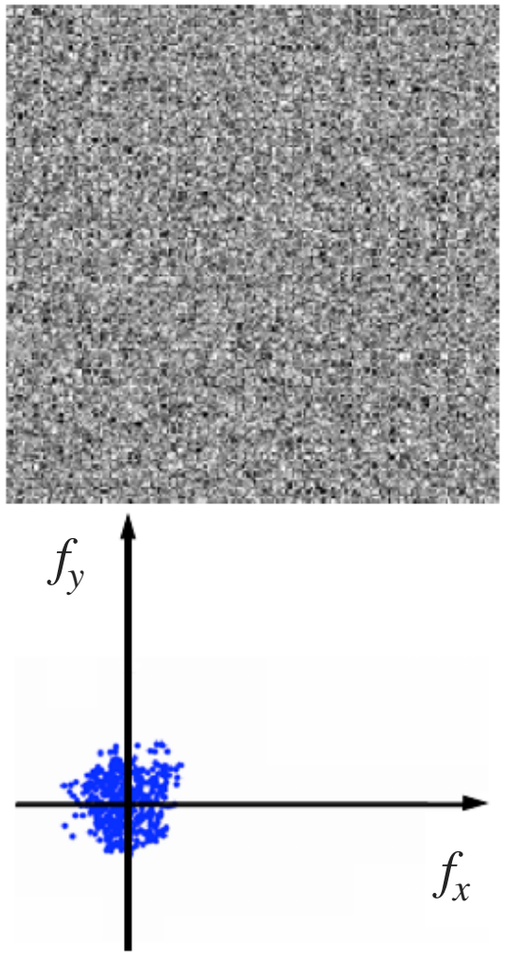
- *Exemples* :
  - ✓ Au voisinage des contours et des jonctions, la **distribution** des dérivées partielles est caractéristique

Source : B. Goldlücke “Introduction to Image Processing on the GPU” Short Course, 2012

3. Caractéristiques ponctuelles

1. Tenseur de structure

- *Exemples* :
  - ✓ Au voisinage des contours et des jonctions, la **distribution** des dérivées partielles est caractéristique

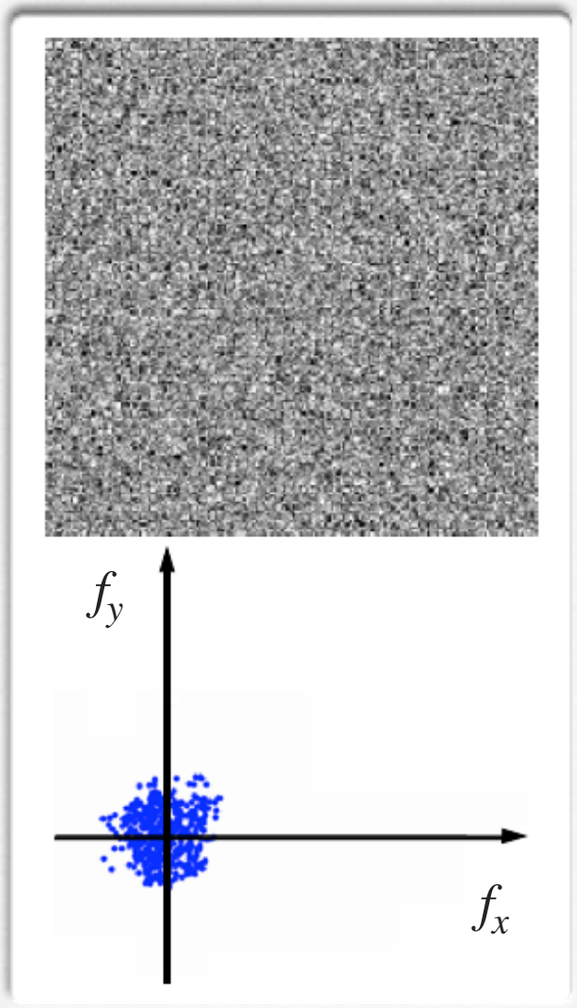


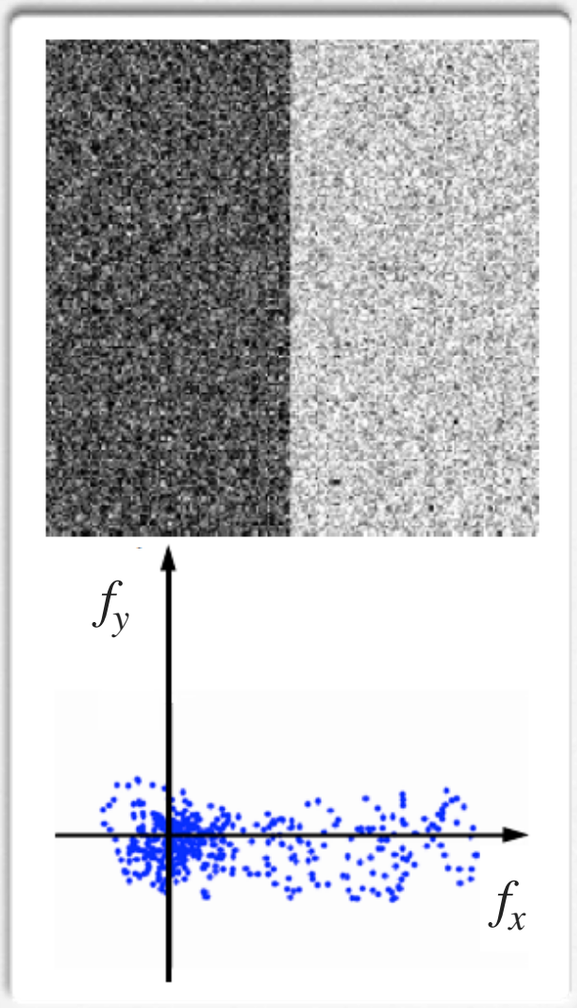
Source : B. Goldlücke “Introduction to Image Processing on the GPU” Short Course, 2012

3. Caractéristiques ponctuelles

1. Tenseur de structure

- *Exemples* :
  - ✓ Au voisinage des contours et des jonctions, la **distribution** des dérivées partielles est caractéristique



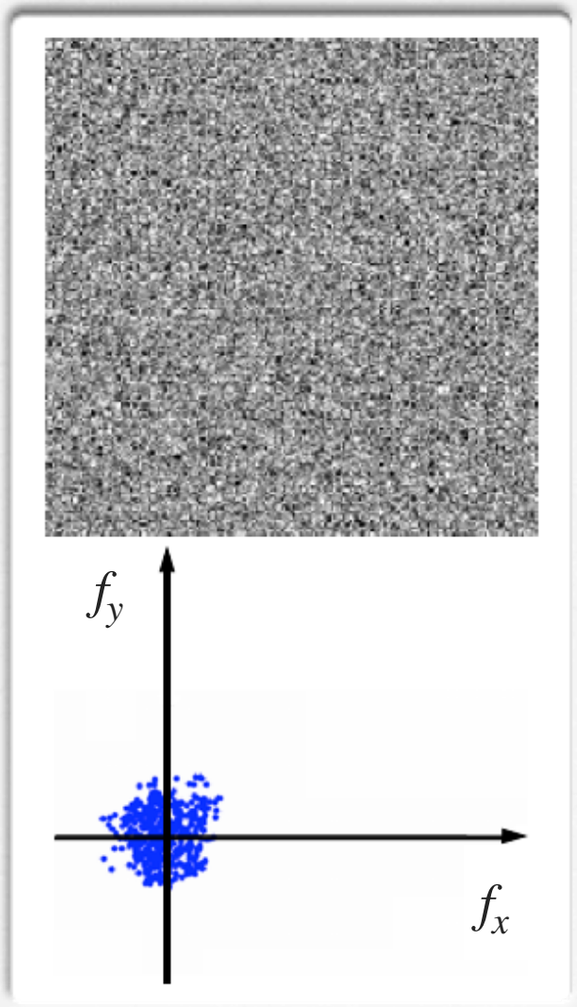


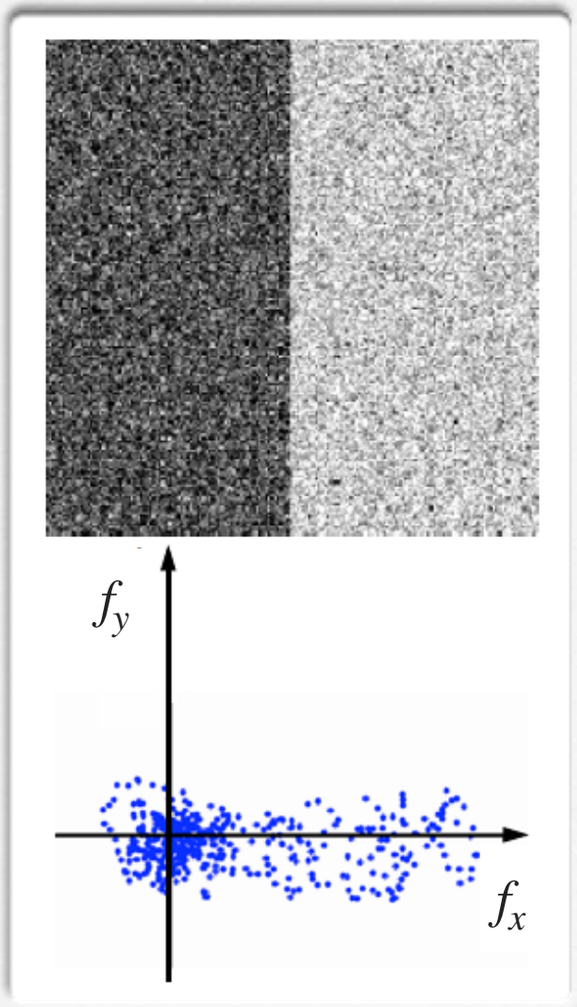
Source : B. Goldlücke “Introduction to Image Processing on the GPU” Short Course, 2012

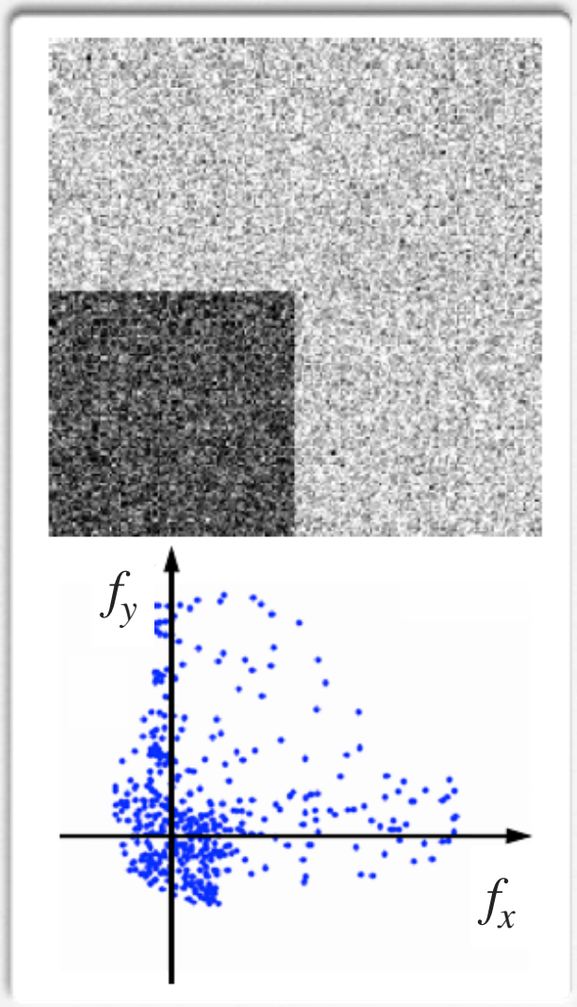
3. Caractéristiques ponctuelles

1. Tenseur de structure

- *Exemples* :
  - ✓ Au voisinage des contours et des jonctions, la **distribution** des dérivées partielles est caractéristique



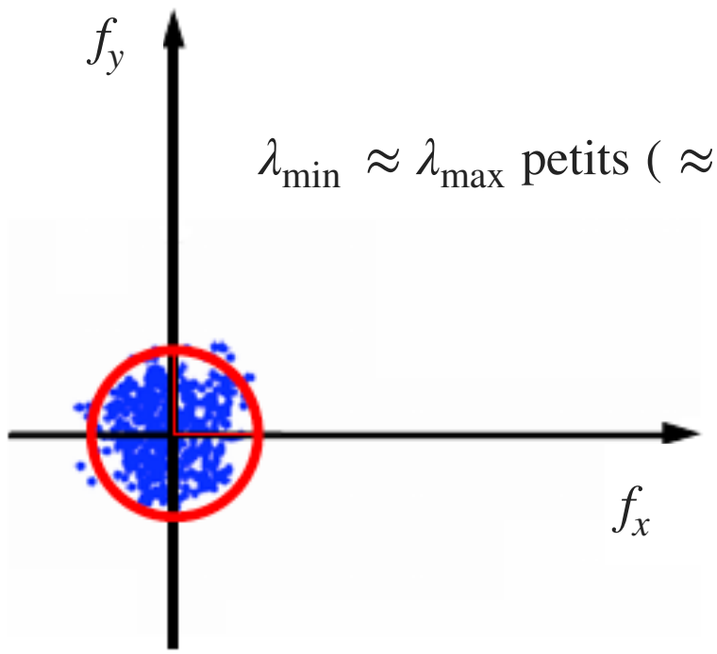




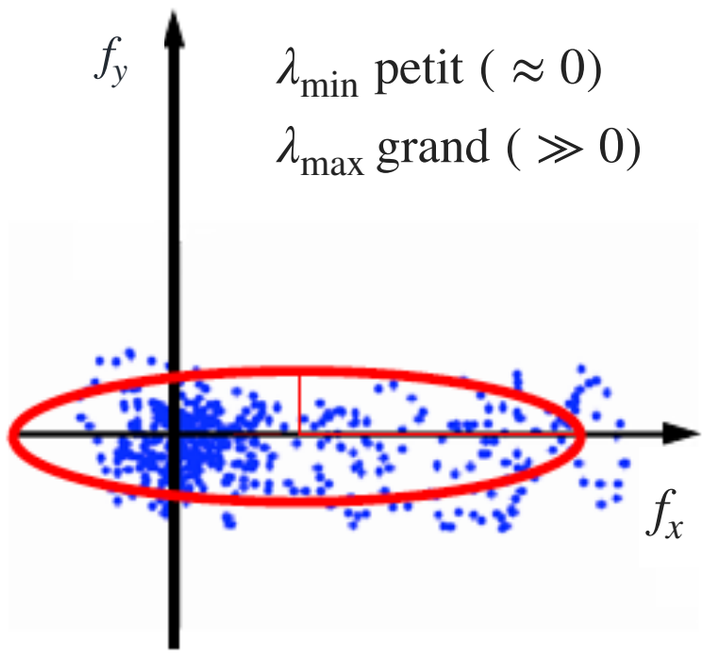
Source : B. Goldlücke "Introduction to Image Processing on the GPU" Short Course, 2012

3. CARACTÉRISTIQUES PONCTUELLES

1. Tenseur de structure

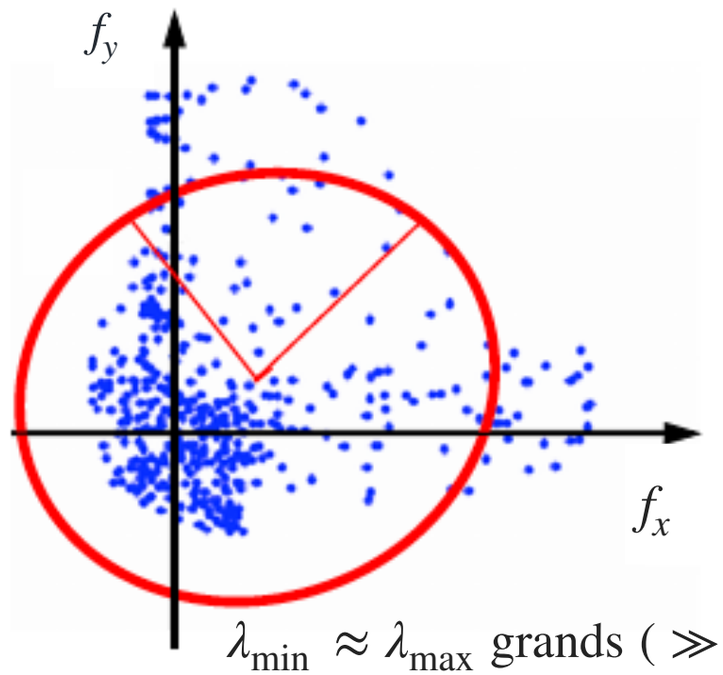


$\lambda_{min} \approx \lambda_{max}$ petits (≈ 0)



λ_{min} petit (≈ 0)

λ_{max} grand ($\gg 0$)

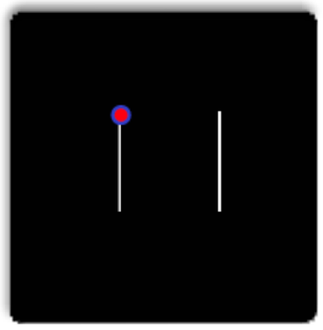


$\lambda_{min} \approx \lambda_{max}$ grands ($\gg 0$)

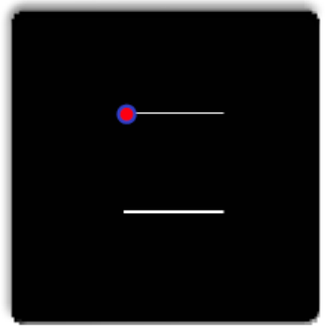
Source : B. Goldlücke “Introduction to Image Processing on the GPU” Short Course, 2012

©Marie-Flavie Auclair-Fortier

La matrice Hessienne



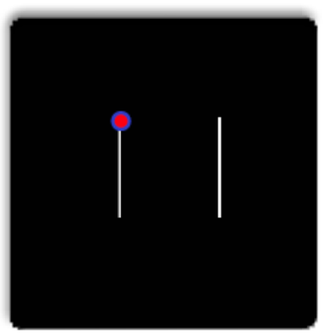
$$\left(\frac{\partial f}{\partial x}[m,n]\right)^2$$



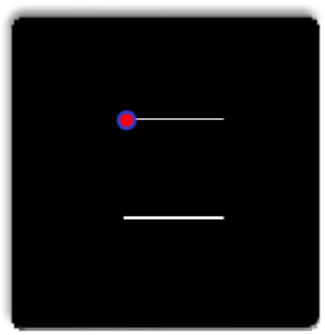
$$\left(\frac{\partial f}{\partial y}[m,n]\right)^2$$

La matrice Hessienne

$$H = \begin{pmatrix} \sum_x \sum_y \left(\frac{\partial f}{\partial x} \right)^2 & \sum_x \sum_y \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \\ \sum_x \sum_y \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} & \sum_x \sum_y \left(\frac{\partial f}{\partial y} \right)^2 \end{pmatrix}$$



$$\left(\frac{\partial f}{\partial x}[m,n] \right)^2$$



$$\left(\frac{\partial f}{\partial y}[m,n] \right)^2$$

3. Caractéristiques ponctuelles

1. Tenseur de structure

- La matrice Hessienne (H)

$$H[m,n] = \left(H * h_b\right)[m,n]$$

  - Ici, h_b est le filtre moyenneur
    - ✓ Chaque élément de S' est une moyenne locale des multiplications de dérivées

$$H[m,n] = \begin{pmatrix} A & B \\ B & C \end{pmatrix} \quad \text{où} \quad A = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \left(f_x[m,n]\right)^2$$

$$B = \sum_{m=-M}^{M} \sum_{n=-N}^{N} f_x[m,n]\, f_y[m,n]$$

$$C = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \left(f_y[m,n]\right)^2$$

Source : PM Jodoin

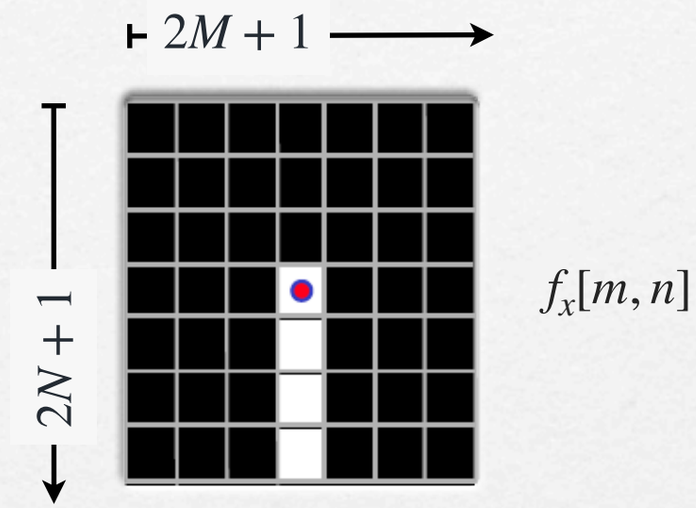
3. Caractéristiques ponctuelles

1. Tenseur de structure

- La matrice Hessienne (H)

$$H[m,n] = \left(H * h_b\right)[m,n]$$

  - Ici, h_b est le filtre moyenneur
    - ✓ Chaque élément de S' est une moyenne locale des multiplications de dérivées



$$H[m,n] = \begin{pmatrix} A & B \\ B & C \end{pmatrix} \quad \text{où} \quad A = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \left(f_x[m,n]\right)^2$$

$$B = \sum_{m=-M}^{M} \sum_{n=-N}^{N} f_x[m,n]\, f_y[m,n]$$

$$C = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \left(f_y[m,n]\right)^2$$

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

1. Tenseur de structure

- La matrice Hessienne (H)

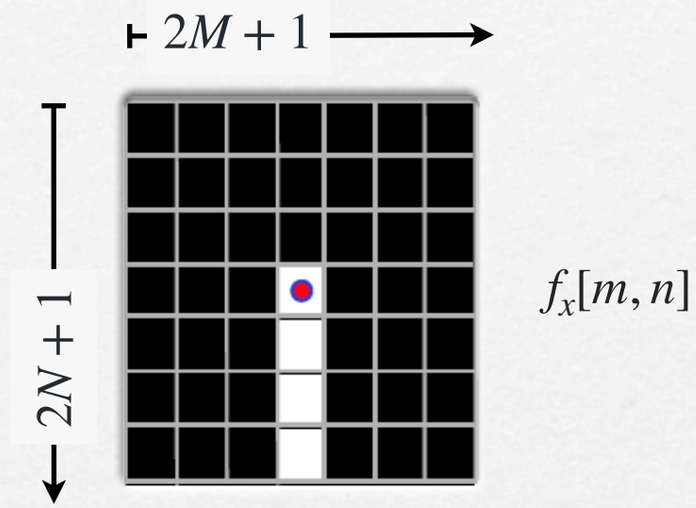
$$H[m,n] = \left(H * h_b\right)[m,n]$$

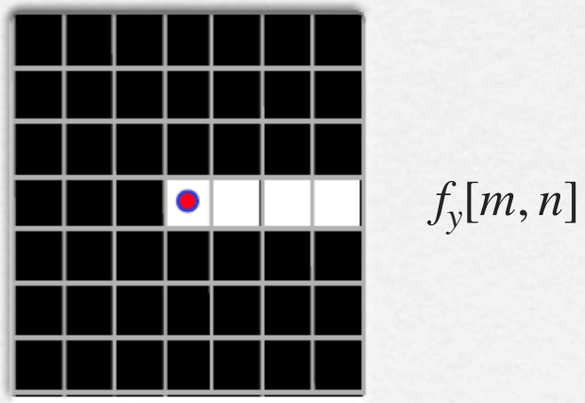
  - Ici, h_b est le filtre moyenneur
    - ✓ Chaque élément de S' est une moyenne locale des multiplications de dérivées

$$H[m,n] = \begin{pmatrix} A & B \\ B & C \end{pmatrix} \quad \text{où} \quad A = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \left(f_x[m,n]\right)^2$$

$$B = \sum_{m=-M}^{M} \sum_{n=-N}^{N} f_x[m,n]\, f_y[m,n]$$

$$C = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \left(f_y[m,n]\right)^2$$





Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

1. Tenseur de structure

- La matrice Hessienne H
  - ✓ Certains auteurs prétendent qu'on peut améliorer les résultats en utilisant un filtre gaussien

$$H[m,n] = \left(H * h_b\right)[m,n]$$

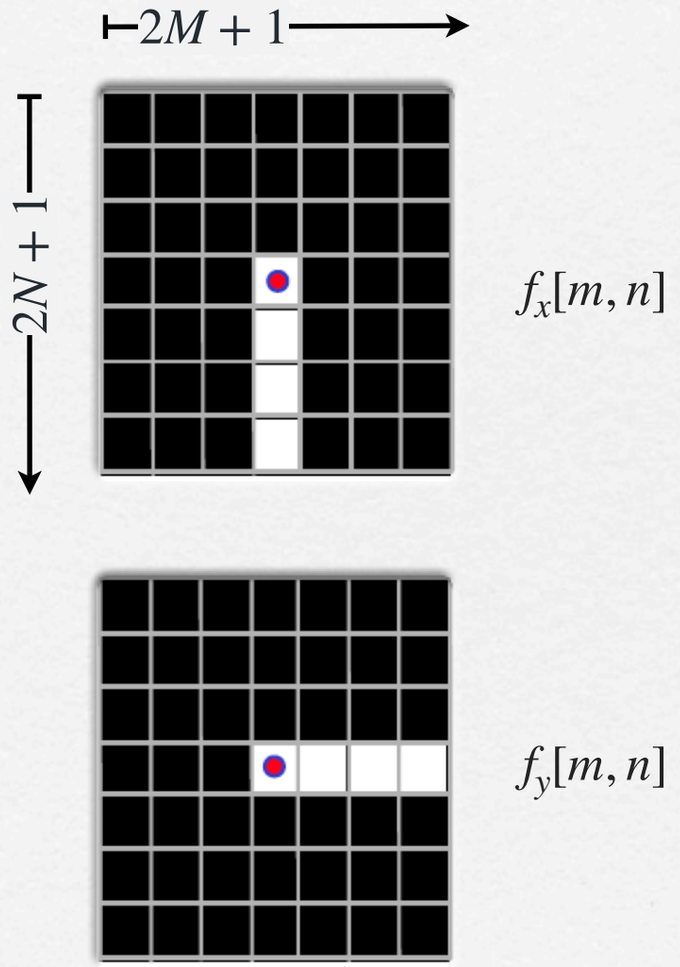
- Ici, h_b est le filtre gaussien :
  - ✓ Chaque élément de H est une moyenne locale pondérée des multiplications de dérivées

$H[m,n]$:

$$H[m,n] = \begin{pmatrix} A & B \\ B & C \end{pmatrix} \quad \text{où} \quad A = \sum_{m=-M}^{M} \sum_{n=-N}^{N} W_{i,j} \left(f_x[m,n]\right)^2$$

$$B = \sum_{m=-M}^{M} \sum_{n=-N}^{N} W_{i,j} f_x[m,n]\, f_y[m,n]$$

$$W_{i,j} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-m)^2 + (j-n)^2}{2\sigma^2}} \qquad C = \sum_{m=-M}^{M} \sum_{n=-N}^{N} W_{i,j} \left(f_y[m,n]\right)^2$$



Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

3. CARACTÉRISTIQUES PONCTUELLES

1. Tenseur de structure

a) Détecteur de Harris

$$M_c(x,y) = det(H) - k.trace(H)^2$$

$f[m,n]$

Source :

©Marie-Flavie Auclair-Fortier

3. CARACTÉRISTIQUES PONCTUELLES

Voisinage gaussien $\sigma = 1.5$
$k = 0.04$

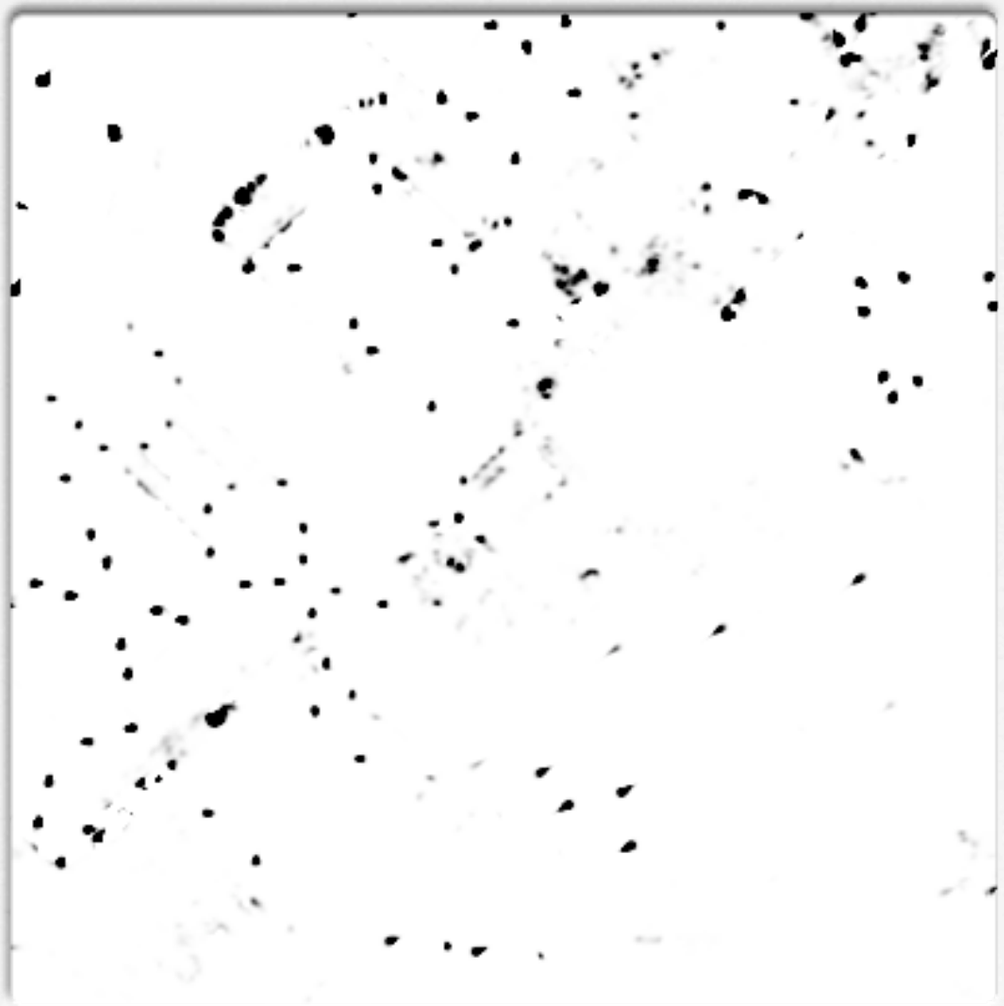
1. Tenseur de structure

a) Détecteur de Harris

$$M_c(x, y) = det(H) - k.trace(H)^2$$

$f[m, n]$

$M_c[m, n]$ (inversé)



Source :

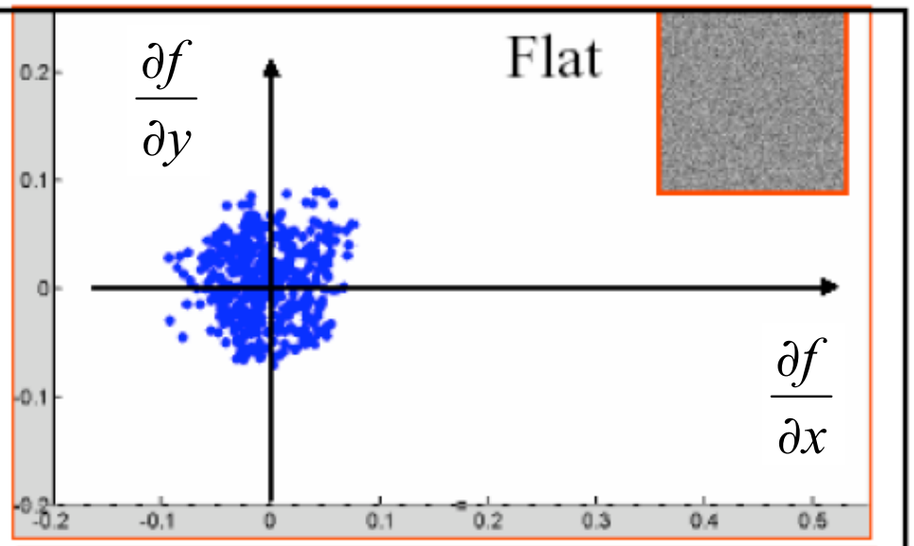
Crédit : Yung-Yu Chuang

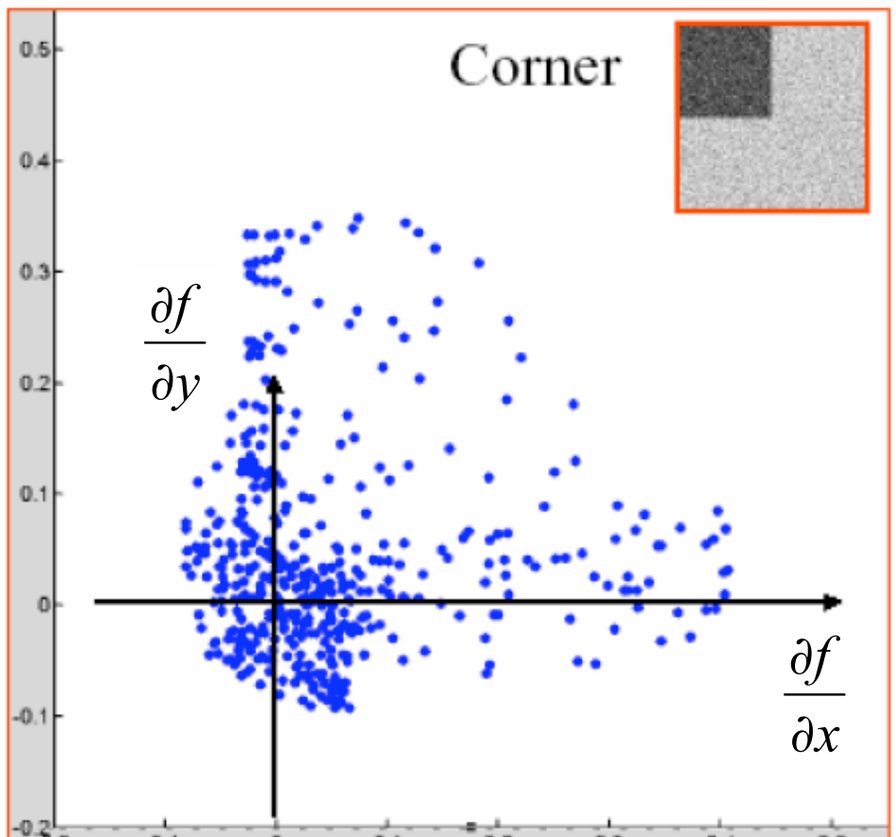
Détecteur de Harris (démo des maths)

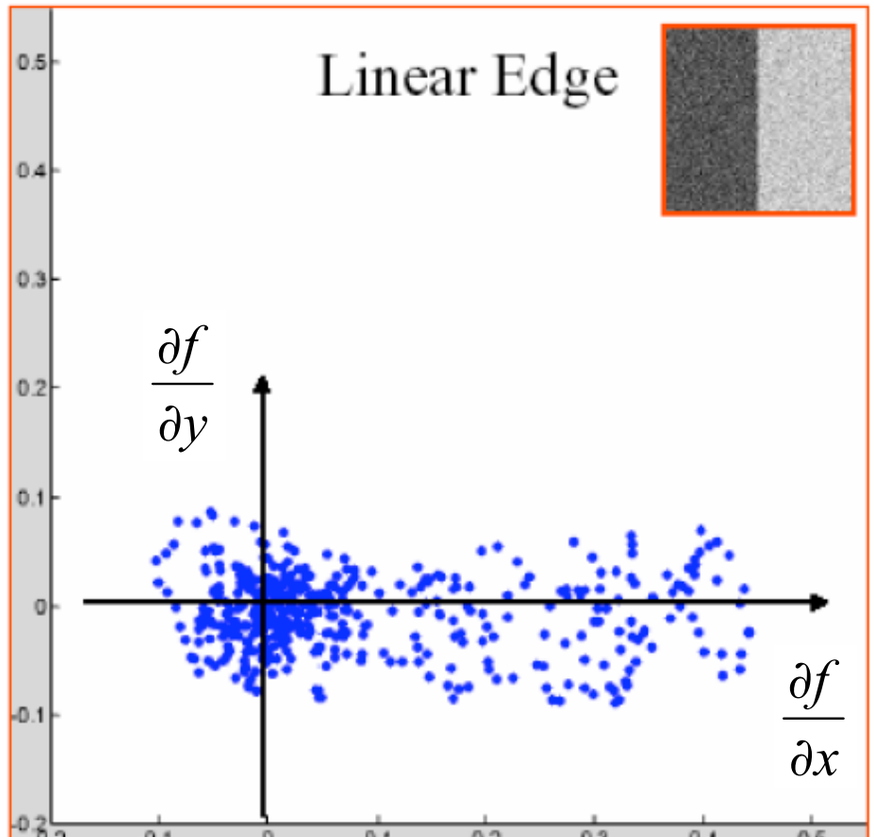
Propriétés de H :

- **vecteurs propres** $\vec{v}_1, \vec{v}_2$: principaux axes de variation
- **valeurs propres** λ_1, λ_2 :amplitude de la variation le long de ces axes

The distribution of the x and y derivatives is very different for all three types of patches







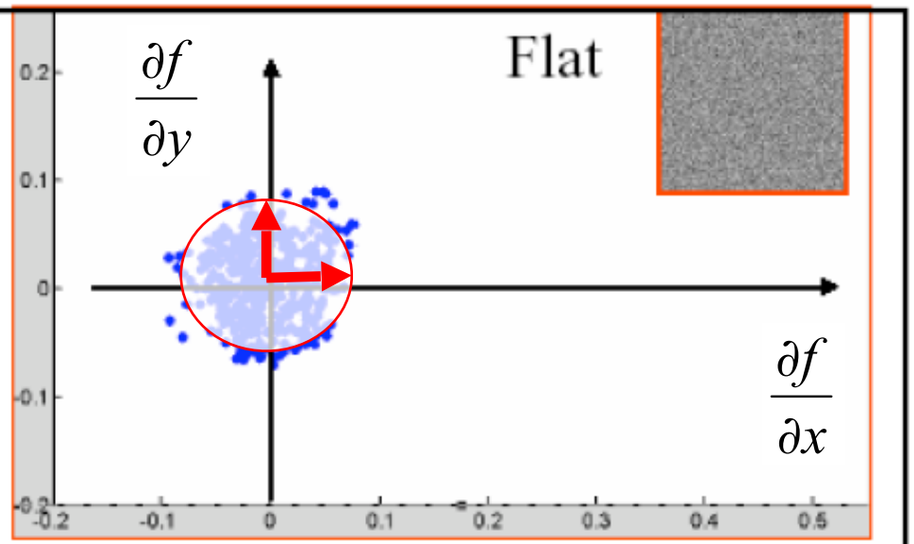
Crédit : Yung-Yu Chuang

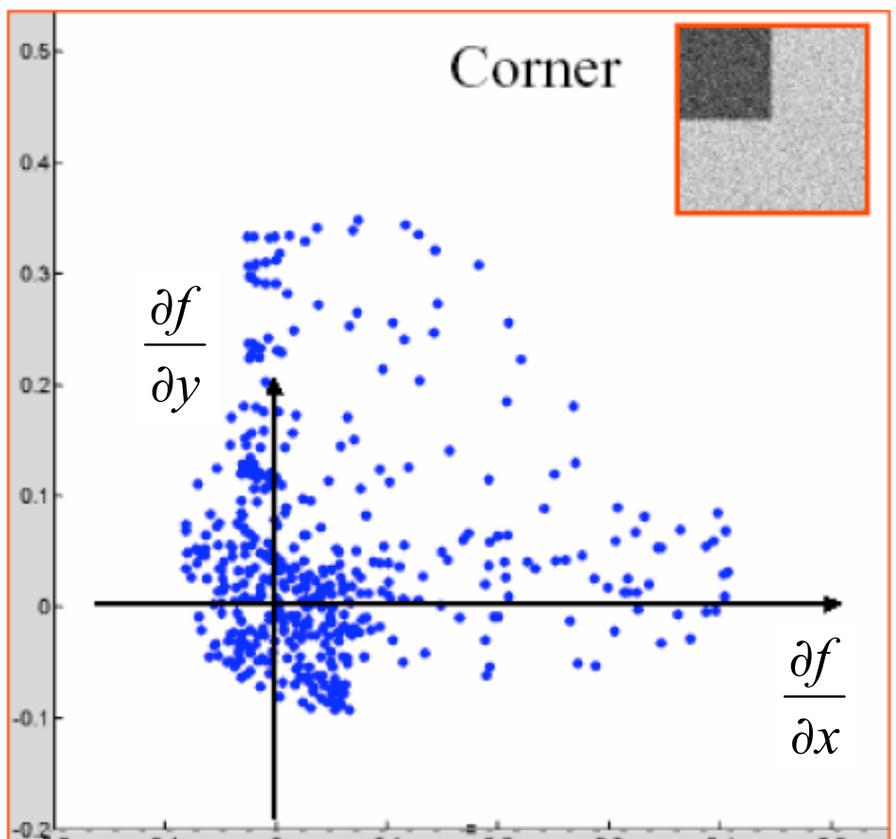
Détecteur de Harris (démo des maths)

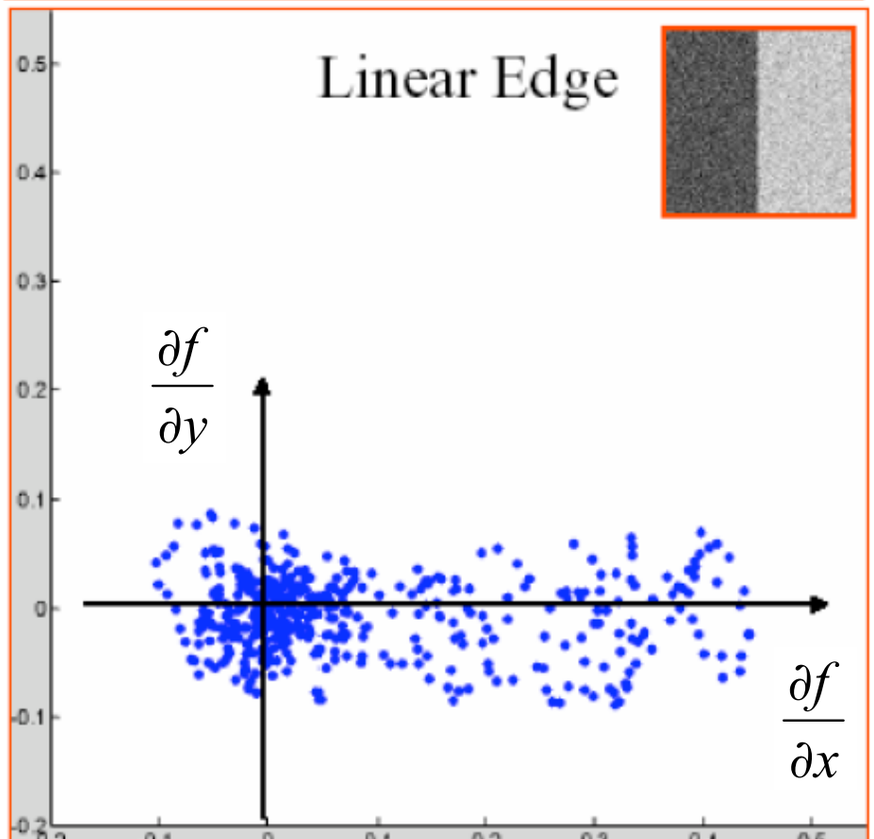
Propriétés de H :

- **vecteurs propres** $\vec{v}_1, \vec{v}_2$: principaux axes de variation
- **valeurs propres** λ_1, λ_2 :amplitude de la variation le long de ces axes

The distribution of the x and y derivatives is very different for all three types of patches





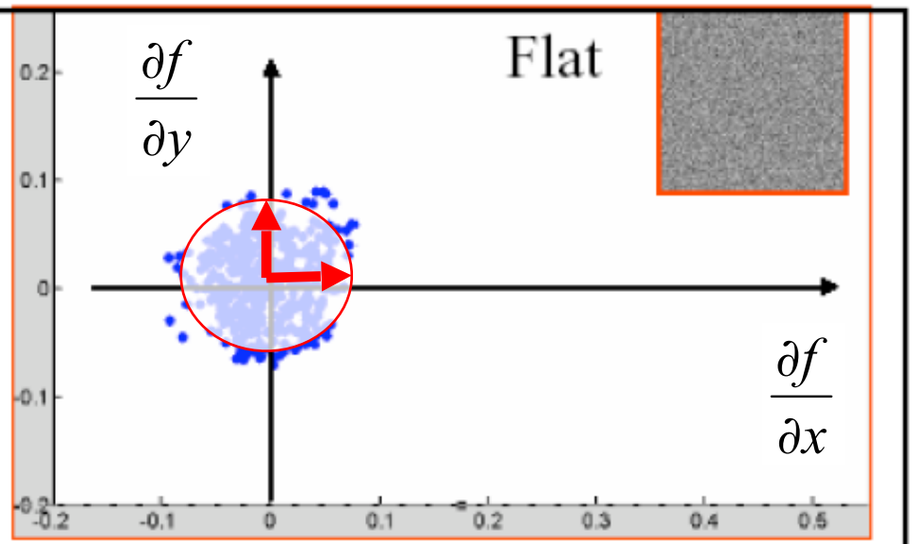


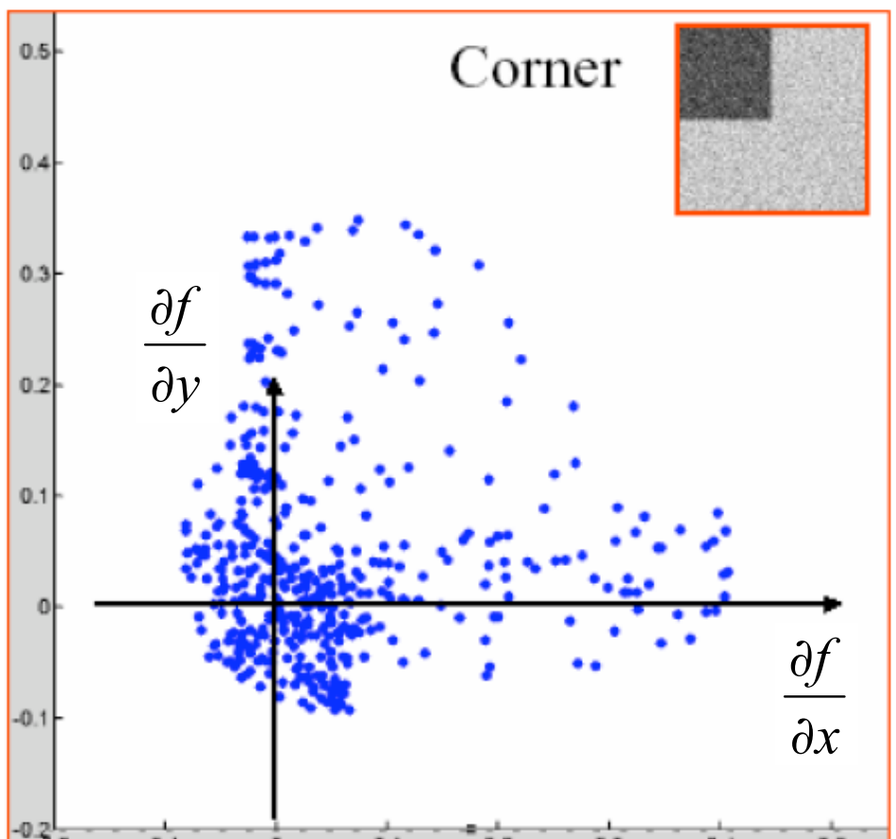
Détecteur de Harris (démo des maths)

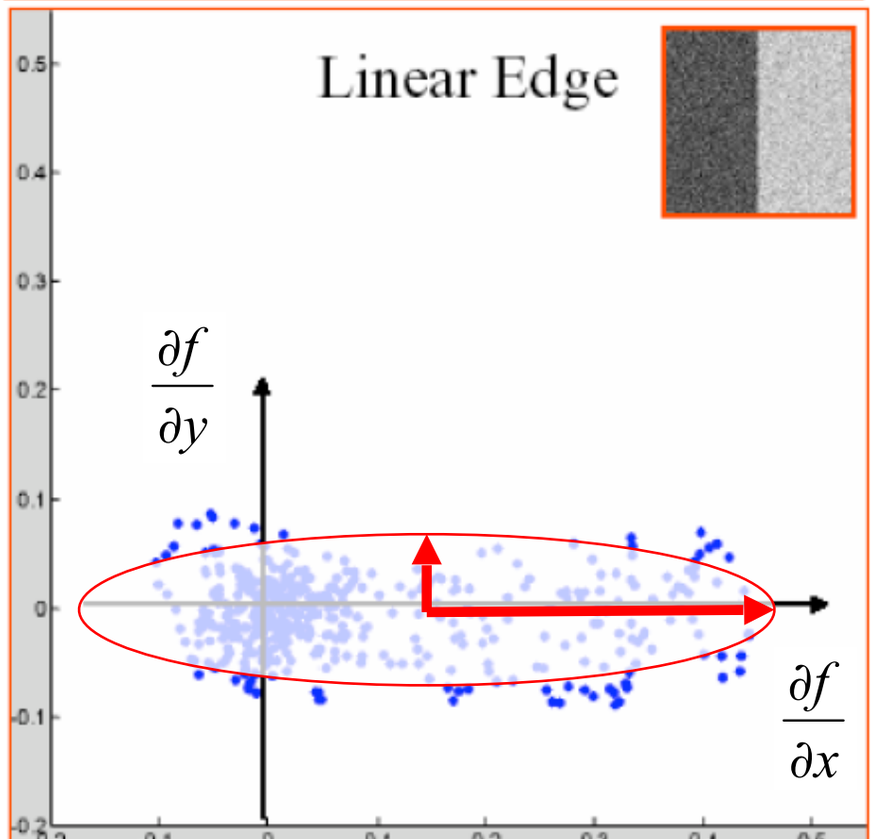
Propriétés de H :

- **vecteurs propres** $\vec{v}_1, \vec{v}_2$: principaux axes de variation
- **valeurs propres** λ_1, λ_2 :amplitude de la variation le long de ces axes

The distribution of the x and y derivatives is very different for all three types of patches





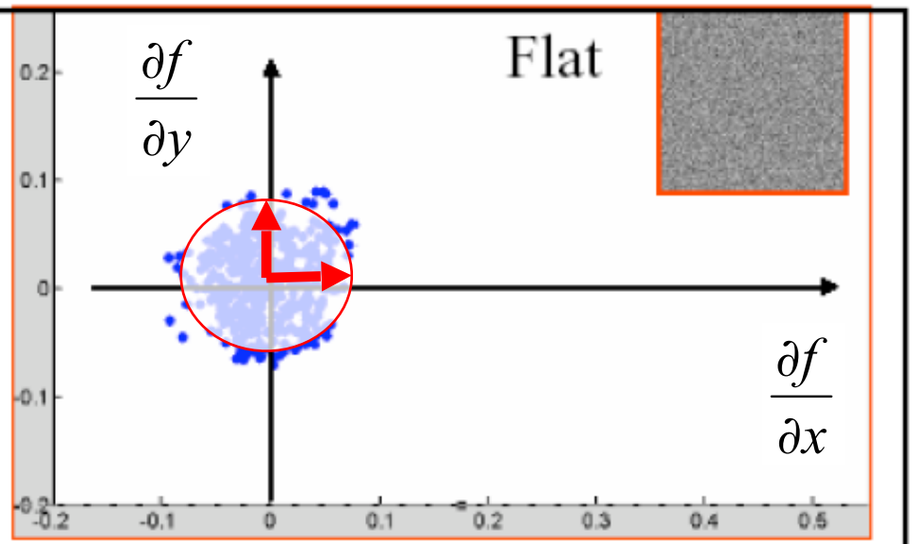


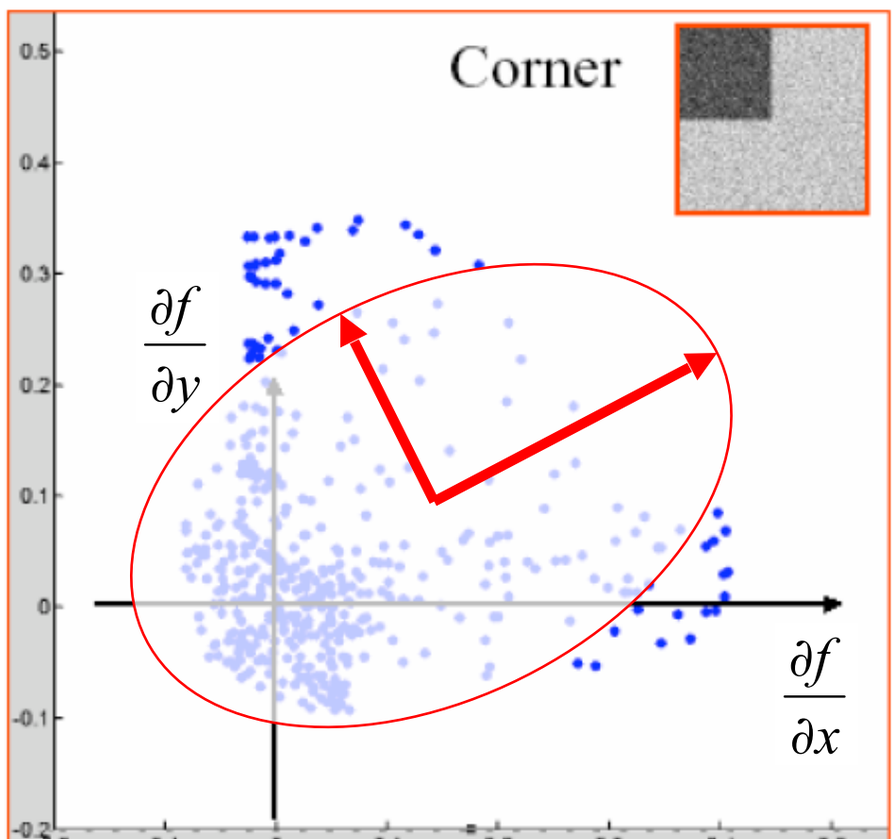
Détecteur de Harris (démo des maths)

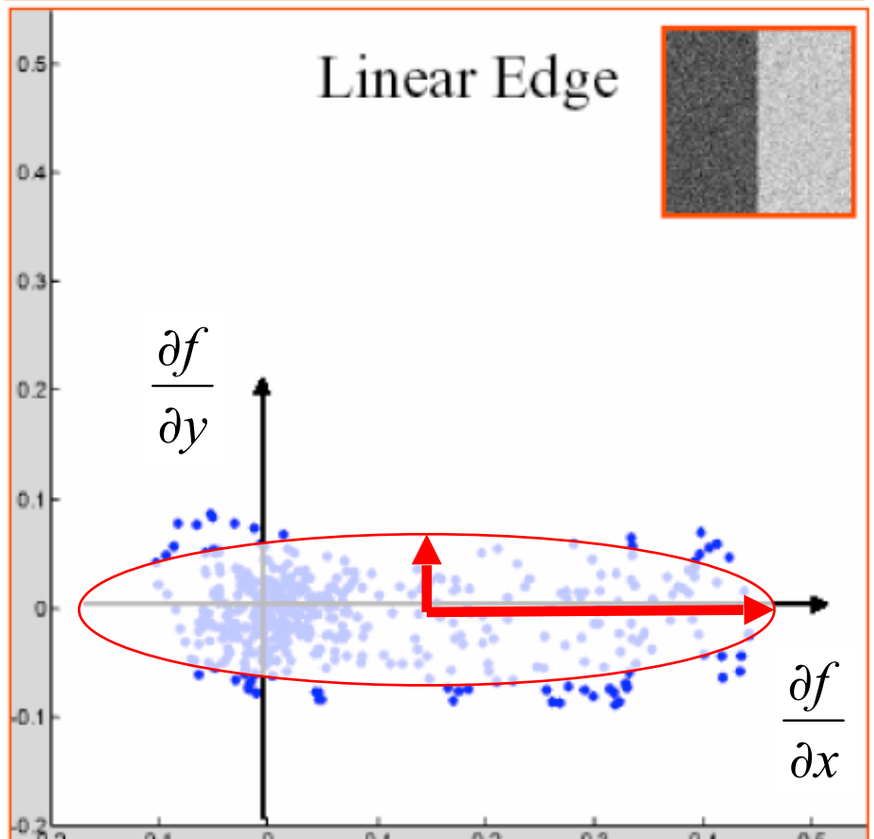
Propriétés de H :

- **vecteurs propres** $\vec{v}_1, \vec{v}_2$: principaux axes de variation
- **valeurs propres** λ_1, λ_2 :amplitude de la variation le long de ces axes

The distribution of the x and y derivatives is very different for all three types of patches





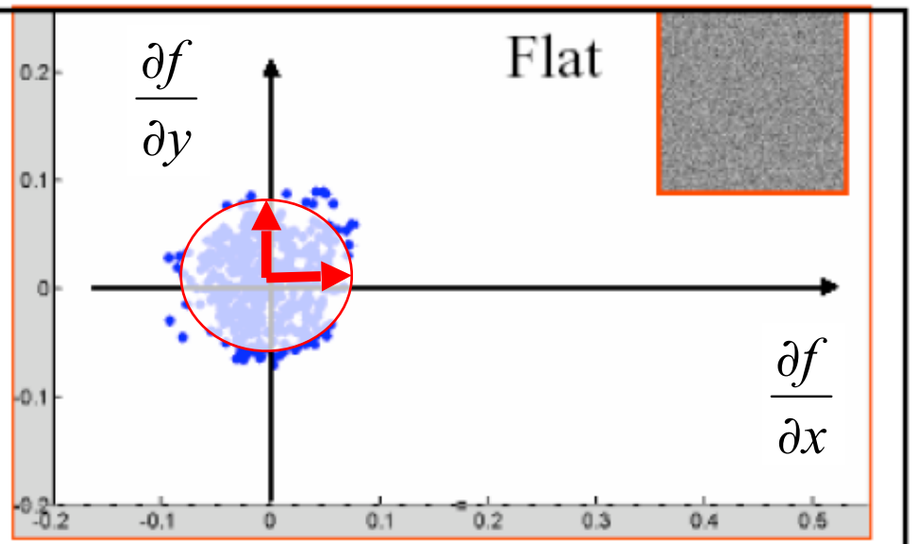


Détecteur de Harris (démo des maths)

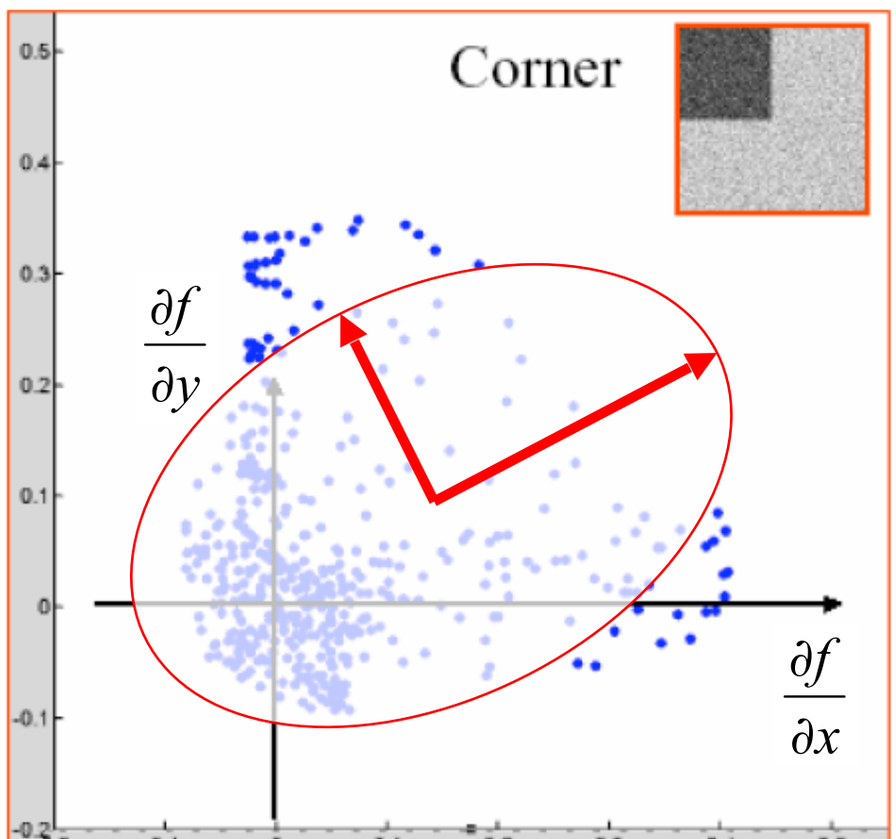
Propriétés de H :

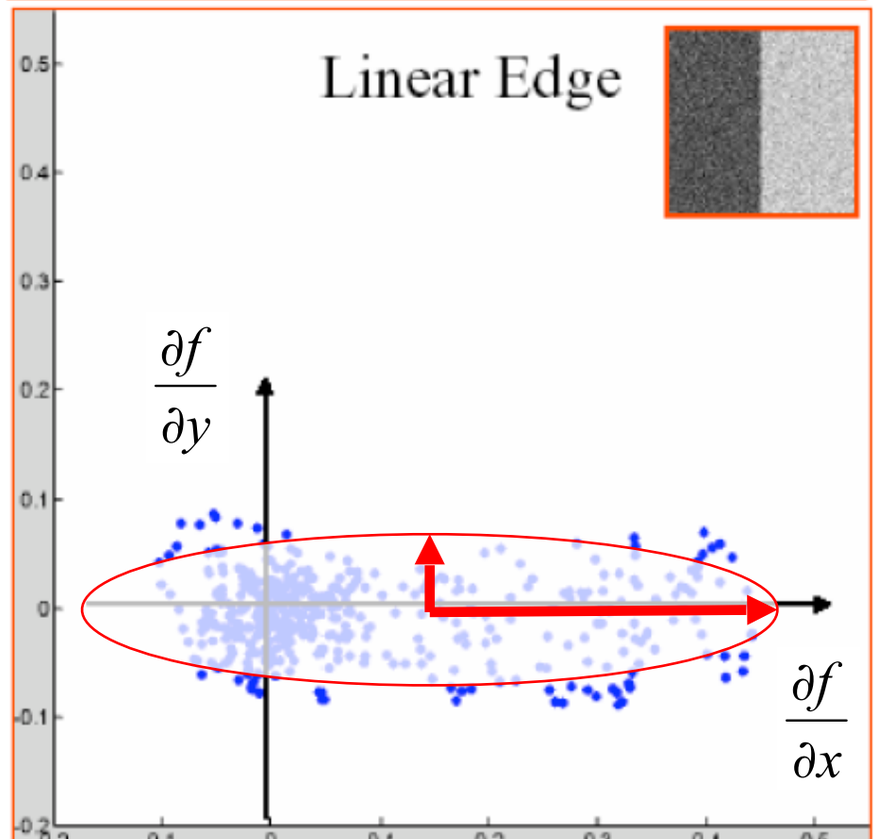
- **vecteurs propres** $\vec{v}_1, \vec{v}_2$: principaux axes de variation
- **valeurs propres** λ_1, λ_2 :amplitude de la variation le long de ces axes

The distribution of the x and y derivatives is very different for all three types of patches



λ_1 et λ_2 faibles





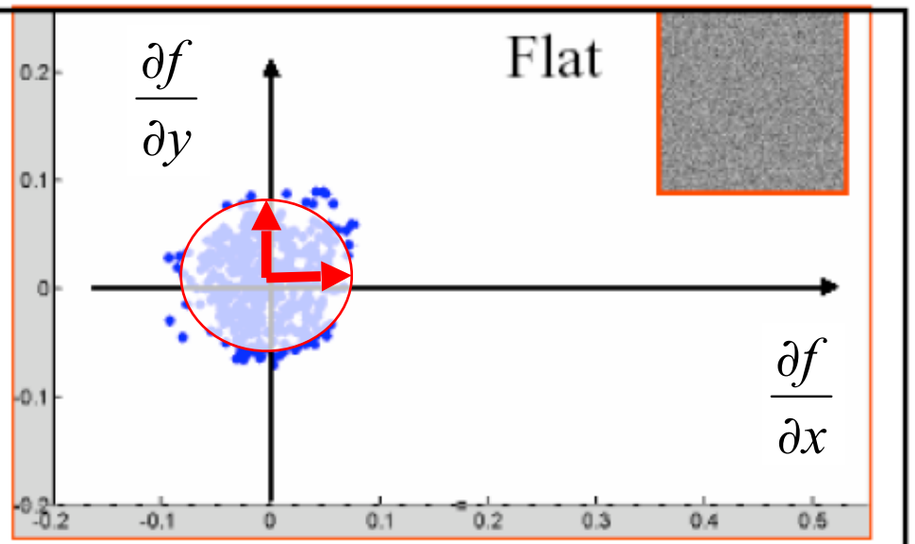
Crédit : Yung-Yu Chuang

Détecteur de Harris (démo des maths)

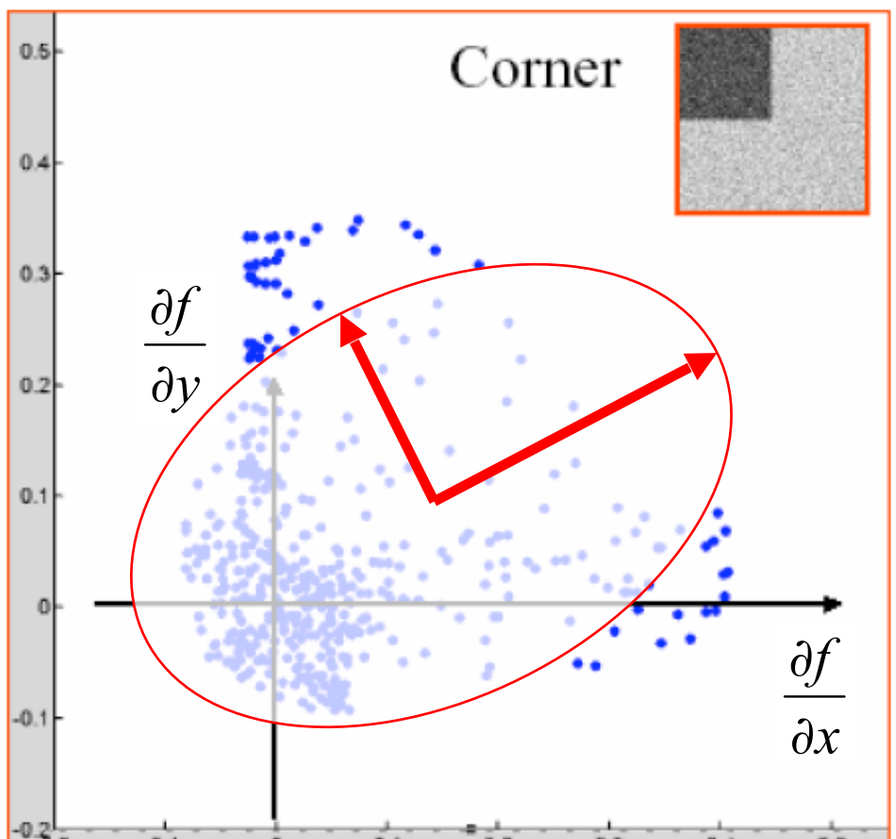
Propriétés de H :

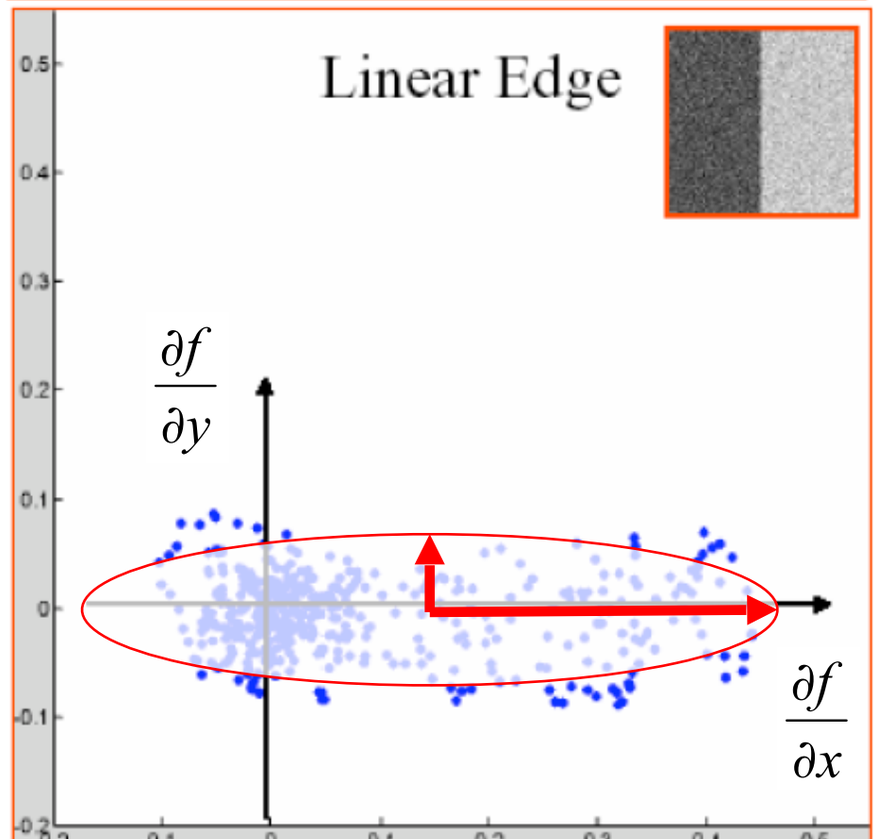
- **vecteurs propres** $\vec{v}_1, \vec{v}_2$: principaux axes de variation
- **valeurs propres** λ_1, λ_2 :amplitude de la variation le long de ces axes

The distribution of the x and y derivatives is very different for all three types of patches



λ_1 et λ_2 faibles





λ_1 élevé

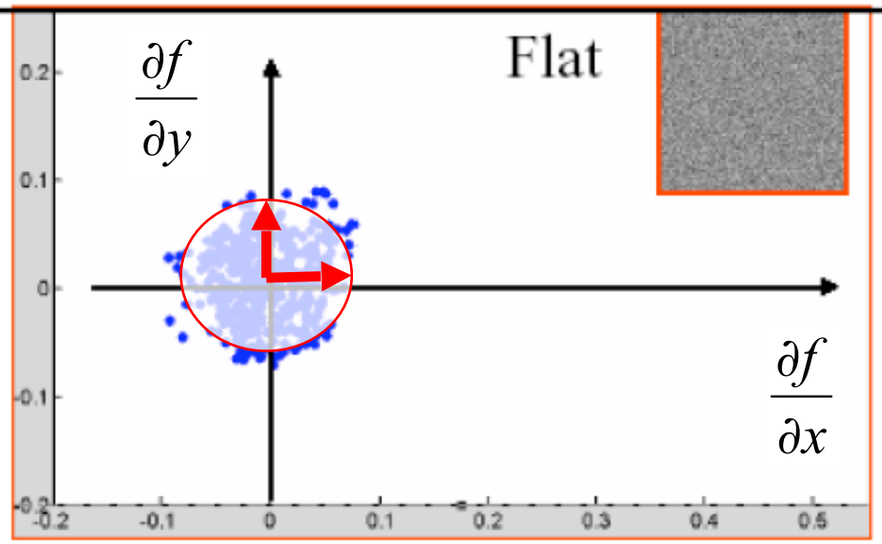
λ_2 faible

Détecteur de Harris (démo des maths)

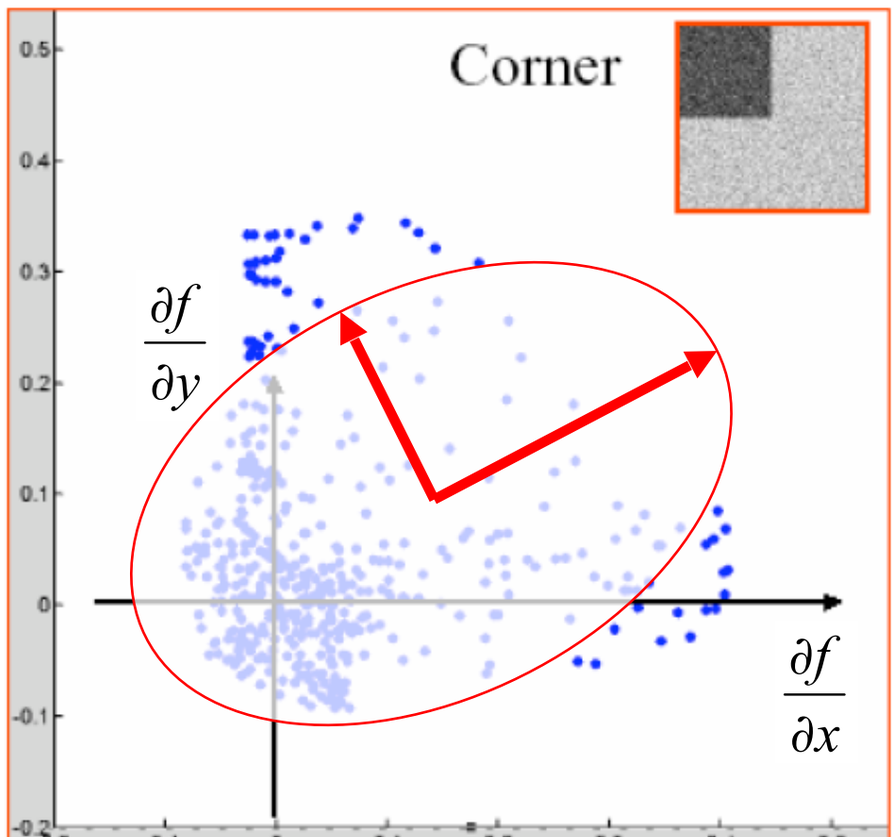
Propriétés de H :

- **vecteurs propres** $\vec{v}_1, \vec{v}_2$: principaux axes de variation
- **valeurs propres** λ_1, λ_2 :amplitude de la variation le long de ces axes

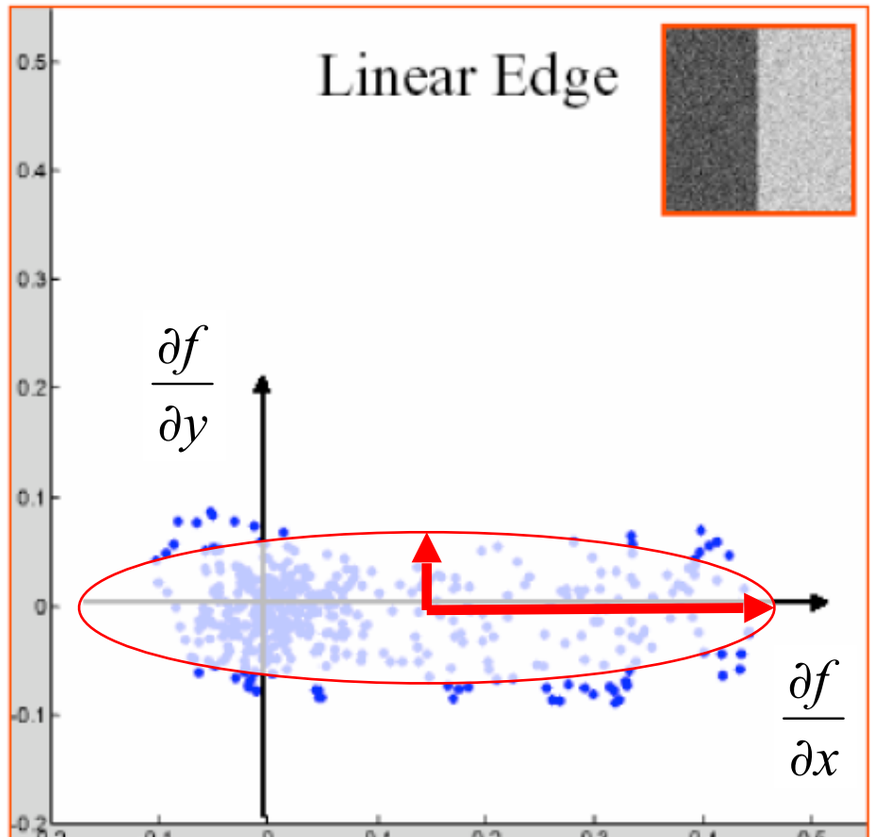
The distribution of the x and y derivatives is very different for all three types of patches



$\lambda_1$et λ_2 faibles



$\lambda_1$et λ_2 élevés



$\lambda_1$élevé
$\lambda_2$faible

Détecteur de Harris

Rappel: comment extraire les **valeurs propres** λ_1 et $\lambda_2$de la matrice *H:*

Détecteur de Harris

Rappel: comment extraire les **valeurs propres** λ_1 et $\lambda_2$de la matrice *H:*

$$\det(H - \lambda I) = 0$$

$$\det\left(\begin{pmatrix} A & B \\ B & C \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) = 0$$

$$\det\begin{pmatrix} A-\lambda & B \\ B & C-\lambda \end{pmatrix} = 0$$

$$(A-\lambda)(C-\lambda) - B^2 = 0 \qquad \text{(Polynôme caractéristique)}$$

Détecteur de Harris

Rappel: comment extraire les **valeurs propres** λ_1 et $\lambda_2$de la matrice *H:*

$$\det(H - \lambda I) = 0$$

$$\det\left(\begin{pmatrix} A & B \\ B & C \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) = 0$$

$$\det\begin{pmatrix} A-\lambda & B \\ B & C-\lambda \end{pmatrix} = 0$$

$$(A-\lambda)(C-\lambda) - B^2 = 0 \qquad \text{(Polynôme caractéristique)}$$

λ_1 et λ_2 Sont les racines du polynôme caractéristique

Détecteur de Harris

Rappel: comment extraire les **valeurs propres** λ_1 et $\lambda_2$de la matrice *H:*

$$\det(H - \lambda I) = 0$$

$$\det\left(\begin{pmatrix} A & B \\ B & C \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) = 0$$

$$\det\begin{pmatrix} A-\lambda & B \\ B & C-\lambda \end{pmatrix} = 0$$

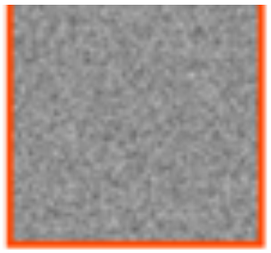
$$(A-\lambda)(C-\lambda) - B^2 = 0 \qquad \text{(Polynôme caractéristique)}$$

λ_1 et λ_2 Sont les racines du polynôme caractéristique

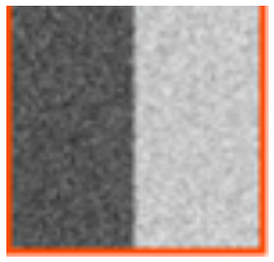
$$\lambda = \frac{A + C \pm \sqrt{(A+C)^2 + 4B^2}}{2}$$

Détecteur de Harris

$$\lambda_1 = \frac{A + C + \sqrt{(A+C)^2 + 4B^2}}{2}, \lambda_2 = \frac{A + C - \sqrt{(A+C)^2 + 4B^2}}{2}$$



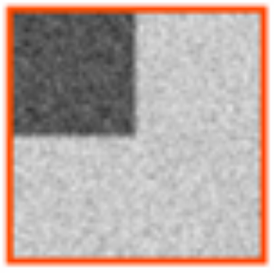
A,B et C sont faibles alors $\lambda_1, \lambda_2 \approx 0$



A est élevé, B et C sont faibles alors

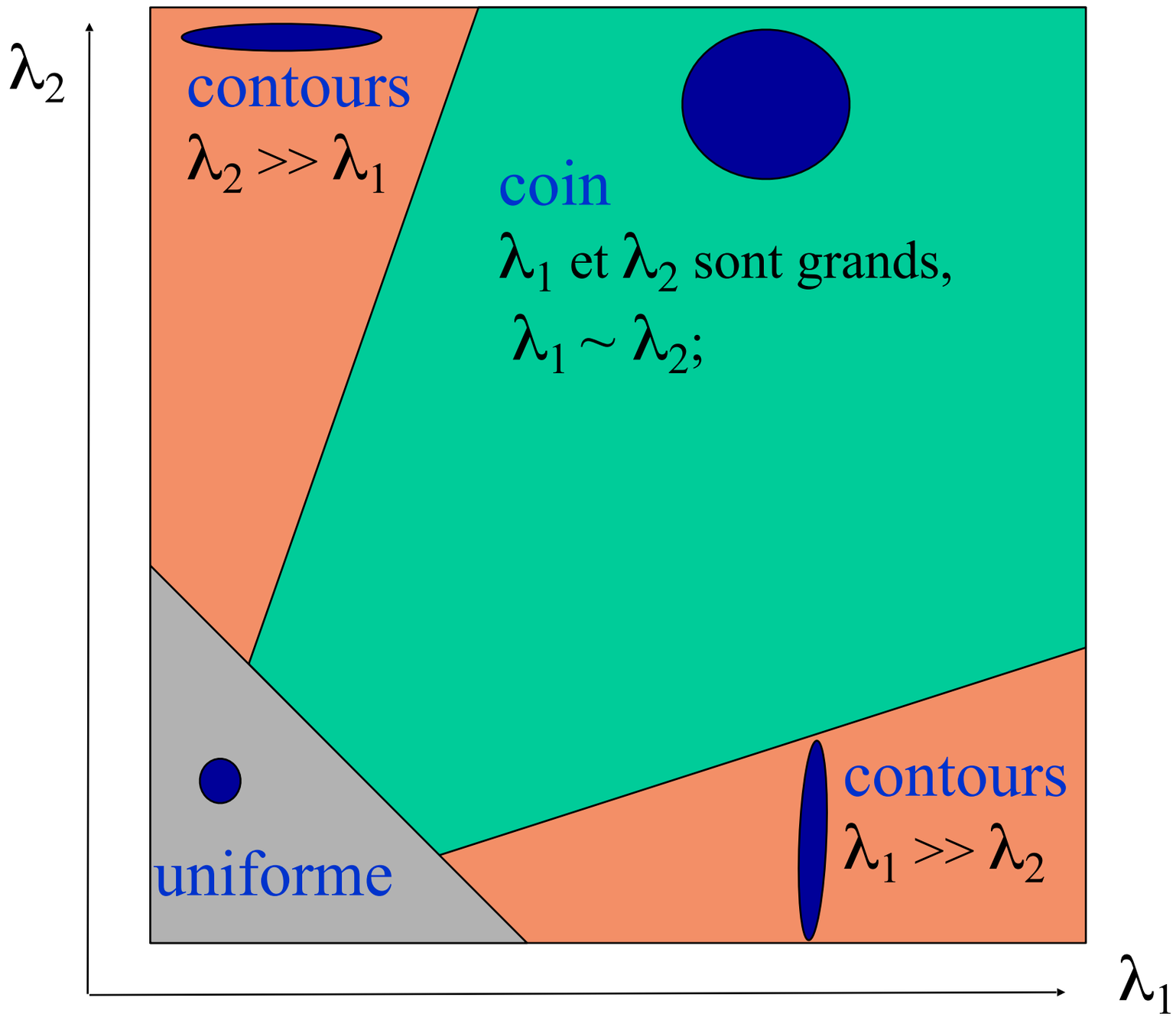
$$\lambda_1 \approx \frac{A + \sqrt{A^2}}{2} = A$$

$$\lambda_2 \approx \frac{A - \sqrt{A^2}}{2} = 0$$



A,B et C sont élevés alors $\lambda_1, \lambda_2 >> 0$

Détecteur de Harris



Détecteur de Harris

Une fois λ_1 et λ_2 calculées on peut en conclure que

Si $\lambda_1 \approx 0$ et $\lambda_2 \approx 0$ alors le pixel est dans une région uniforme

Si $\lambda_1 >> 0$ et $\lambda_2 \approx 0$ alors le pixel est proche d'un contour ou la région est fortement texturée

Si $\lambda_1 >> 0$ et $\lambda_2 >> 0$ alors le pixel est proche d'un coin

Détecteur de Harris

Une fois λ_1 et λ_2 calculées on peut en conclure que

Si $\lambda_1 \approx 0$ et $\lambda_2 \approx 0$ alors le pixel est dans une région uniforme

Si $\lambda_1 >> 0$ et $\lambda_2 \approx 0$ alors le pixel est proche d'un contour ou la région est fortement texturée

Si $\lambda_1 >> 0$ et $\lambda_2 >> 0$ alors le pixel est proche d'un coin

Pour simplifier les calculs (et ainsi éviter de calculer les racines du polynôme caractéristique) Harris suggère de calculer le terme « M_c » à chaque pixel

$$M_c = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$$
$$= \det(H) - \kappa \mathrm{trace}^2(H)$$

k prend généralement une valeur entre 0.02 et 0.1.

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

$f[m,n] + M_c[m,n]$

Source :

©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

1. Tenseur de structure

$M_c > 0.005$

a) Détecteur de Harris

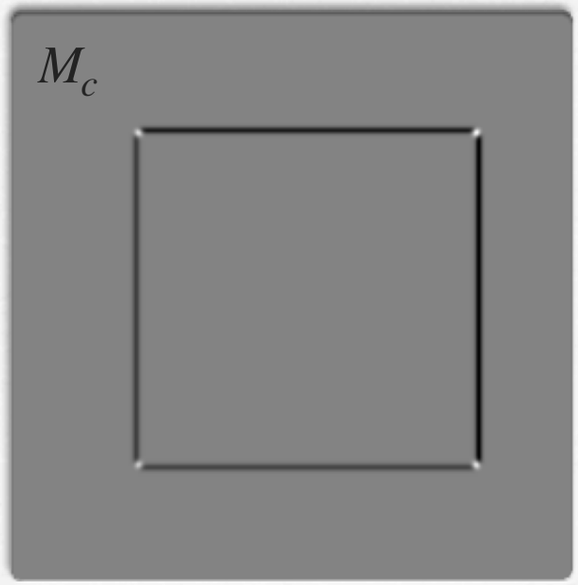
Source :

©Marie-Flavie Auclair-Fortier

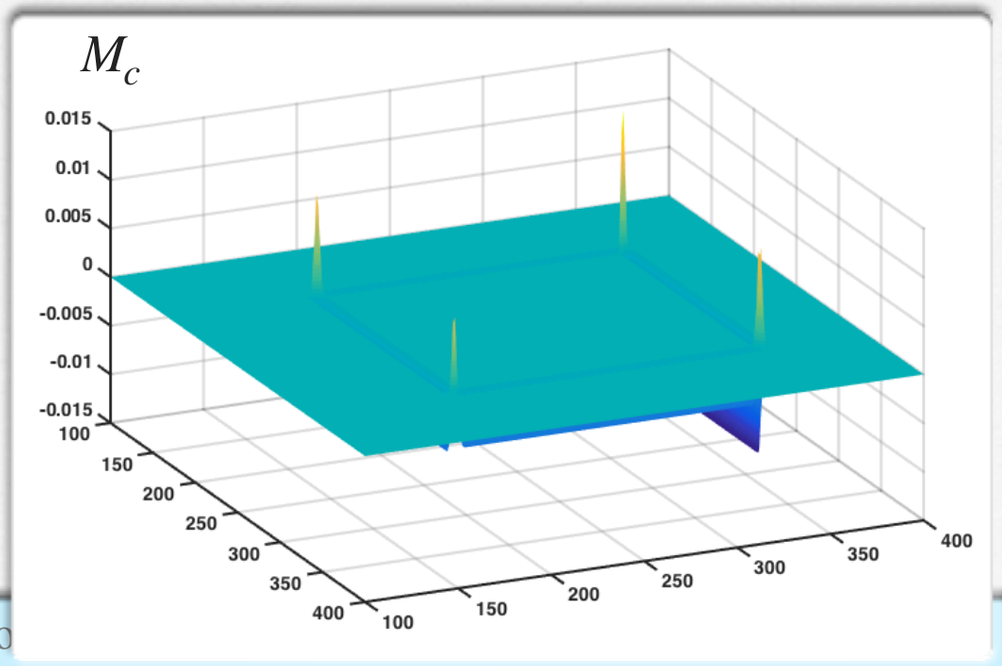
3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris



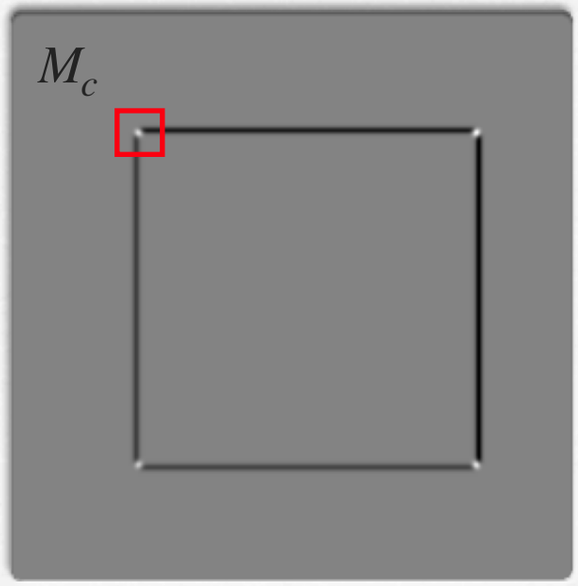
$M_c > 0.005$



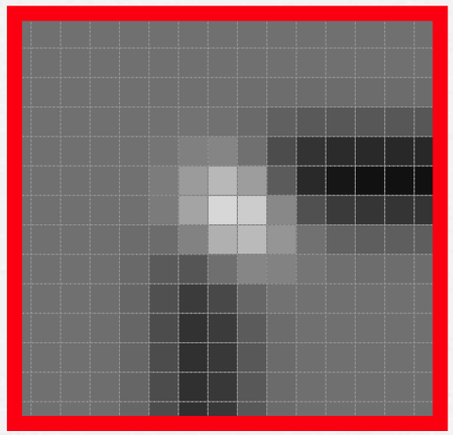
3. Caractéristiques ponctuelles

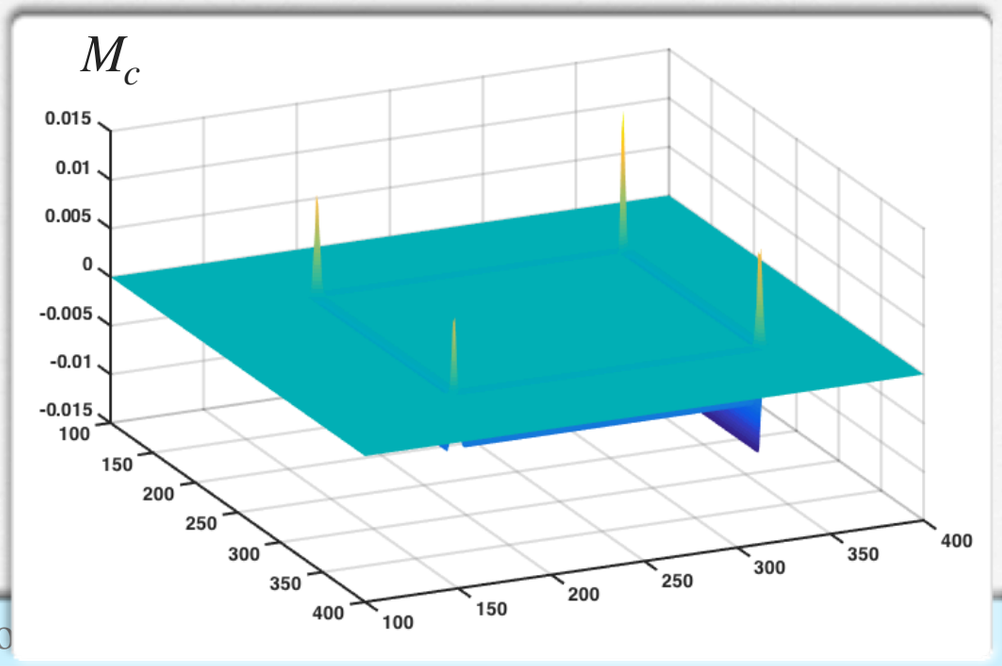
1. Tenseur de structure

a) Détecteur de Harris



$M_c > 0.005$

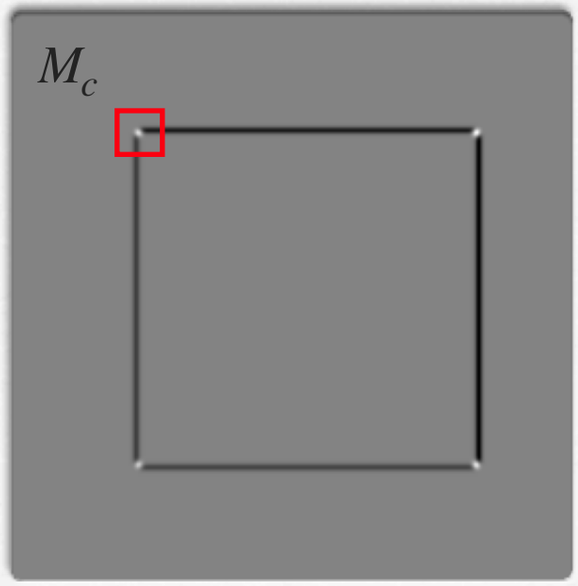


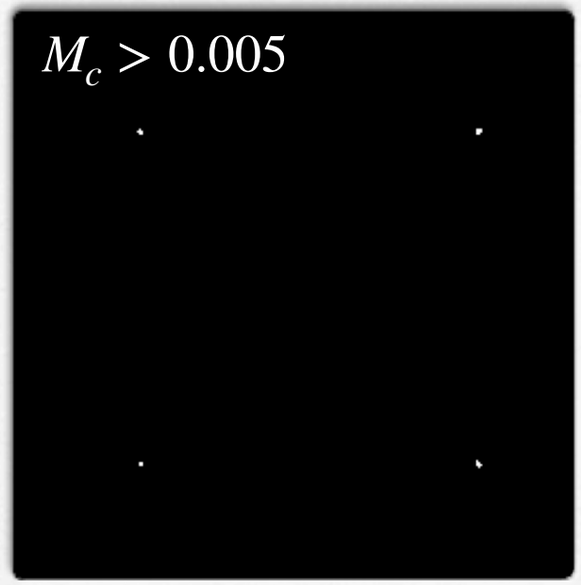


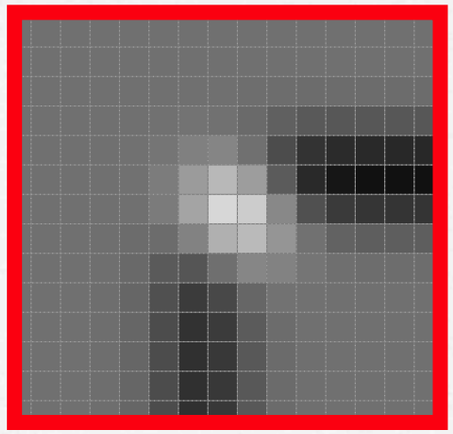
3. Caractéristiques ponctuelles

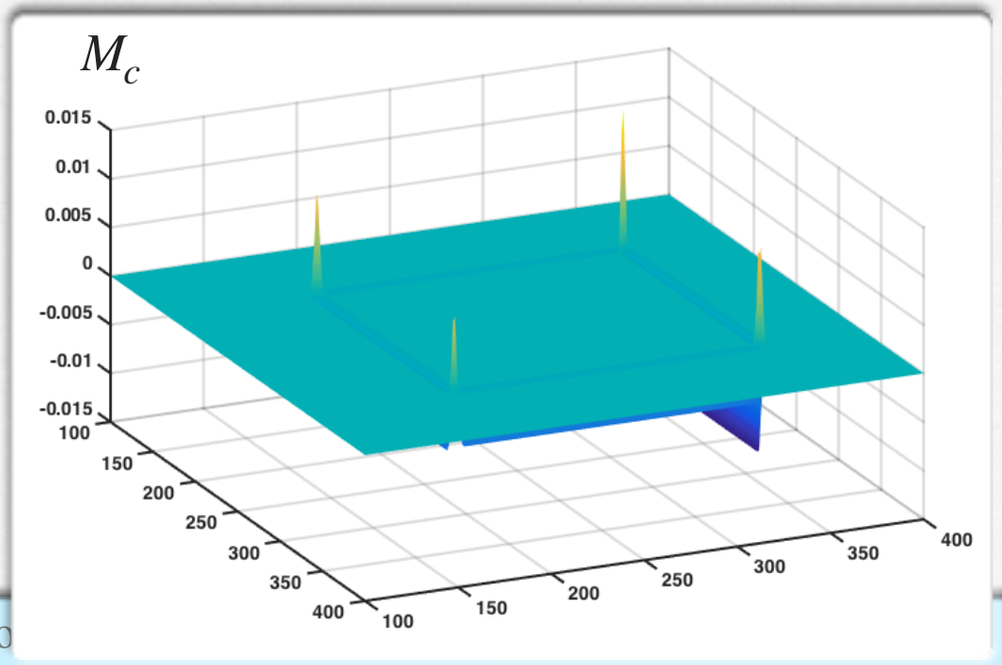
1. Tenseur de structure

a) Détecteur de Harris





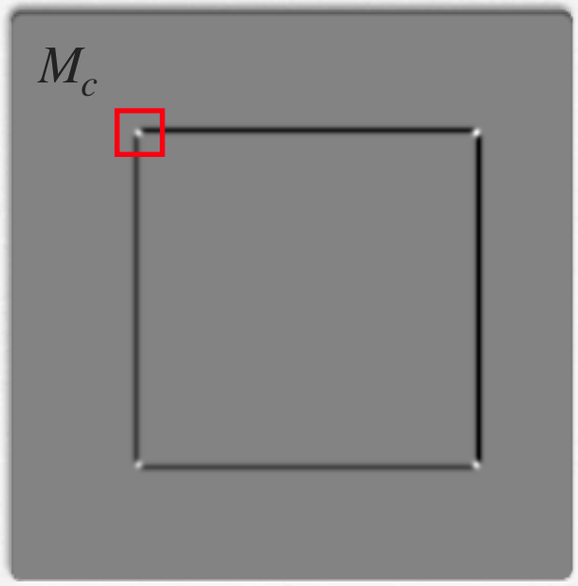


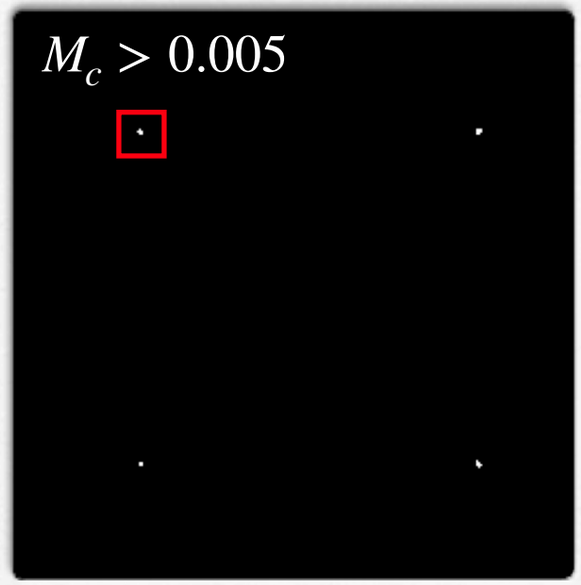


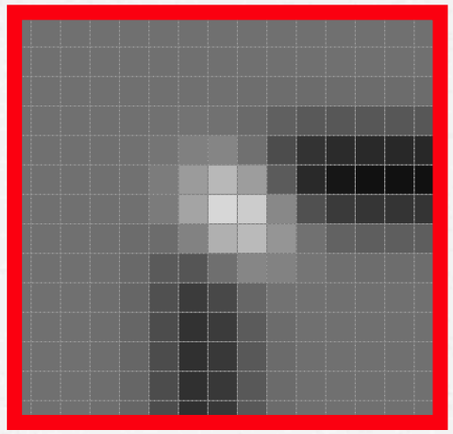
3. Caractéristiques ponctuelles

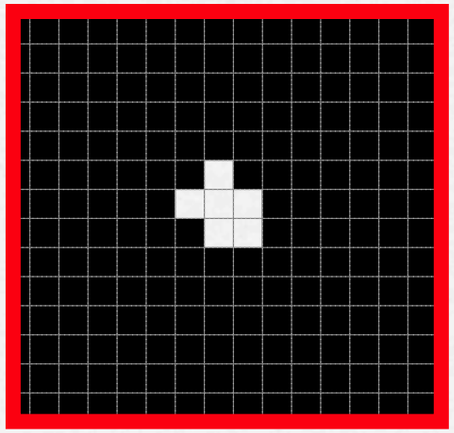
1. Tenseur de structure

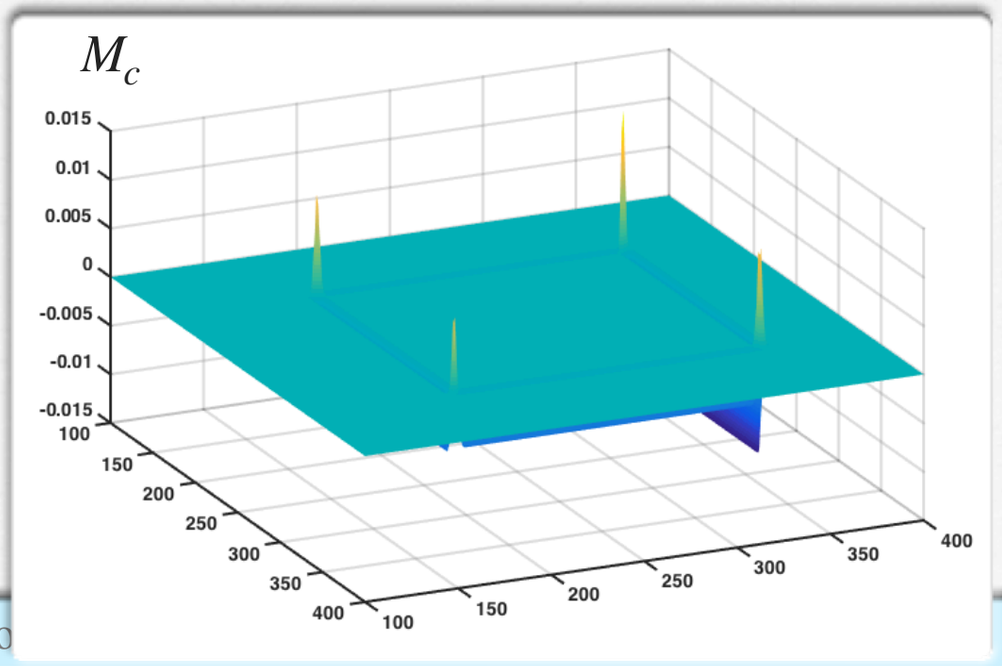
a) Détecteur de Harris







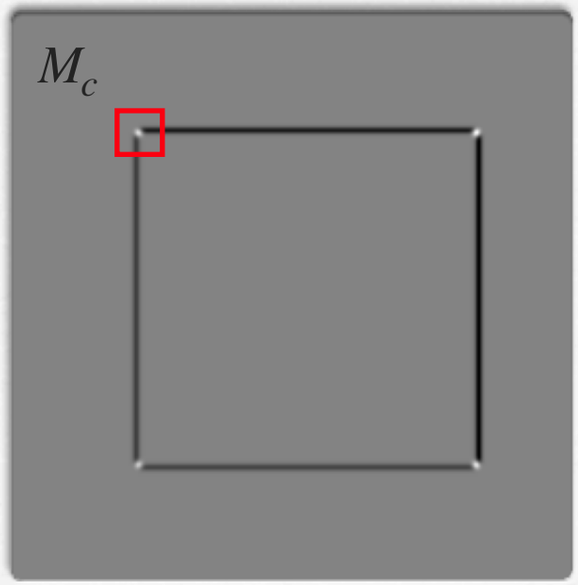


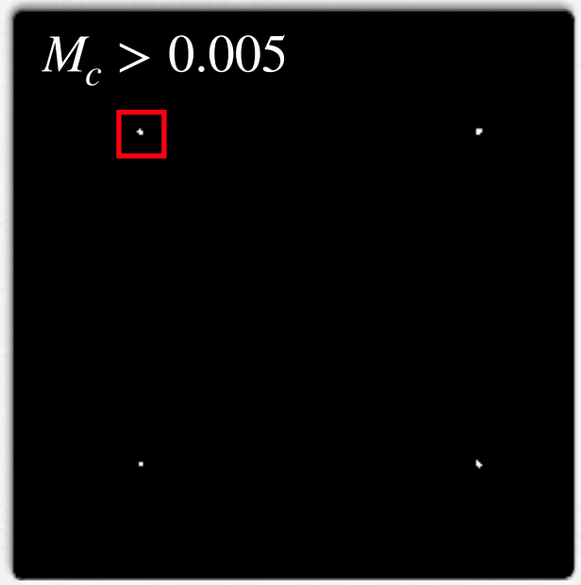


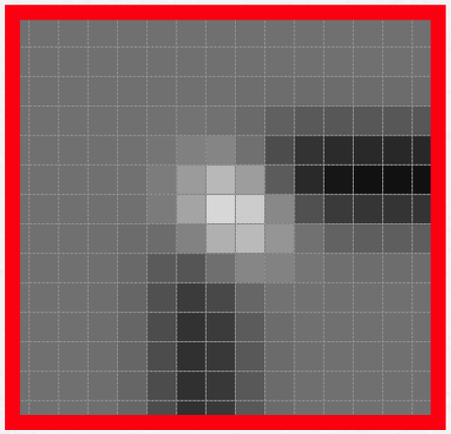
3. Caractéristiques ponctuelles

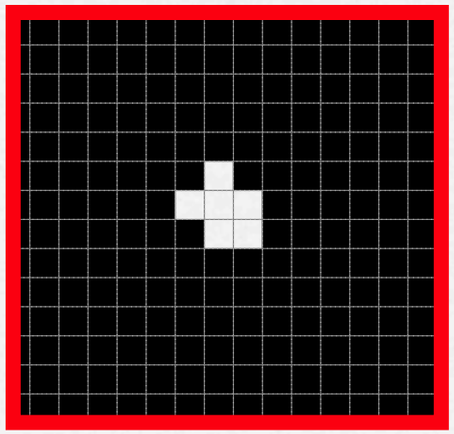
1. Tenseur de structure

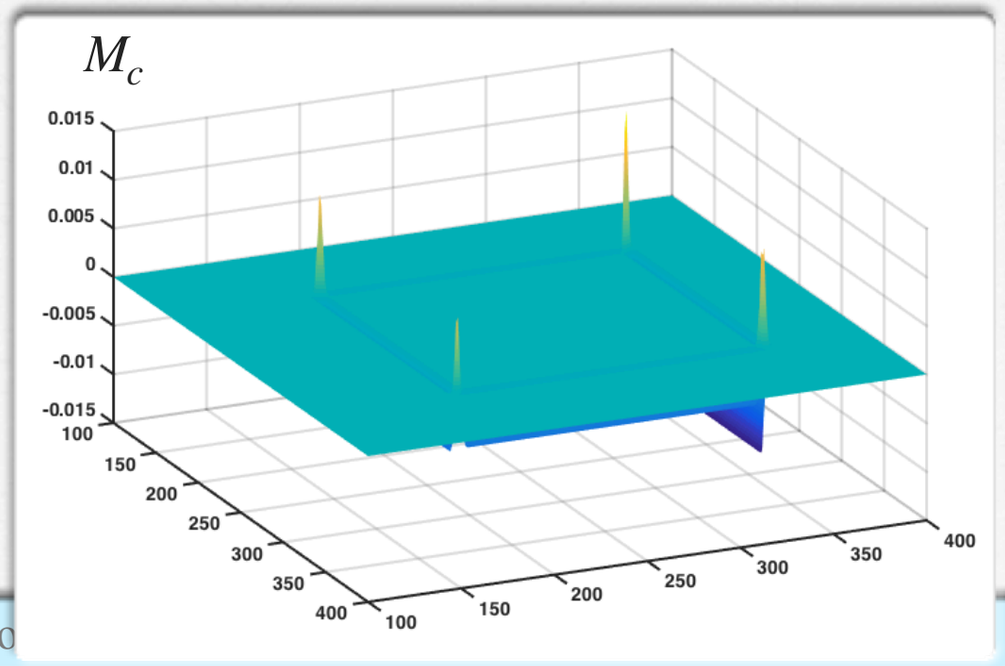
a) Détecteur de Harris











Effectuer une suppression des non-maxima locaux sur $M_c[m, n]$

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

Images originales

Source : Shai Bagon

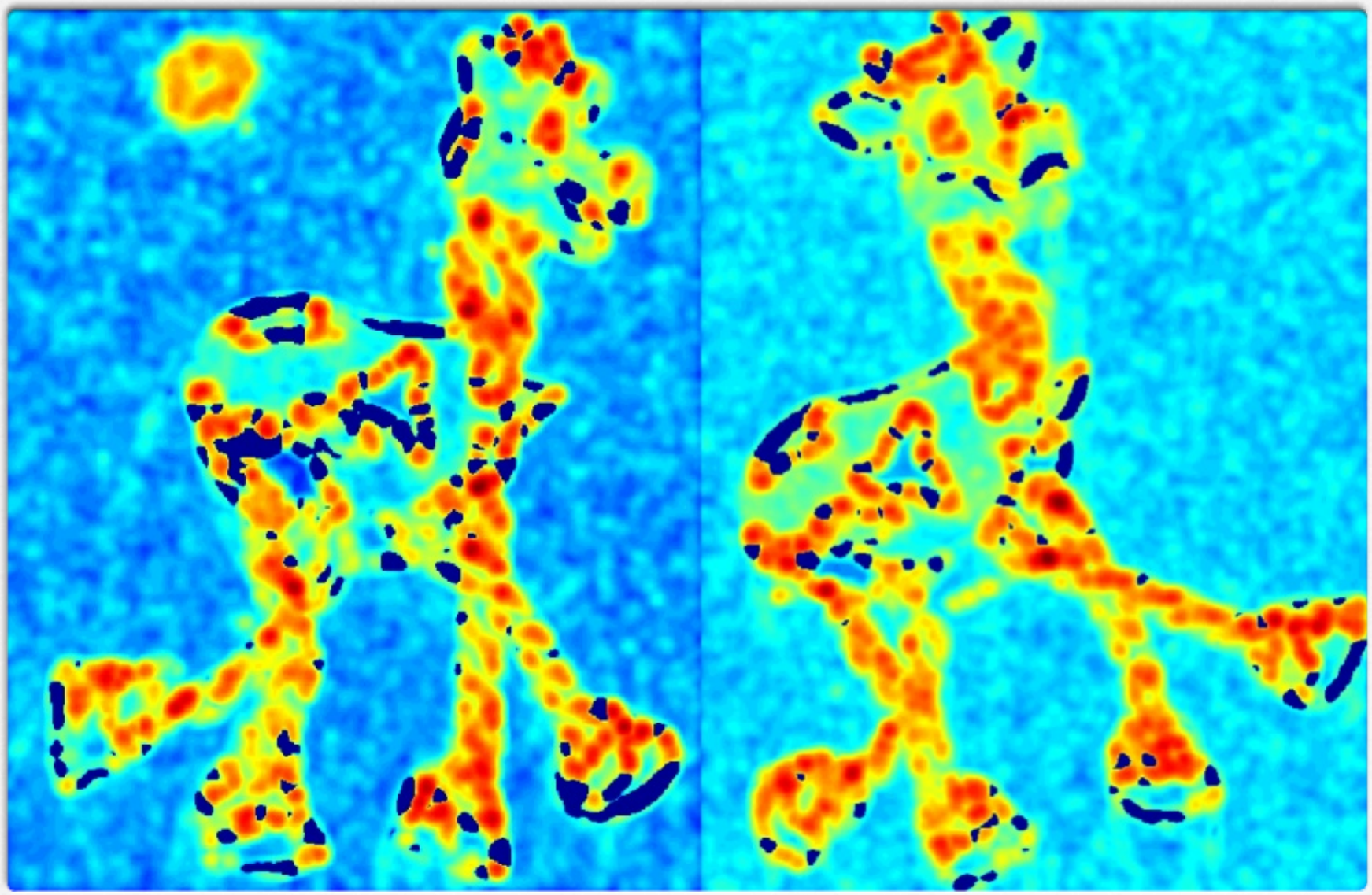
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

$M_c[m,n]$



Source : Shai Bagon

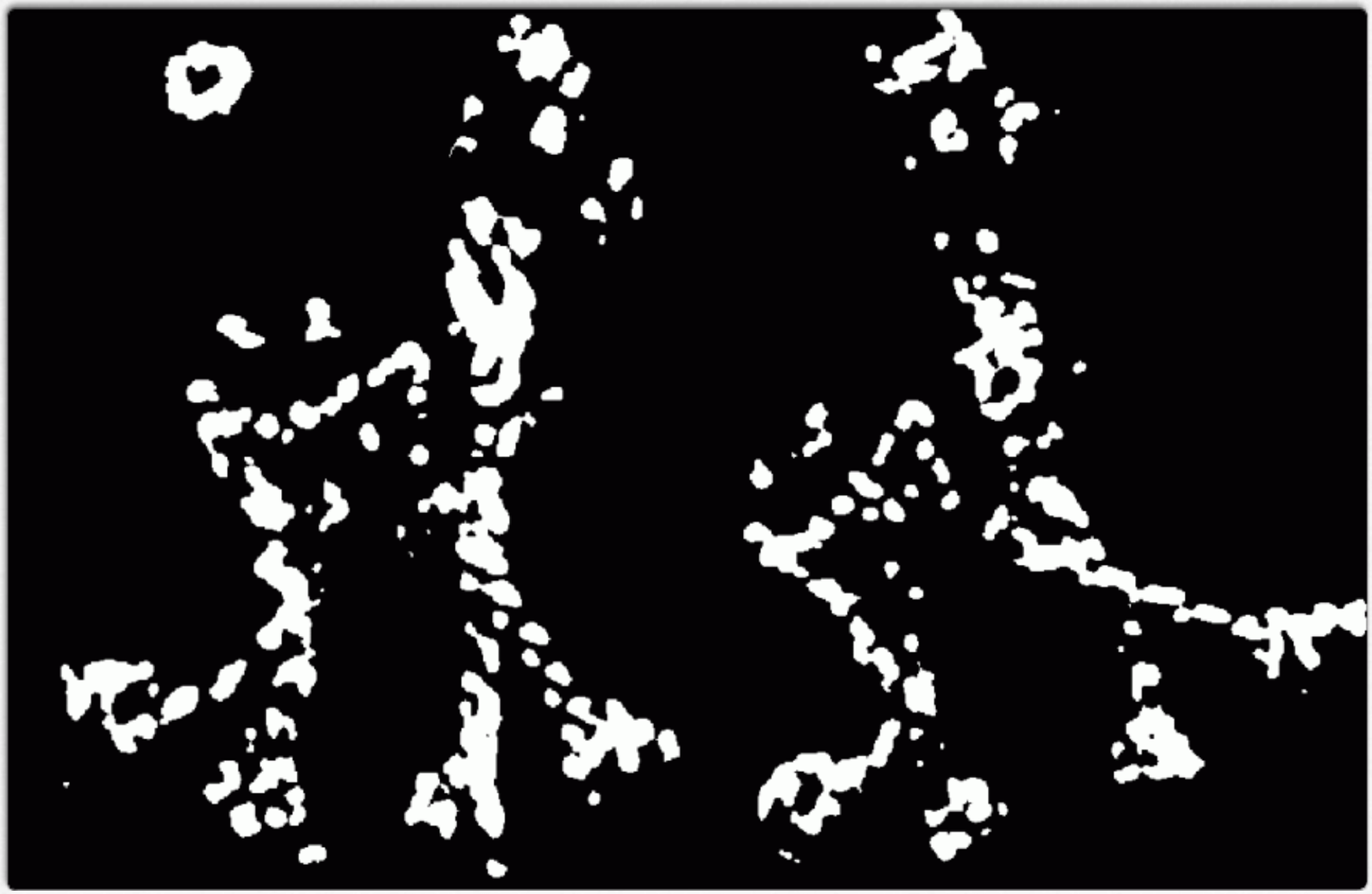
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

$M_c[m,n] >$ seuil



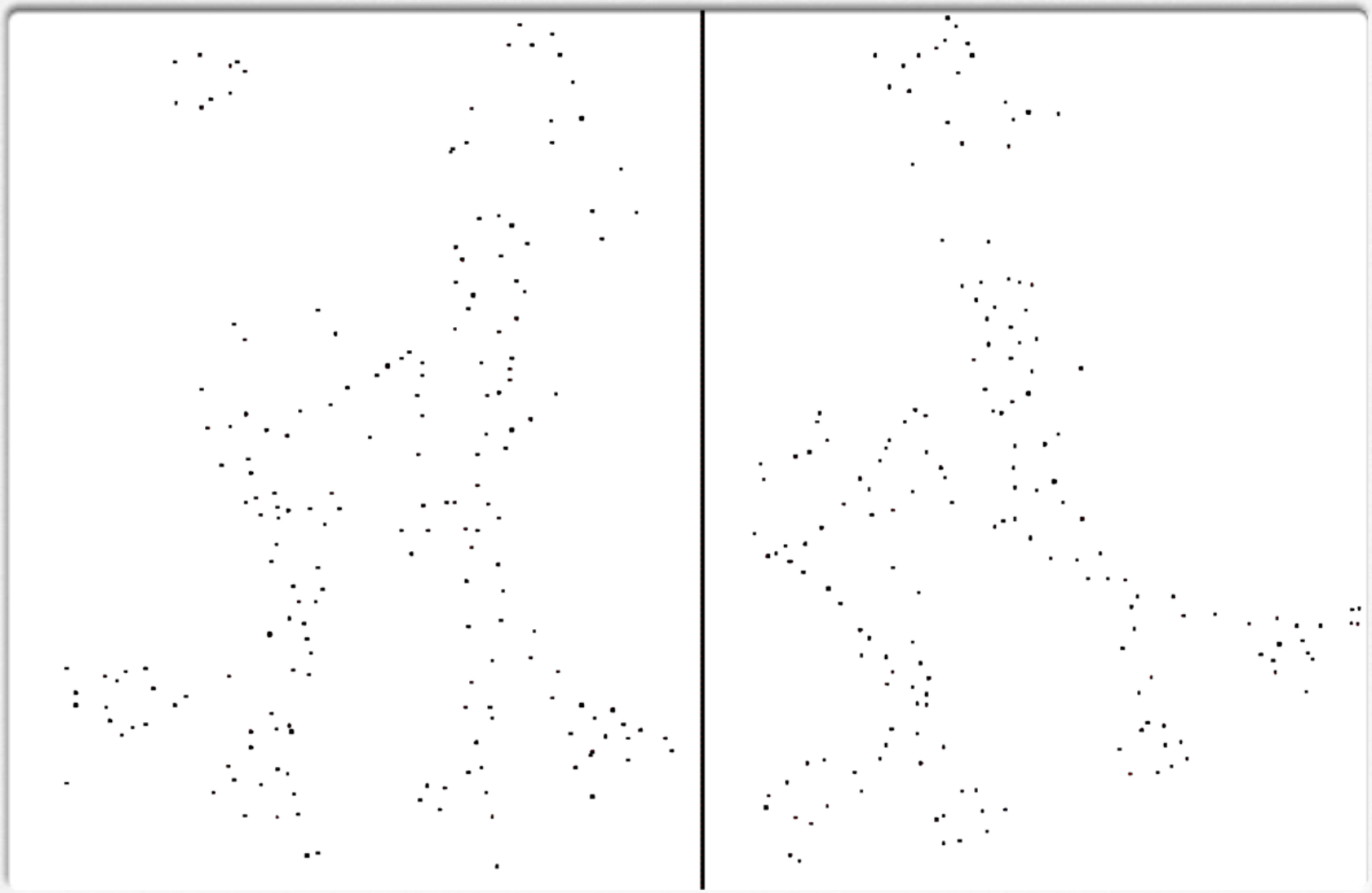
Source : Shai Bagon

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

$M_c[m,n]$ maximum local (inversé)



Source : Shai Bagon

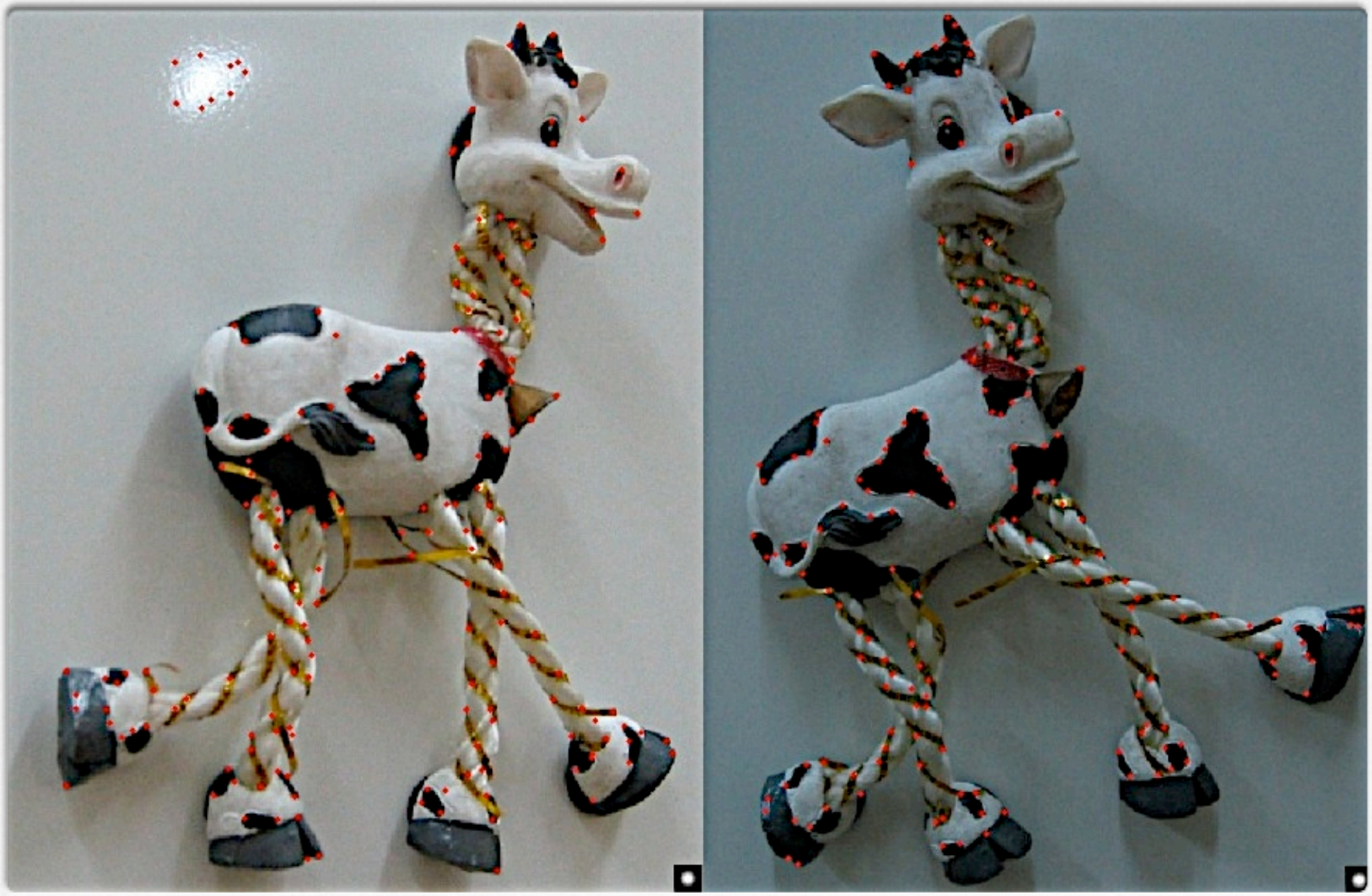
©Marie-Flavie Auclair-Fortier

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

$M_c[m,n]$ maximum localement superposés sur l'image originale



Source : Shai Bagon

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

- Invariance des jonctions face à la rotation

3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

- Invariance des jonctions face à la rotation

Voisinage gaussien $\sigma = 1.5$
$k = 0.04$

Source : PM Jodoin

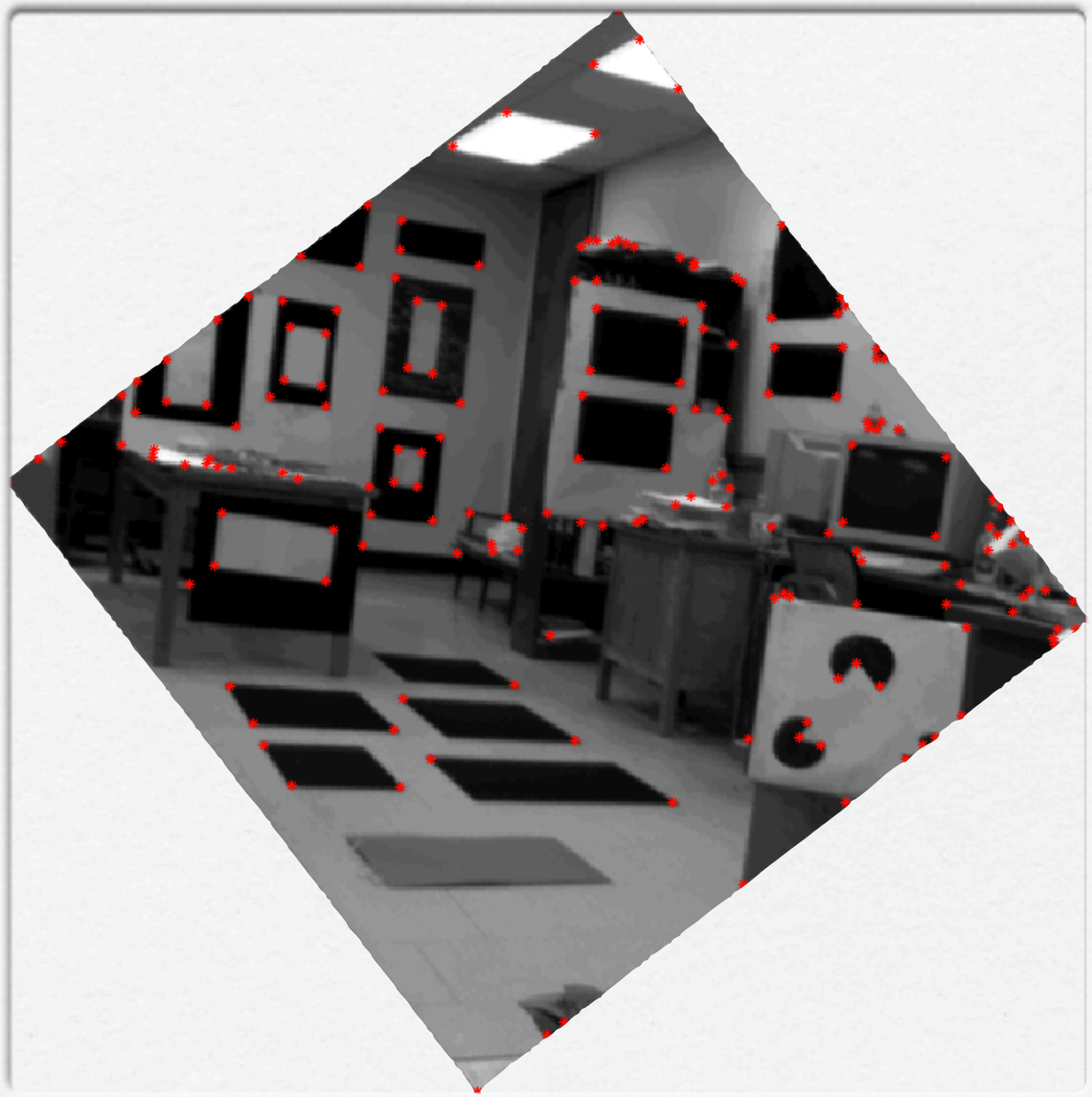
3. Caractéristiques ponctuelles

1. Tenseur de structure

a) Détecteur de Harris

- Invariance des jonctions face à la rotation

Voisinage gaussien $\sigma = 1.5$
$k = 0.04$



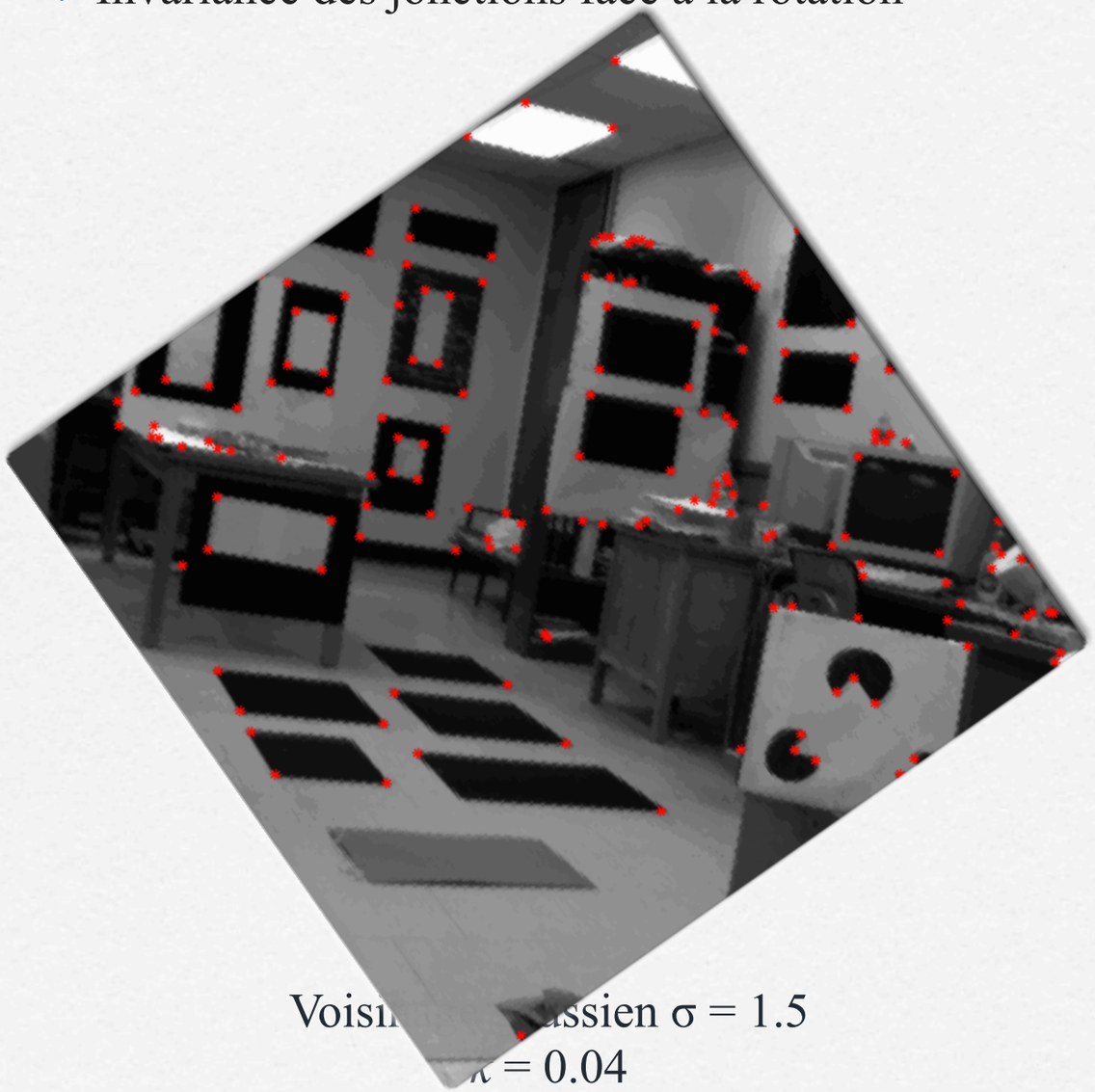
Source : PM Jodoin

3. Caractéristiques ponctuelles

1. Tenseur de structure

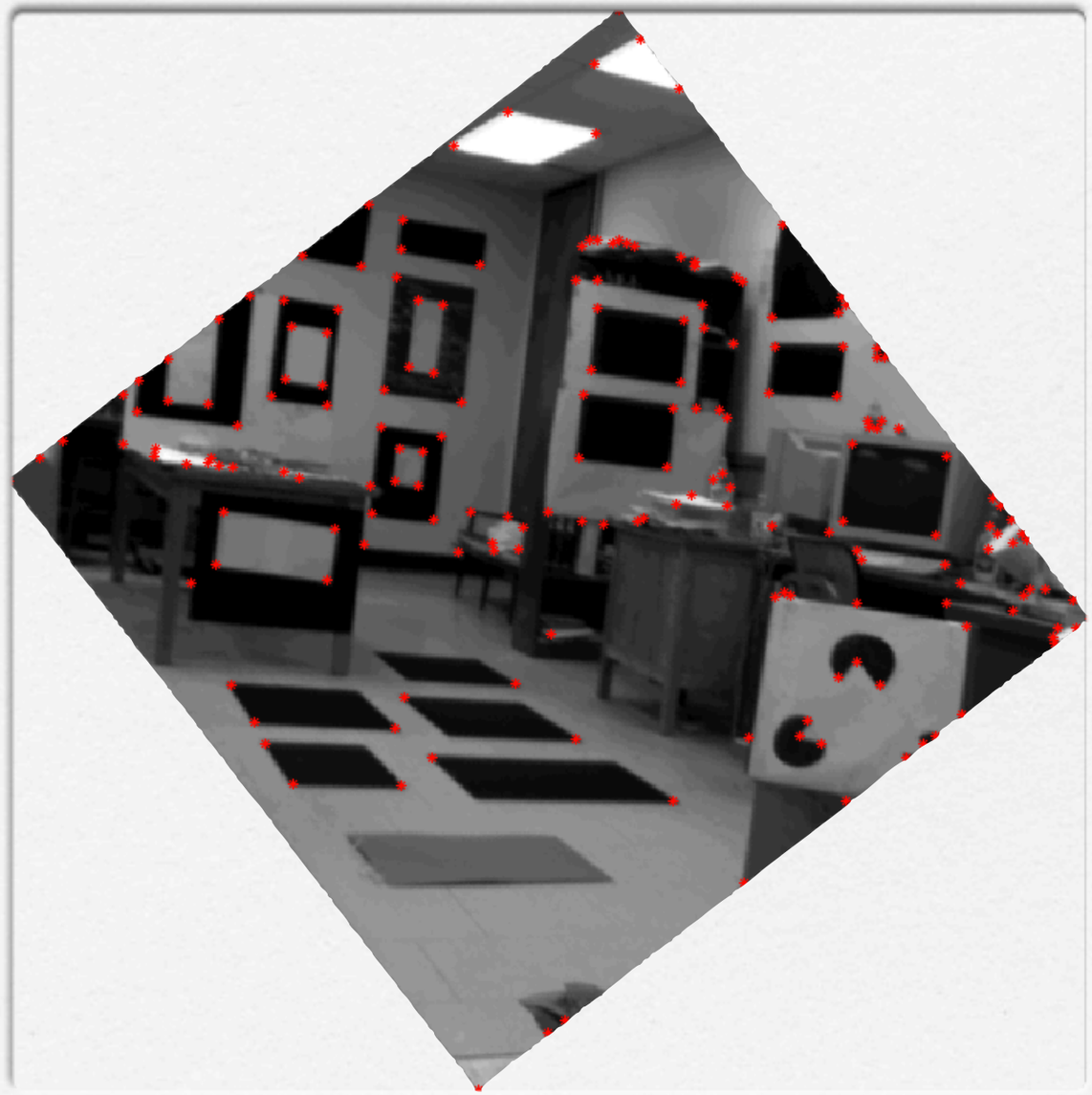
a) Détecteur de Harris

- Invariance des jonctions face à la rotation



Voisi[illegible]ssien $\sigma = 1.5$

$\kappa = 0.04$



Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

Détecteur de Harris

Détecteur de Harris avec suppression des non max

1. Calculer le gradient de l'image d'entrée

$$f_x = \frac{\partial f}{\partial x}, f_y = \frac{\partial f}{\partial y}$$

2. POUR CHAQUE pixel (x,y) de l'image FAIRE
   Calculer la matrice hessienne H autour d'un voisinage de taille $N \times N$

$$M_c(x, y) = \det(H) - \kappa \text{trace}^2(H)$$

2. POUR CHAQUE pixel (x,y) de l'image FAIRE
   POUR CHAQUE pixel (x+a,y+b) voisin de (x,y) FAIRE
   SI $M_c(x + a, y + b) > M_c(x, y)$ ALORS

$$M_c(x, y) = 0$$

4. $C = M_c > Seuil$

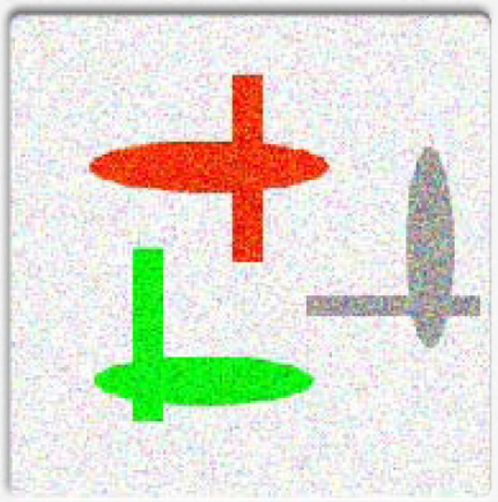
Plan

1. Calcul des dérivées d'une image
2. Contours
3. Caractéristiques ponctuelles
4. **Segmentation**
   - **4.1. Segmentation à base de classification**
     a) Seuillage
     b) K-Moyenne

4. RÉGIONS

Région

- ✓ Ensemble de pixels topologiquement connexes et ayant des attributs similaires : niveaux de gris, couleur, texture, mouvement.
- ✓ Une région est souvent différente d'un objet de la scène



Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

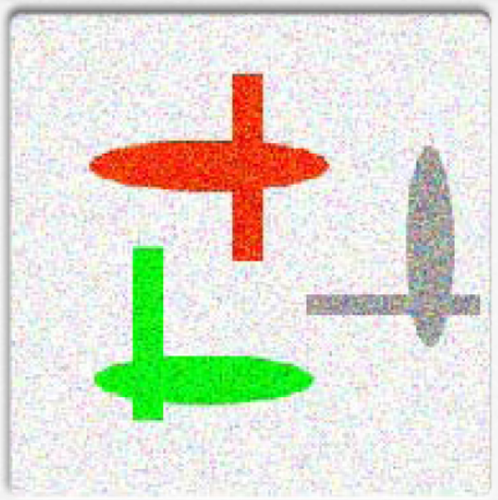
Région

- ✓ Ensemble de pixels topologiquement connexes et ayant des attributs similaires : niveaux de gris, couleur, texture, mouvement.
- ✓ Une région est souvent différente d'un objet de la scène

Segmentation

- ✓ Opération qui a pour but de rassembler des pixels entre eux suivant des critères pré-définis. Les pixels sont ainsi regroupés en régions, qui constituent une partition de l'image

- But
  - ✓ Description en objets : simplification de l'image



Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

Région

- ✓ Ensemble de pixels topologiquement connexes et ayant des attributs similaires : niveaux de gris, couleur, texture, mouvement.
- ✓ Une région est souvent différente d'un objet de la scène

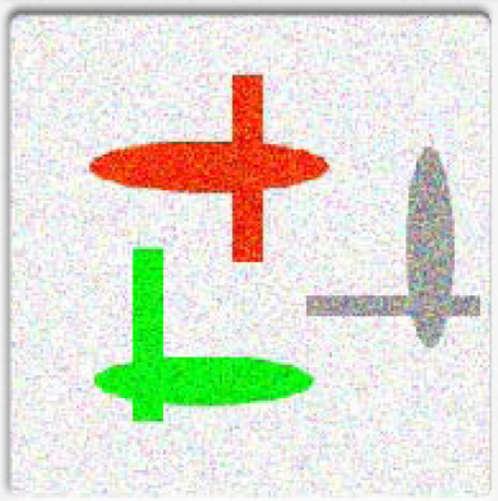
Classe

- ✓ Catégorie de pixels semblables

Segmentation

- ✓ Opération qui a pour but de rassembler des pixels entre eux suivant des critères pré-définis. Les pixels sont ainsi regroupés en régions, qui constituent une partition de l'image

- But
  - ✓ Description en objets : simplification de l'image



Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

Région

- ✓ Ensemble de pixels topologiquement connexes et ayant des attributs similaires : niveaux de gris, couleur, texture, mouvement.
- ✓ Une région est souvent différente d'un objet de la scène

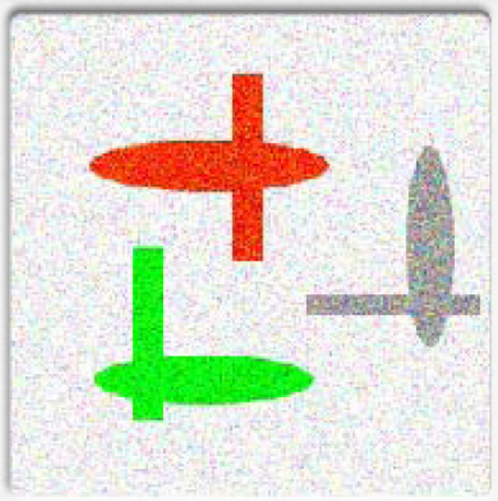
Classe

- ✓ Catégorie de pixels semblables

Segmentation

- ✓ Opération qui a pour but de rassembler des pixels entre eux suivant des critères pré-définis. Les pixels sont ainsi regroupés en régions, qui constituent une partition de l'image

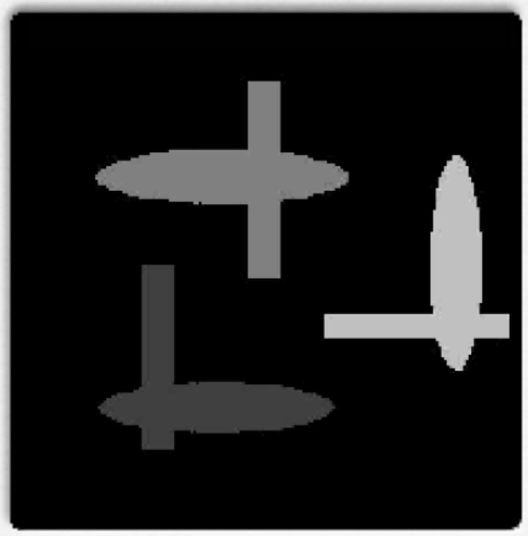
- But
  - ✓ Description en objets : simplification de l'image



Segmentation en 4 classes/régions

$r\,[m, n] \in$ {fond, objet vert, objet rouge, objet gris}

étiquettes



Source : PM Jodoin

4. RÉGIONS

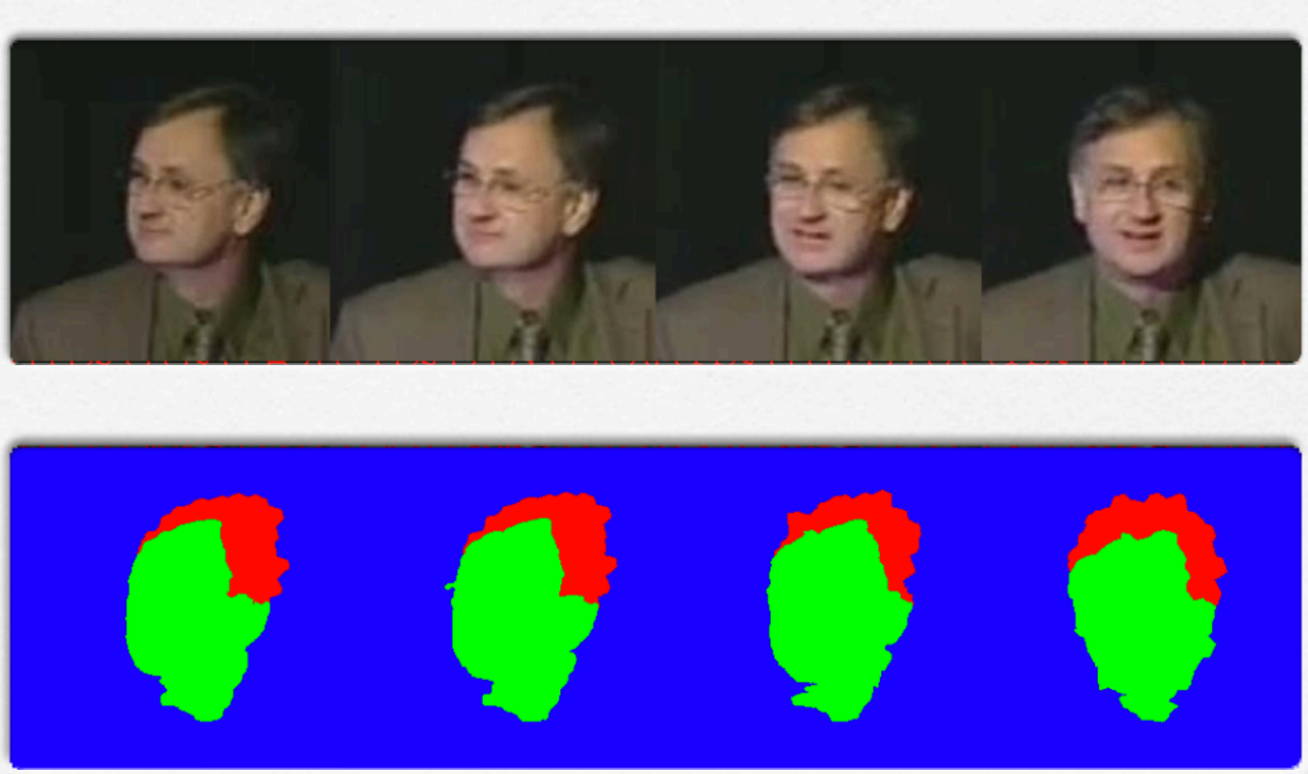
- segmentation des objets saillants

Source : A. Hati *et al.*, An image texture insensitive method for saliency detection. 2017

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- A quoi sert la segmentation en régions?
  - ✓ Important prérequis pour les étapes de mesure, de compréhension de la scène :
    - ➡ reconnaissance d'objets
    - ➡ indexation : rechercher dans une base d'images, les images "ressemblantes" à une image initiale
    - ➡ compression
    - ➡ recalage d'images, mises en correspondance

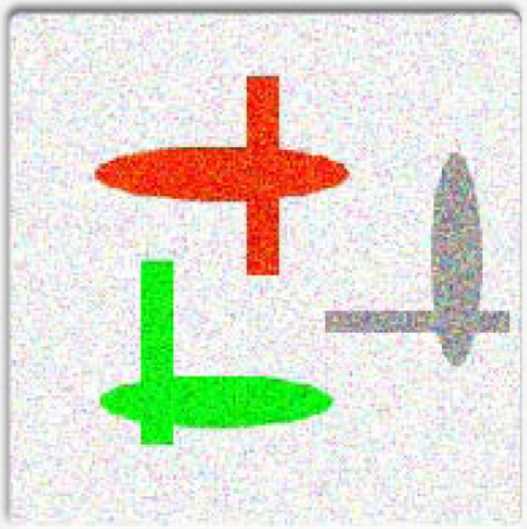


Source : A. Kae *et al.* The Shape-Time Random Field for Semantic Video Labeling, CVPR 2014

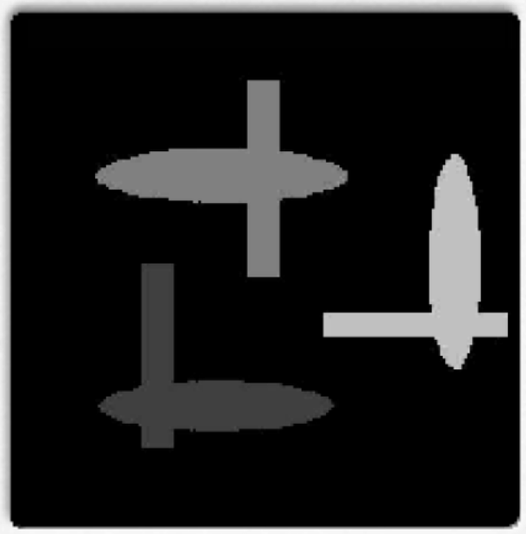
©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Prédicat
  - ✓ *Exemple*



Segmentation en 4 classes/régions

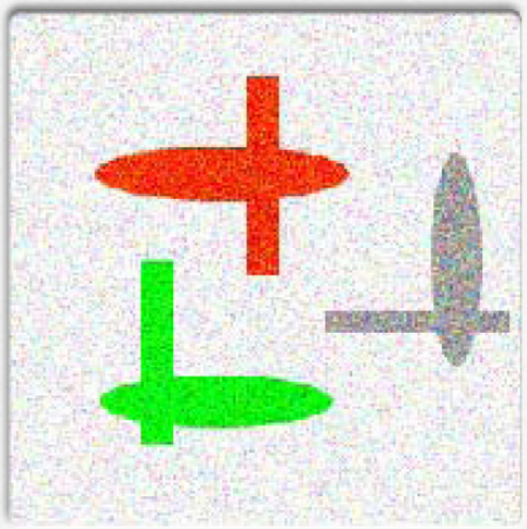


$r\,[m, n] \in$ {fond, objet vert, objet rouge, objet gris}

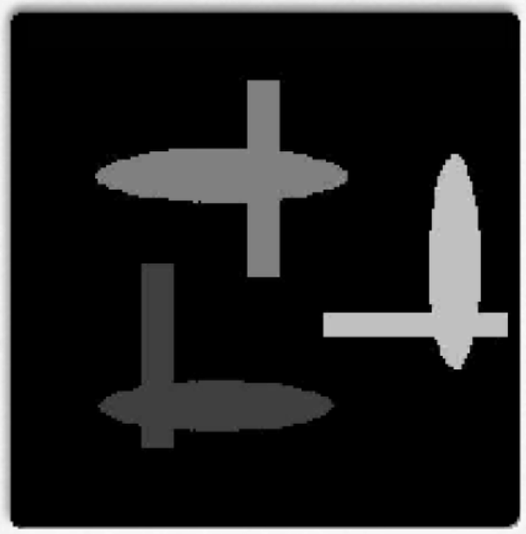
Source : PM Jodoin

4. RÉGIONS

- Prédicat
  - ✓ *Exemple*



Segmentation en 4 classes/régions



$r\,[m, n] \in$ {fond, objet vert, objet rouge, objet gris}

Prédicat	Classe 1 fond	Classe 2 obj. vert	Classe 3 obj. rouge	Classe 4 obj. gris
$\mathcal{P}_1$ pixels gris pâles	TRUE	FALSE	FALSE	FALSE
$\mathcal{P}_2$ pixels verts	FALSE	TRUE	FALSE	FALSE
$\mathcal{P}_3$ pixels rouges	FALSE	FALSE	TRUE	FALSE
$\mathcal{P}_4$ pixels gris moyen	FALSE	FALSE	FALSE	TRUE

Source : PM Jodoin

4. RÉGIONS

- Segmentation
  - ✓ *Exemple*

    2 classes, mais plus que 2 régions

$r[m, n]$ est une image binaire

2 classes

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Segmentation
  - ✓ *Exemple*

    2 classes, mais plus que 2 régions

$r[m, n]$ est une image binaire

2 classes

$r[m, n] \in \{\text{terre}, \text{mer}\}$

étiquettes

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Segmentation
  - ✓ *Exemple*

    2 classes, mais plus que 2 régions

Remarques :

- 2 régions non connexes peuvent appartenir à la même classe

$r[m, n]$ est une image binaire

2 classes

$r[m, n] \in \{\text{terre}, \text{mer}\}$

étiquettes

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Segmentation
  - ✓ *Exemple*

    2 classes, mais plus que 2 régions

Remarques :

- 2 régions non connexes peuvent appartenir à la même classe
  - R_i non adjacent à R_j et P($R_i \cup R_j$) est faux
- Classification ≠ segmentation
  - Une classe de pixels = plusieurs régions dans l'image
  - Il faut étiqueter les pixels connexes pour obtenir des régions

$r[m, n]$ est une image binaire

2 classes

$r[m, n] \in \{\text{terre}, \text{mer}\}$

étiquettes

Source : PM Jodoin

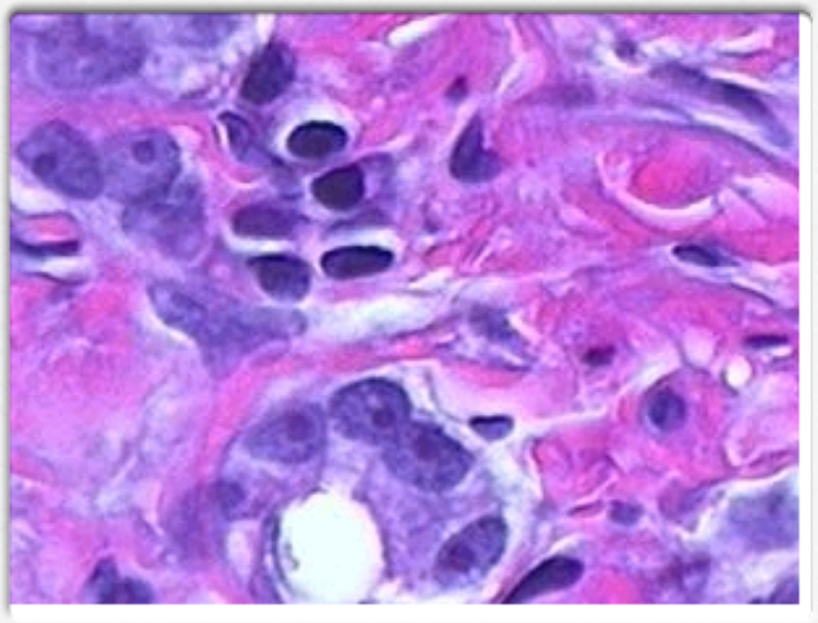
©Marie-Flavie Auclair-Fortier

4. RÉGIONS

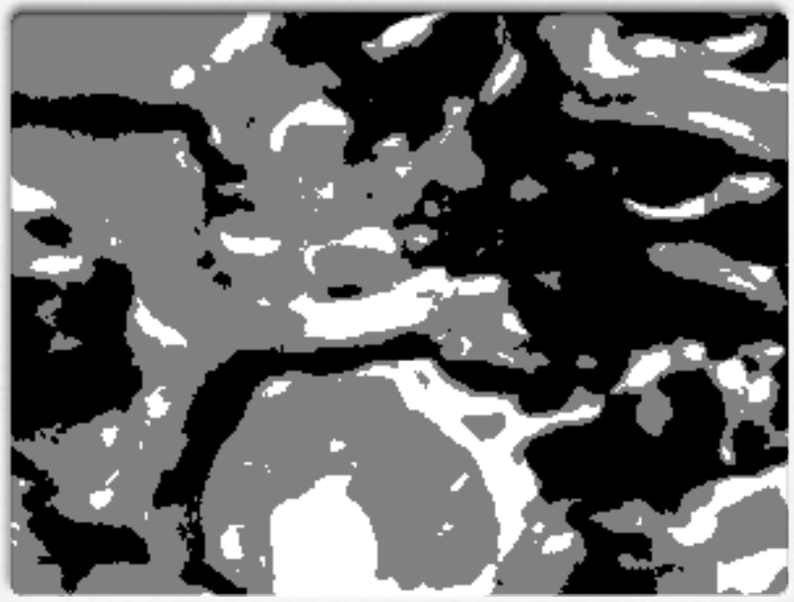
- Segmentation
  - ✓ *Exemple*

Remarque :

- 3 classes, mais plus que 3 régions



3 classes



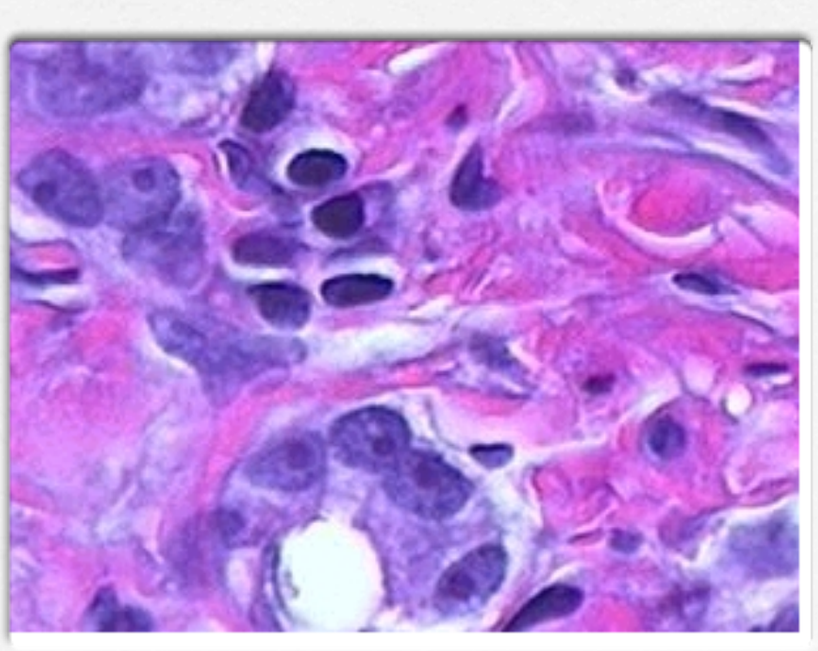
Source : Docum. Matlab

4. RÉGIONS

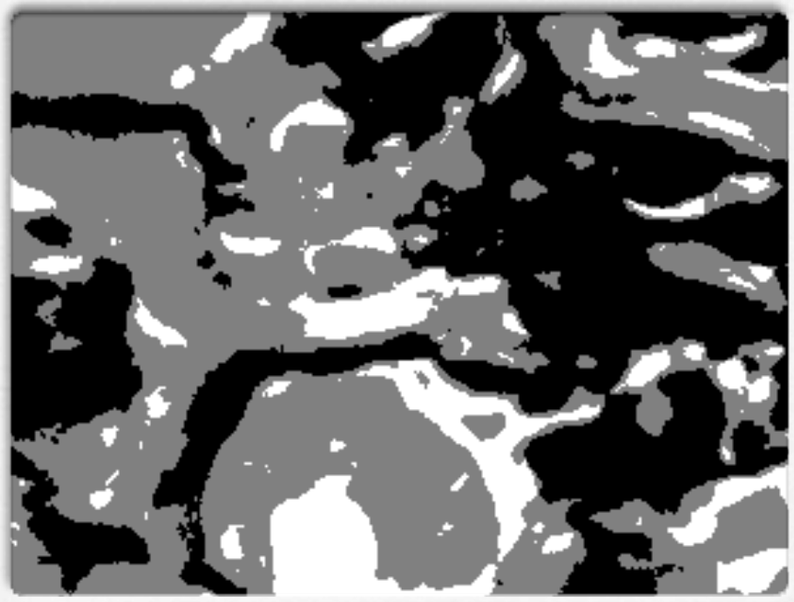
- Segmentation
  - ✓ *Exemple*

Remarque :

- 3 classes, mais plus que 3 régions



3 classes



r [*m, n*] est une image à trois valeurs

Source : Docum. Matlab

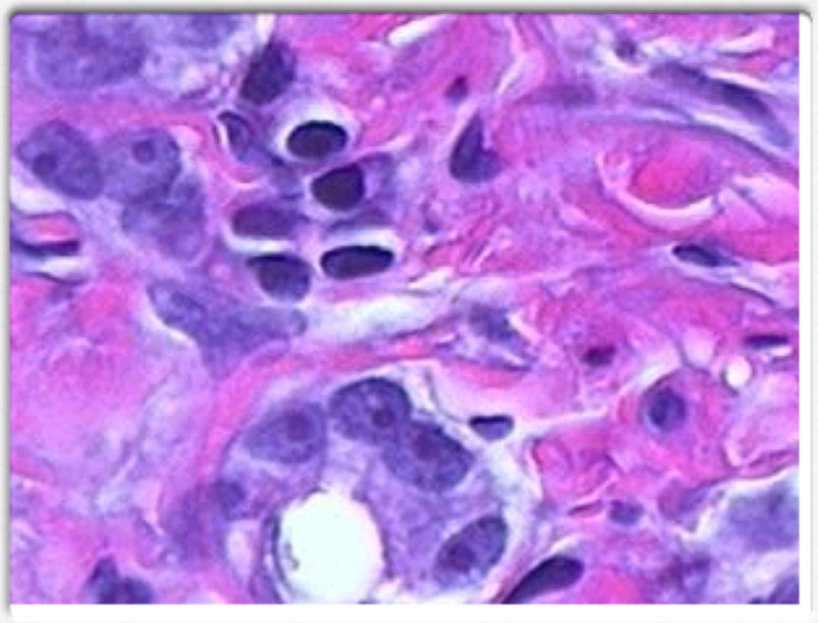
©Marie-Flavie Auclair-Fortier

4. RÉGIONS

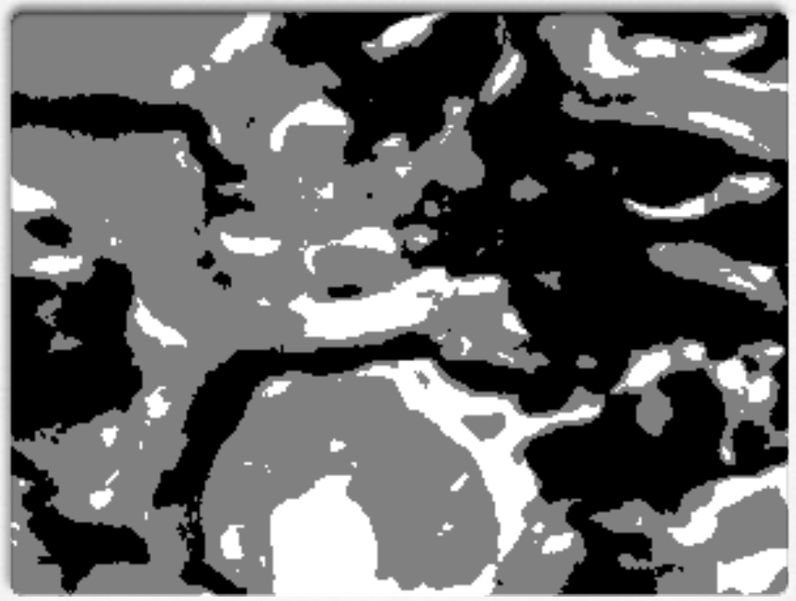
- Segmentation
  - ✓ *Exemple*

Remarque :

- 3 classes, mais plus que 3 régions



3 classes



$r\,[m, n]$ est une image à trois valeurs

$$r\,[m, n] \in \{C_1, C_2, C_3\}$$

étiquettes

Source : Docum. Matlab

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Segmentation
  - ✓ *Exemple*

Remarque :

- La sémantique n'est pas toujours claire

5 classes

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Segmentation
  - ✓ *Exemple*

Remarque :

- La sémantique n'est pas toujours claire

5 classes

$r\,[m, n] \in \{C1, C2, C3, C4, C5\}$

étiquettes

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Segmentation
  - ✓ *Exemple*

Remarque :

- La sémantique n'est pas toujours claire

5 classes

$r\,[m, n] \in \{C1, C2, C3, C4, C5\}$

étiquettes

Remarques

- Dualité contours/régions : intérieur d'un contour fermé = région
- Attributs utilisés pour la segmentation très variés (niveaux de gris, couleur, texture, mouvement)

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- Segmentation

Remarque :

- un prédicat peut inclure une ou plusieurs caractéristiques et des conditions sur
  - couleur, n.g.
  - texture, statistique
  - mouvement, profondeur
  - ...

2 classes

Source :

©Marie-Flavie Auclair-Fortier

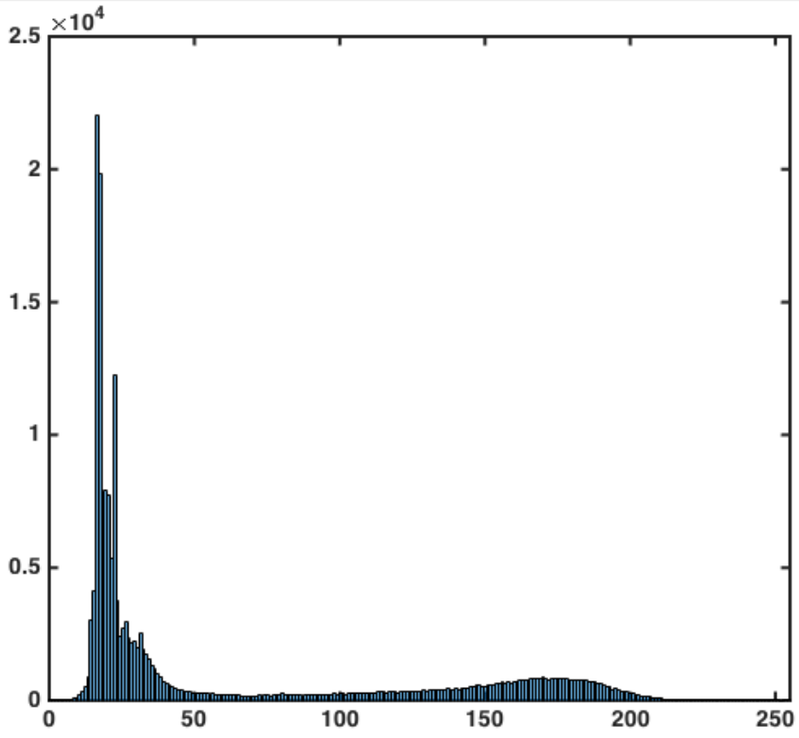
4. RÉGIONS

- Segmentation

Remarque :

- un prédicat peut inclure une ou plusieurs caractéristiques et des conditions sur
  - couleur, n.g.
  - texture, statistique
  - mouvement, profondeur
  - ...

2 classes



Source :

©Marie-Flavie Auclair-Fortier

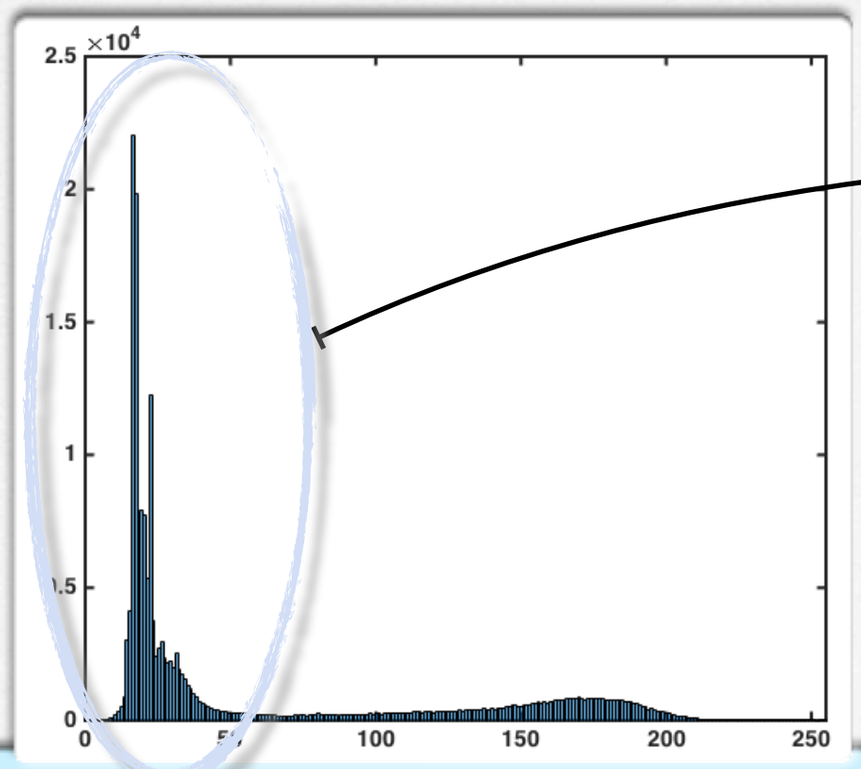
4. RÉGIONS

- Segmentation

Remarque :

- un prédicat peut inclure une ou plusieurs caractéristiques et des conditions sur
  - couleur, n.g.
  - texture, statistique
  - mouvement, profondeur
  - ...

2 classes



Niveaux de gris foncés = Mer

Source :

©Marie-Flavie Auclair-Fortier

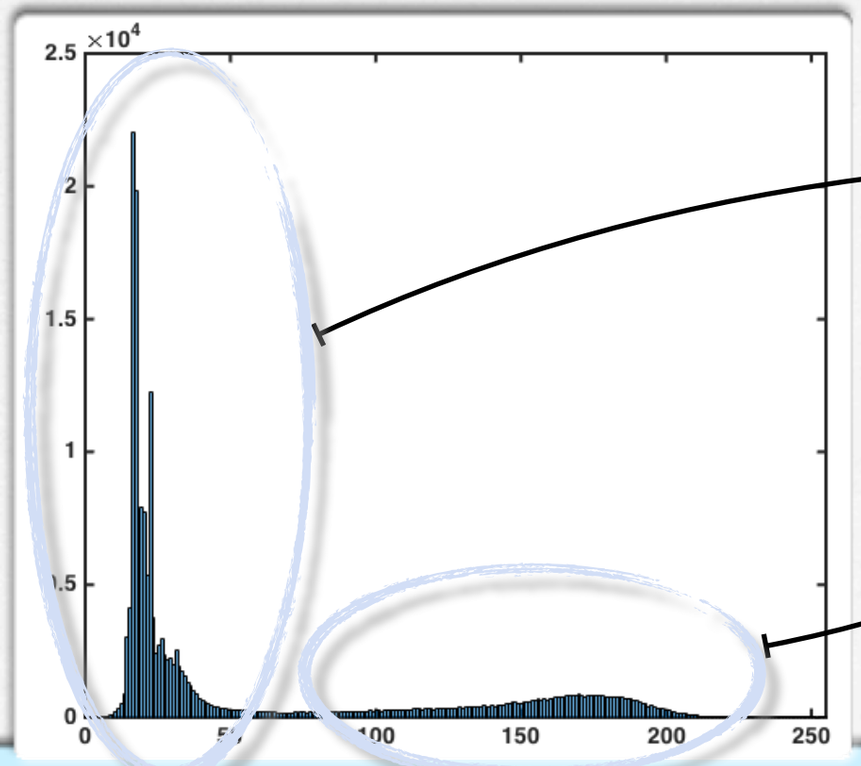
4. RÉGIONS

- Segmentation

Remarque :

- un prédicat peut inclure une ou plusieurs caractéristiques et des conditions sur
  - couleur, n.g.
  - texture, statistique
  - mouvement, profondeur
  - ...

2 classes



Niveaux de gris foncés = Mer

Niveaux de gris pâles = Terre

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- *Problème* : Quel prédicat choisir?
  - ✓∄ de définition universelle
  - ✓On peut regrouper plusieurs prédicats en un seul

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

- *Problème* : Quel prédicat choisir?
  - ✓∄ de définition universelle
  - ✓On peut regrouper plusieurs prédicats en un seul

C'est un **problème mal posé**

- La solution n'est pas unique
- la solution ne dépend pas uniquement des données
  - choix du prédicat
  - ordre d'évaluation des pixels

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

a) Seuillage

Algorithme 4.7 : Seuillage

Entrées :	$f[m, n]$	→Image d'origine
	S	→Seuil
Sorties :	$r[m, n]$	→ une carte binaires des deux classes

1	Pour tous les pixels $[m, n]$	
1.1	si $f[m, n] > S$	// Prédicat : n. g. > S
	$r[m, n] = 1$	
	sinon	
	$r[m, n] = 0$	

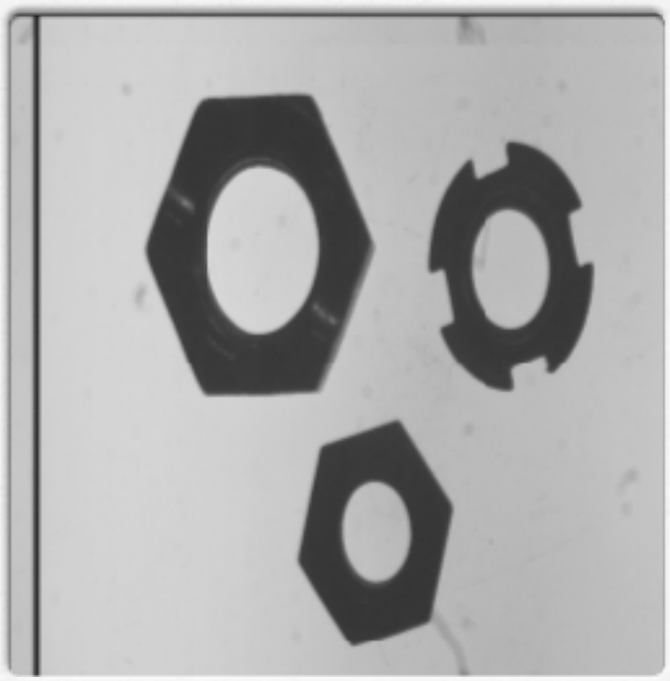
Source :

©Marie-Flavie Auclair-Fortier

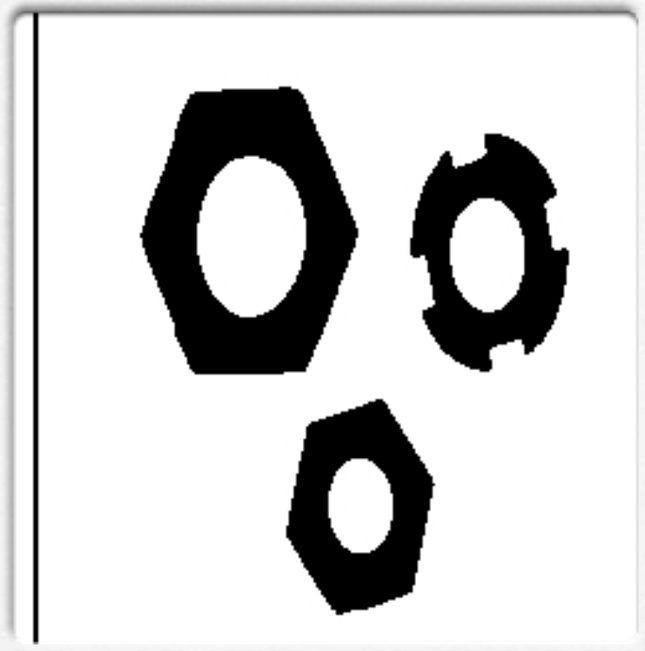
4. RÉGIONS

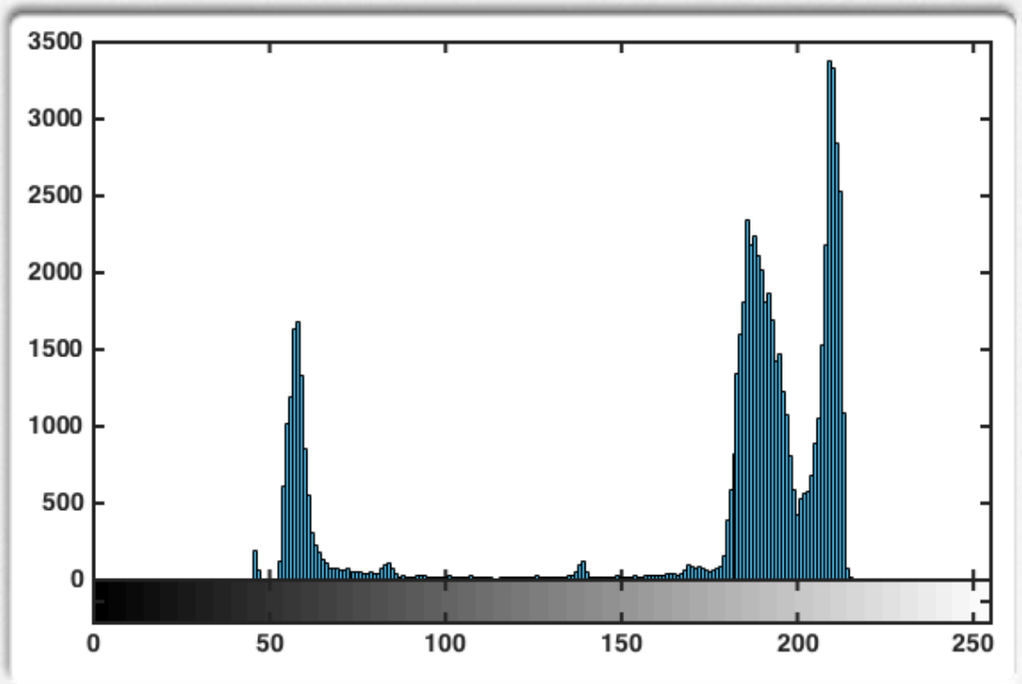
1. Segmentation à base de classification

a) Seuillage



Segmentation en 2 classes





Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

a) Seuillage

- Avantages
  - ✓ facile à implémenter

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

a) Seuillage

- Avantages
  - ✓ facile à implémenter
  - ✓ rapide

Source : PM Jodoin

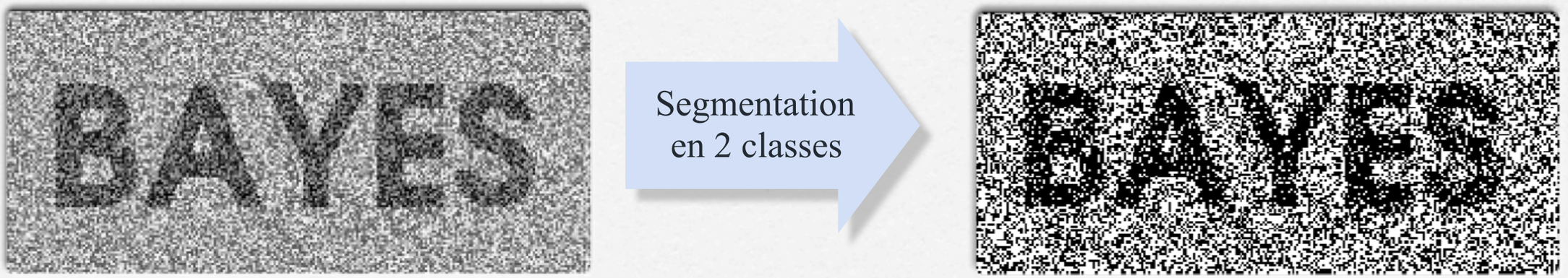
©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

a) Seuillage

- Avantages
  - ✓ facile à implémenter
  - ✓ rapide
- Inconvénients
  - ✓ Pas très efficace pour les images bruitées



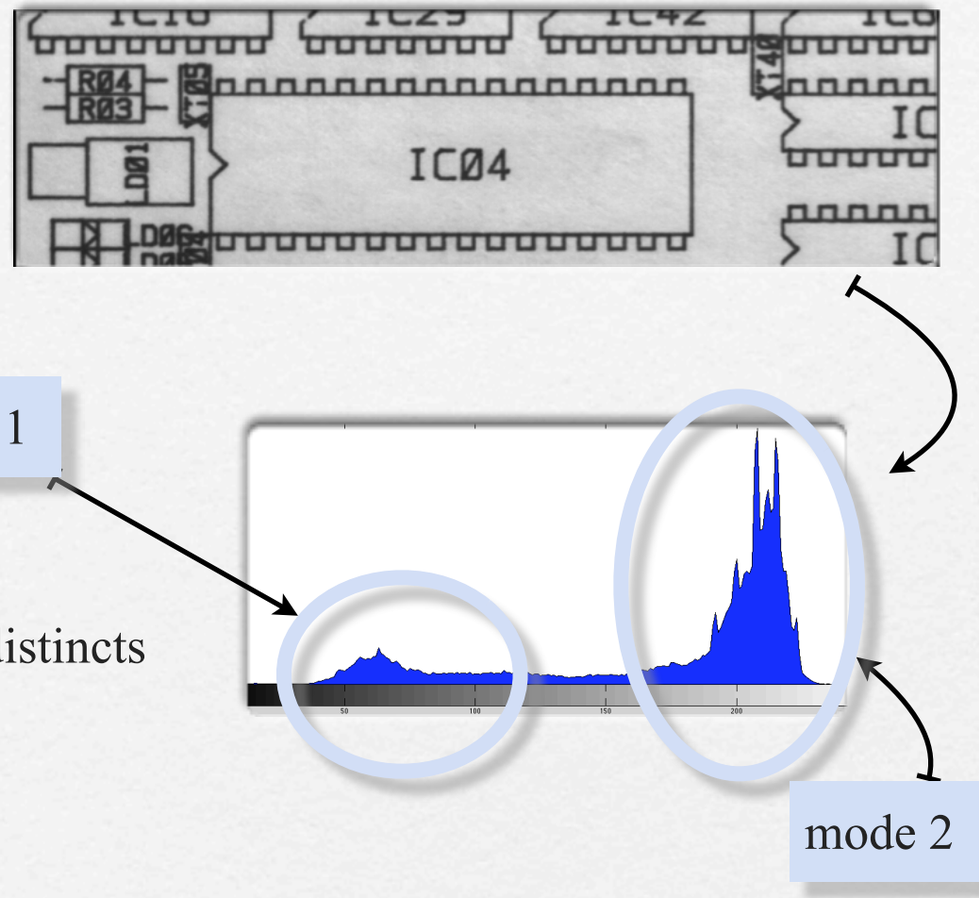
Source : PM Jodoin

4. RÉGIONS

1. Segmentation à base de classification

a) Seuillage

- Avantages
  - ✓ facile à implémenter
  - ✓ rapide
- Inconvénients
  - ✓ Pas très efficace pour les images bruitées
  - ✓ Fonctionne bien avec les images contenant 2 modes de n. g. très distincts



Segmentation en 2 classes

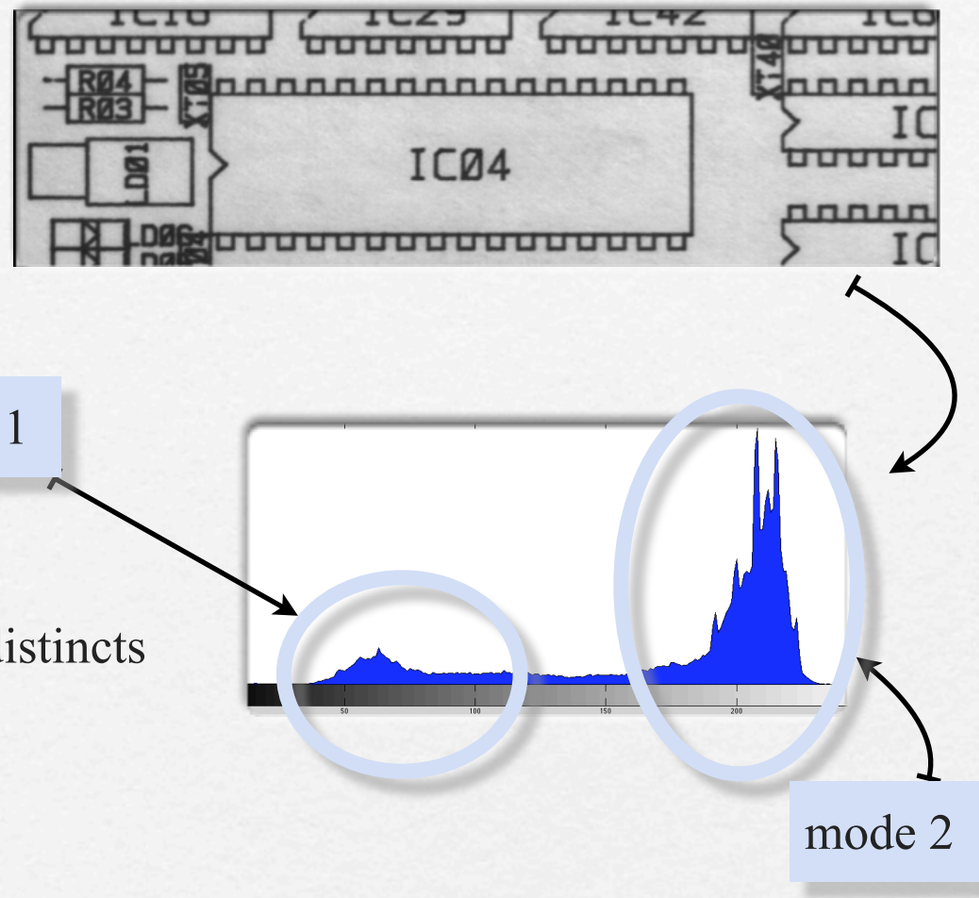
Source : PM Jodoin

4. RÉGIONS

1. Segmentation à base de classification

a) Seuillage

- Avantages
  - ✓ facile à implémenter
  - ✓ rapide
- Inconvénients
  - ✓ Pas très efficace pour les images bruitées
  - ✓ Fonctionne bien avec les images contenant 2 modes de n. g. très distincts
  - ✓ Les résultats dépendent du seuil → algorithme supervisé



Segmentation en 2 classes

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

a) Seuillage

- Problème
  - ✓ Comment déterminer le meilleur seuil automatiquement?

Hypothèses :

- On a une image à 2 modes
- On peut représenter chaque mode par une fonction continue
  - ✓ on peut démontrer que le meilleur seuil est à l'endroit où les deux fonctions se croisent
  - ✓ endroit qui maximise la variabilité entre les 2 modes

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

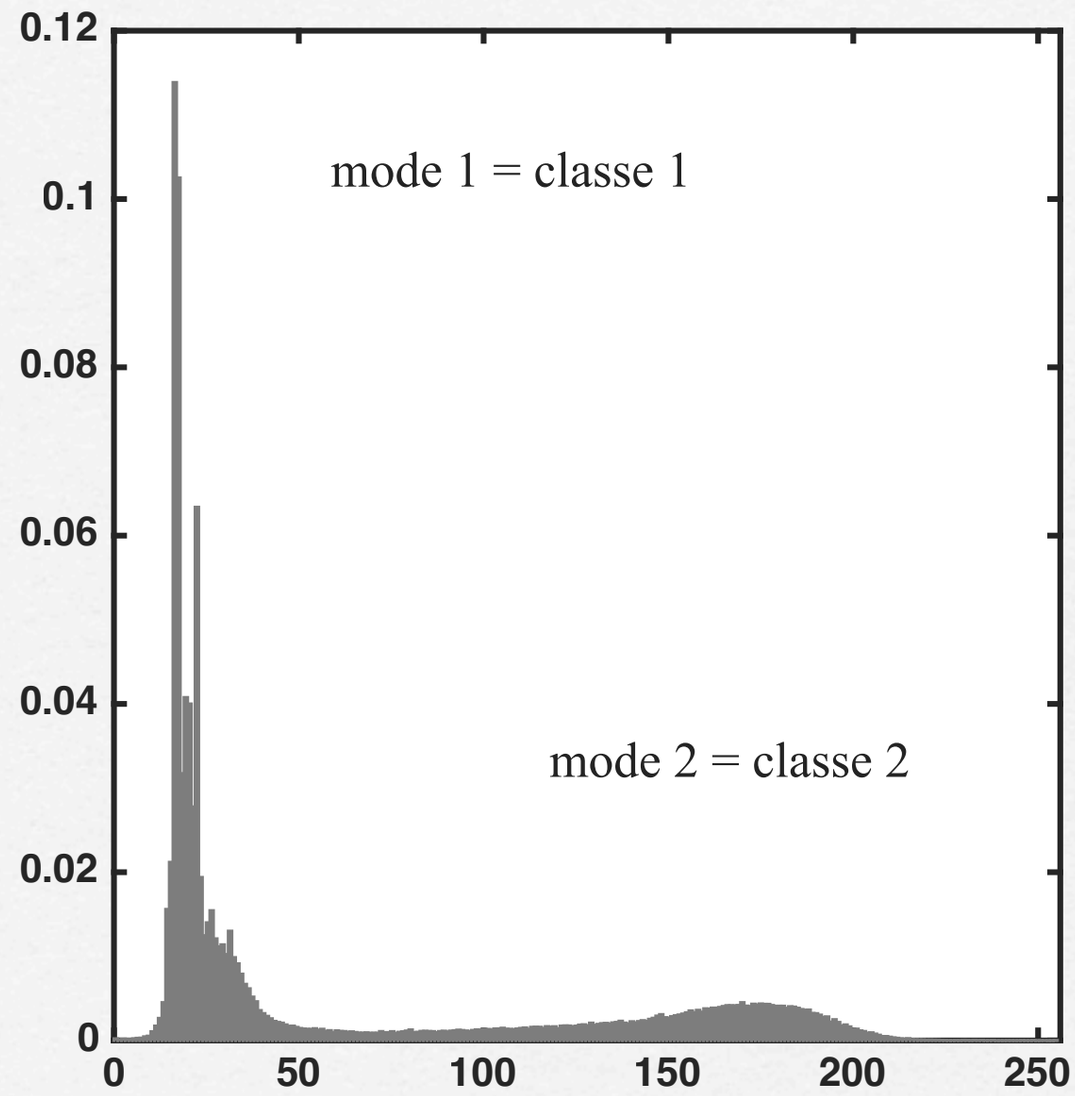
1. Segmentation à base de classification

a) Seuillage

- Problème
  - ✓ Comment déterminer le meilleur seuil automatiquement?

Hypothèses :

- On a une image à 2 modes
- On peut représenter chaque mode par une fonction continue
  - ✓ on peut démontrer que le meilleur seuil est à l'endroit où les deux fonctions se croisent
  - ✓ endroit qui maximise la variabilité entre les 2 modes



Source :

4. RÉGIONS

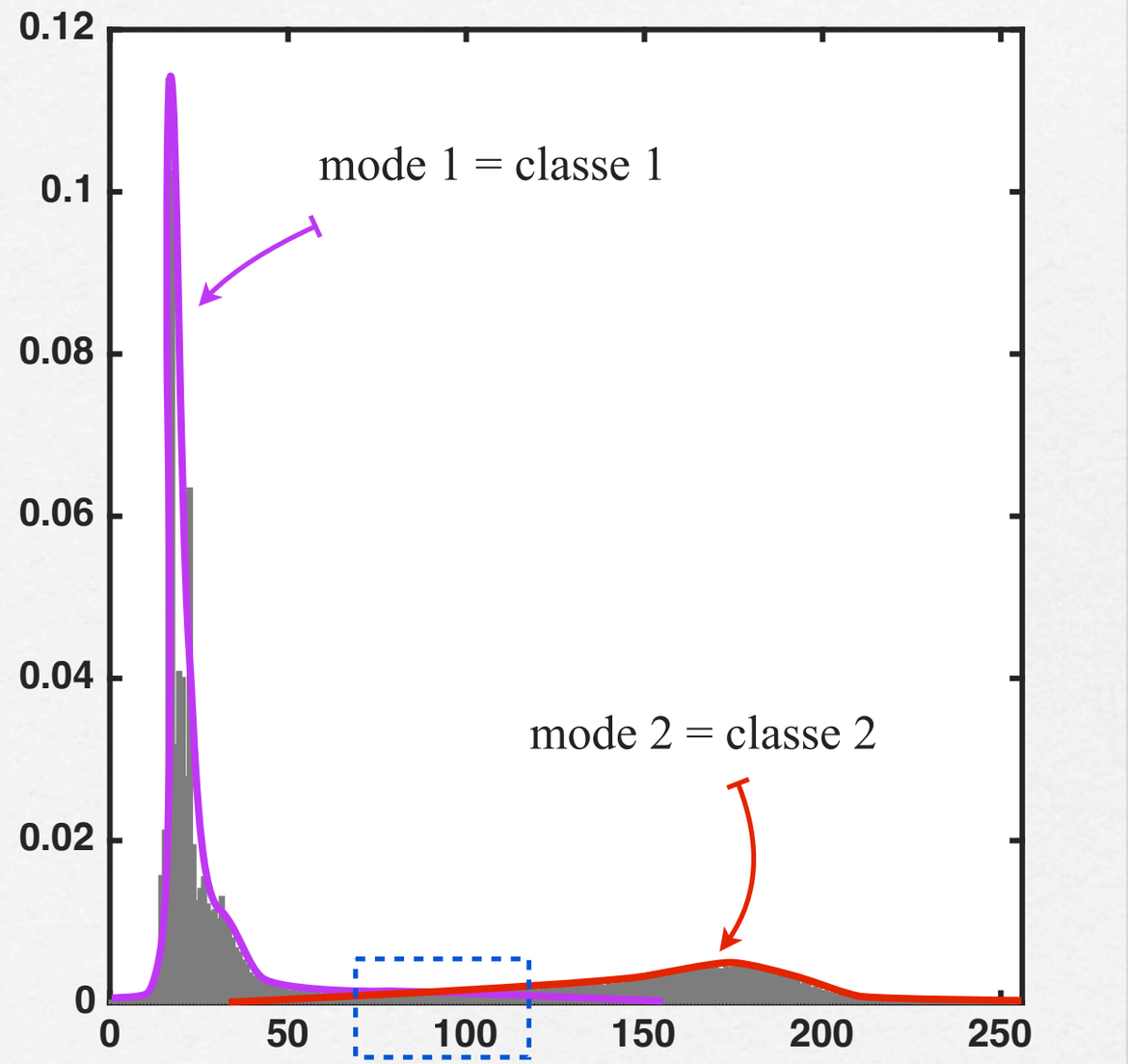
1. Segmentation à base de classification

a) Seuillage

- Problème
  - ✓ Comment déterminer le meilleur seuil automatiquement?

Hypothèses :

- On a une image à 2 modes
- On peut représenter chaque mode par une fonction continue
  - ✓ on peut démontrer que le meilleur seuil est à l'endroit où les deux fonctions se croisent
  - ✓ endroit qui maximise la variabilité entre les 2 modes



Source :

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- 2 classes

Algorithme : Calcul moyennes de classe (deux classes)

Entrées :	$f[m, n]$	→ Image d'origine
	$c[m, n]$	→ Segmentation en deux classes (0 ou 1)
Sorties :	μ_0 et μ_1	→ moyennes des deux classes

Source :

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- 2 classes

Algorithme : Calcul moyennes de classe (deux classes)

Entrées : $f[m, n]$ → Image d'origine

$c[m, n]$ → Segmentation en deux classes (0 ou 1)

Sorties : μ_0 et μ_1 → moyennes des deux classes

1	$\mu_0 = \mu_1 = N_0 = N_1 = 0$

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

- 2 classes

Algorithme : Calcul moyennes de classe (deux classes)

Entrées : $f[m, n]$ → Image d'origine

$c[m, n]$ → Segmentation en deux classes (0 ou 1)

Sorties : μ_0 et μ_1 → moyennes des deux classes

1	$\mu_0 = \mu_1 = N_0 = N_1 = 0$	
2	Pour chaque pixel $[m, n]$	
	Si $c[m, n] = 0$	// classe 0

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

- 2 classes

Algorithme : Calcul moyennes de classe (deux classes)

Entrées : $f[m, n]$ → Image d'origine

$c[m, n]$ → Segmentation en deux classes (0 ou 1)

Sorties : μ_0 et μ_1 → moyennes des deux classes

1	$\mu_0 = \mu_1 = N_0 = N_1 = 0$	
2	Pour chaque pixel $[m, n]$	
	Si $c[m, n] = 0$	// classe 0
	μ_0 += $f[m, n]$	
	N_0 += 1	

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

- 2 classes

Algorithme : Calcul moyennes de classe (deux classes)

Entrées : $f[m, n]$ → Image d'origine

$c[m, n]$ → Segmentation en deux classes (0 ou 1)

Sorties : μ_0 et μ_1 → moyennes des deux classes

1	$\mu_0 = \mu_1 = N_0 = N_1 = 0$	
2	Pour chaque pixel $[m, n]$	
	Si $c[m, n] = 0$	// classe 0
	μ_0 += $f[m, n]$	
	N_0 += 1	
	Sinon	
	μ_1 += $f[m, n]$	
	N_1 += 1	
3	μ_0 /= N_0 et μ_1 /= N_1	

Source :

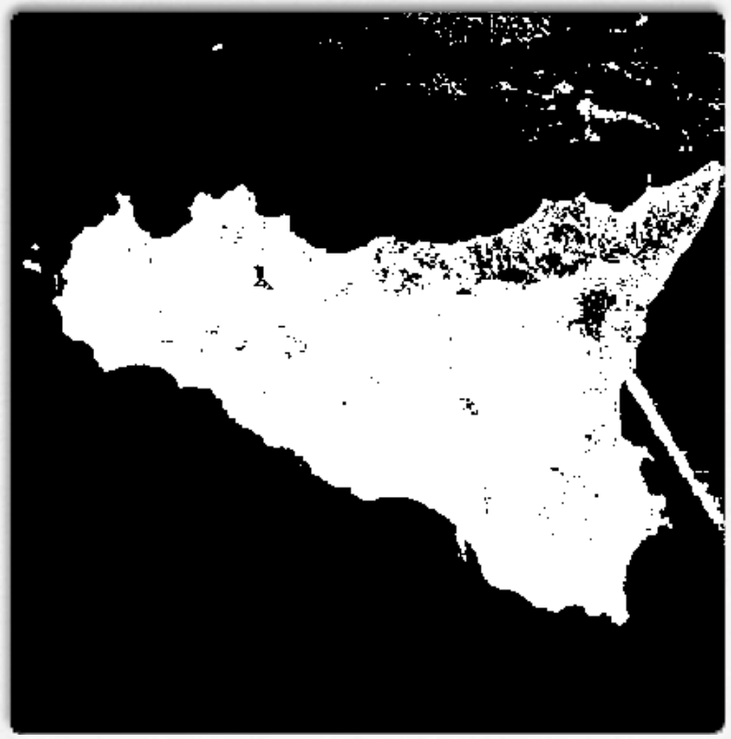
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- 2 classes

Image segmentée



Source :

©Marie-Flavie Auclair-Fortier

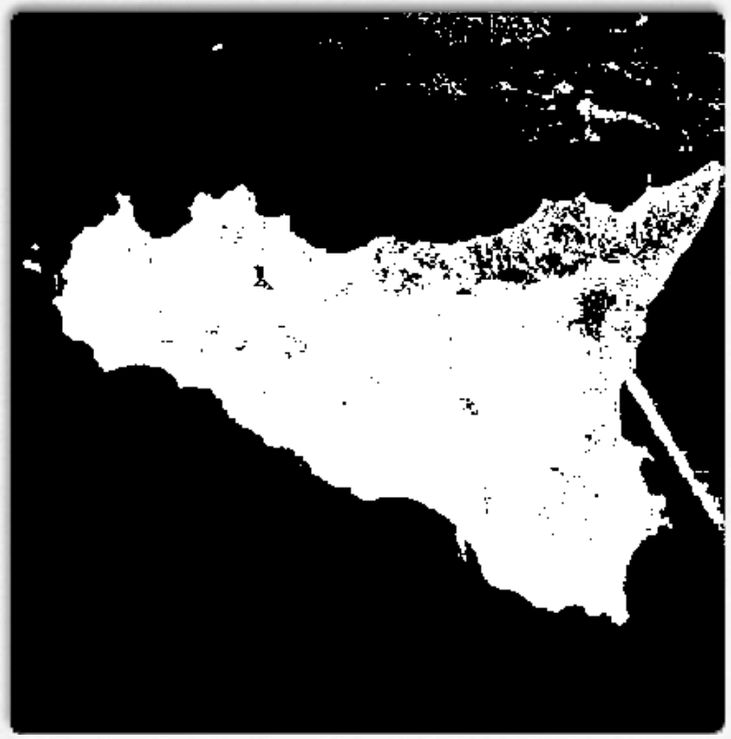
4. RÉGIONS

1. Segmentation à base de classification

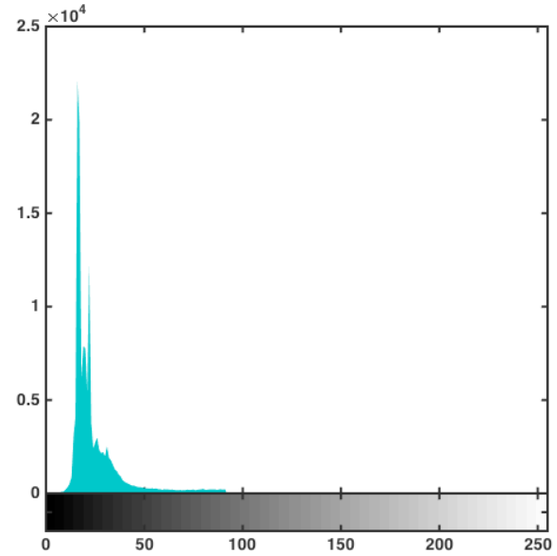
b) *K*-moyennes

- 2 classes

Image segmentée



Pixels de la classe 0



Source :

©Marie-Flavie Auclair-Fortier

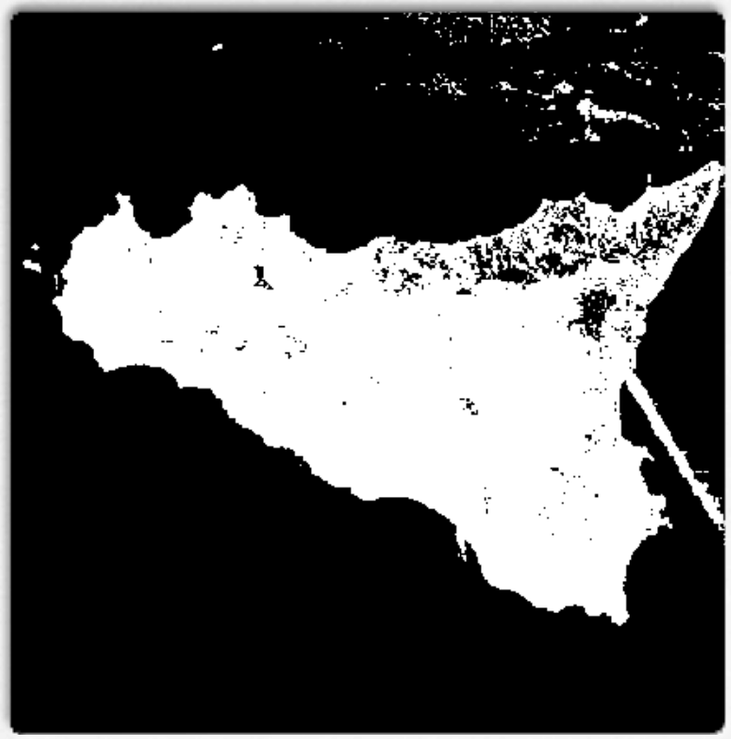
4. RÉGIONS

1. Segmentation à base de classification

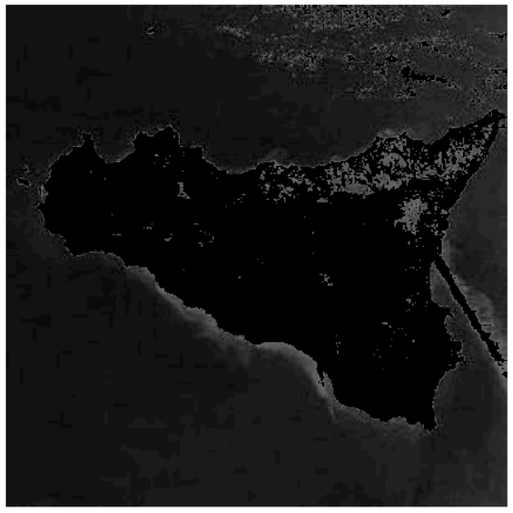
b) *K*-moyennes

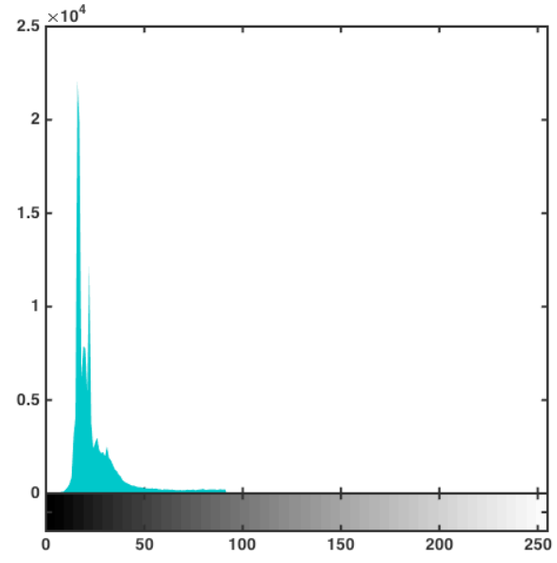
- 2 classes

Image segmentée

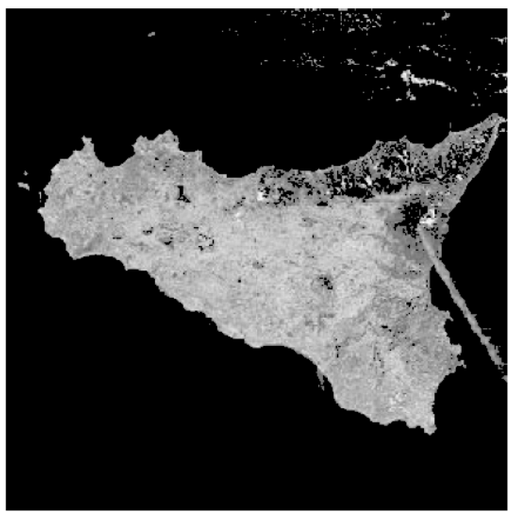


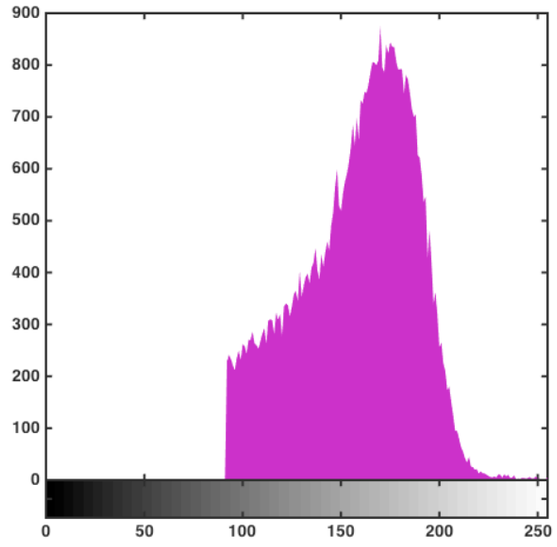
Pixels de la classe 0





Pixels de la classe 1





Source :

©Marie-Flavie Auclair-Fortier

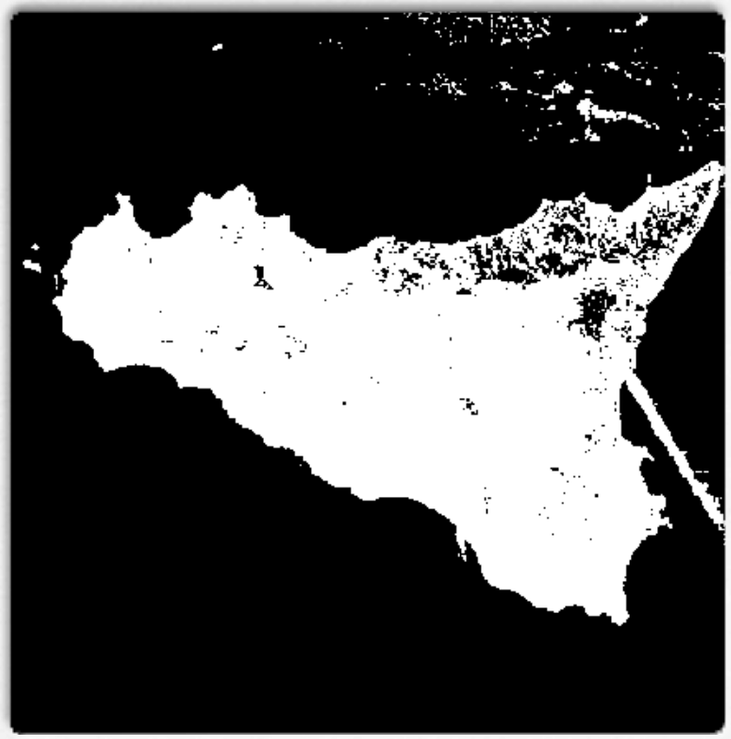
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

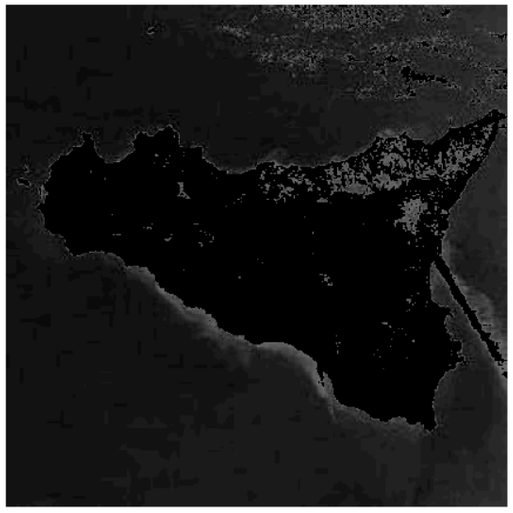
- 2 classes

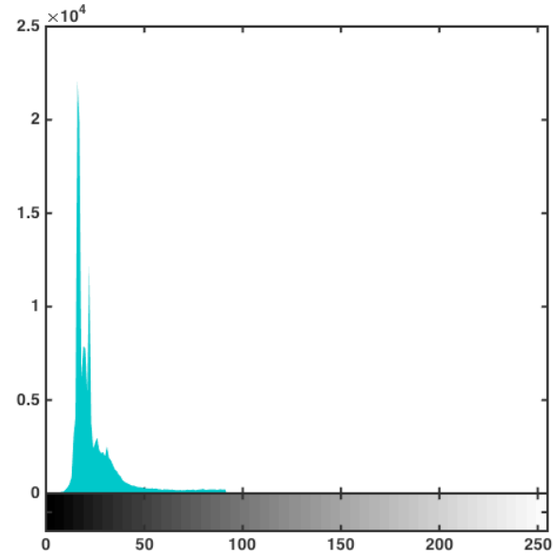
Image segmentée



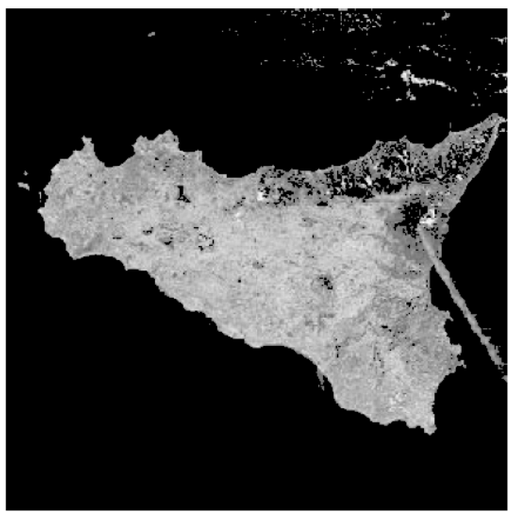
Classe i	moyenne μ_i
0	24.71
1	157.83

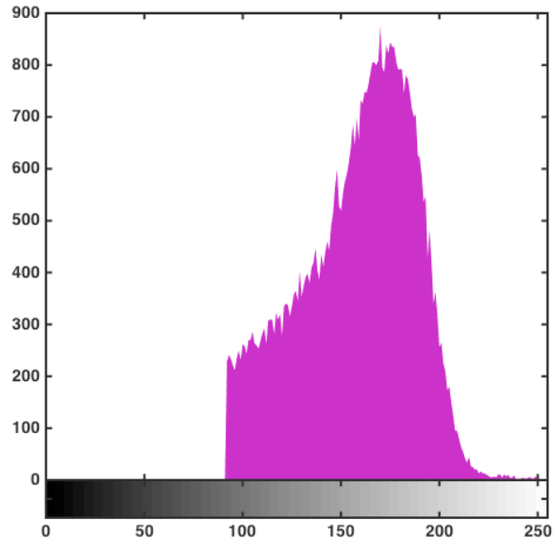
Pixels de la classe 0





Pixels de la classe 1





Source :

©Marie-Flavie Auclair-Fortier

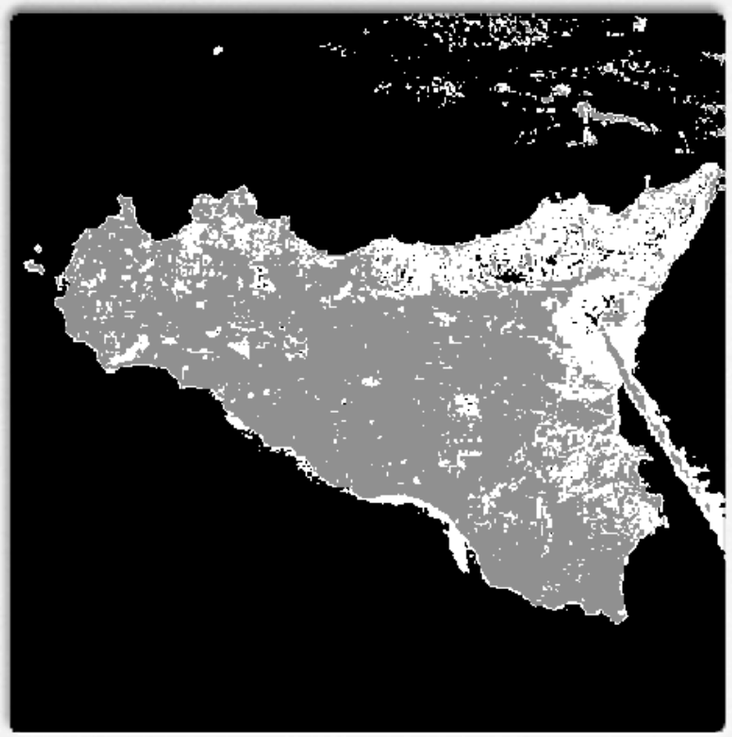
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- 3 classes

Image segmentée



Source :

©Marie-Flavie Auclair-Fortier

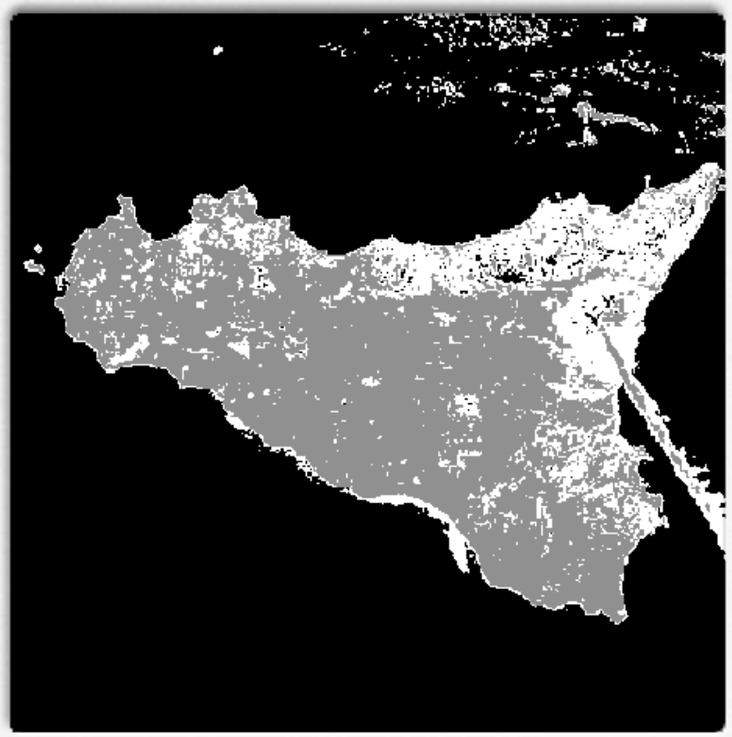
4. RÉGIONS

1. Segmentation à base de classification

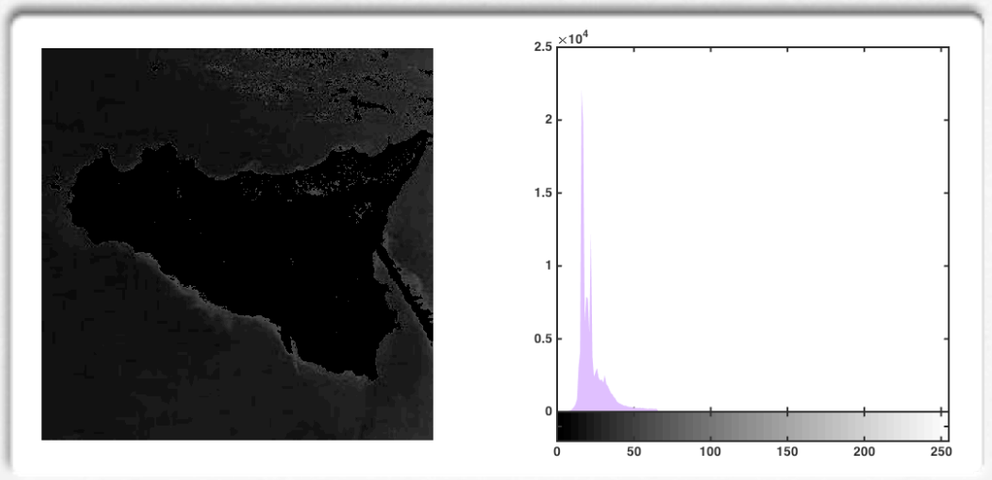
b) *K*-moyennes

- 3 classes

Image segmentée



Pixels de la classe 0



Source :

©Marie-Flavie Auclair-Fortier

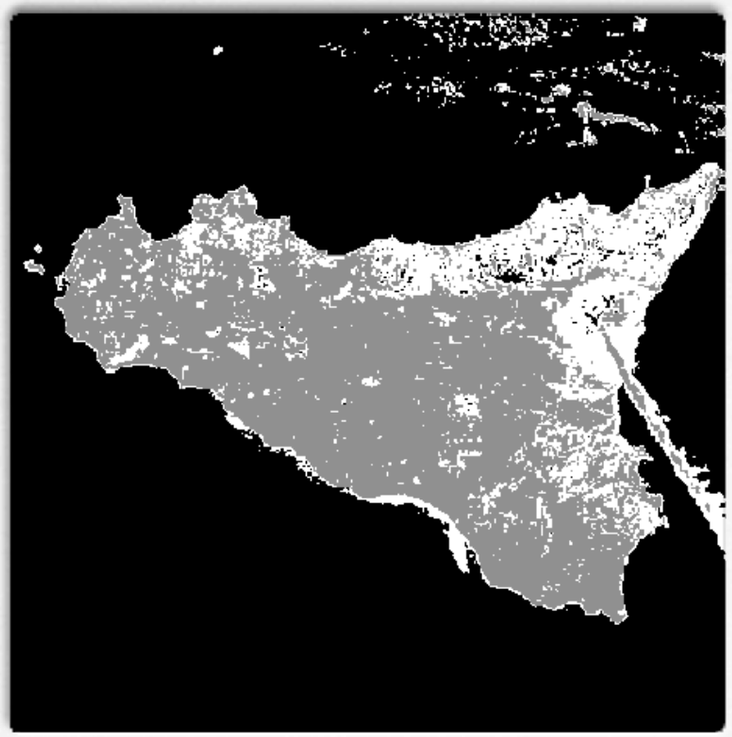
4. RÉGIONS

1. Segmentation à base de classification

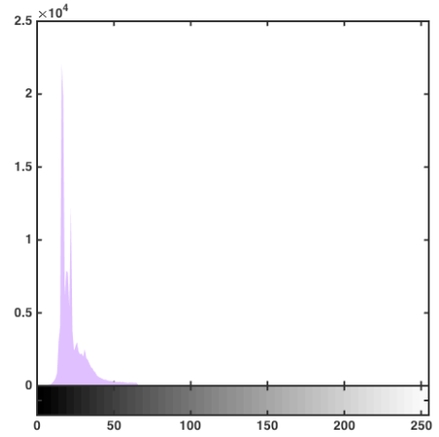
b) *K*-moyennes

- 3 classes

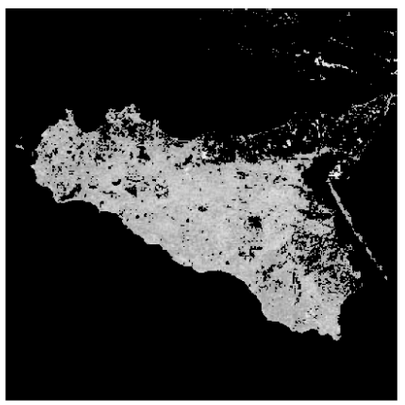
Image segmentée

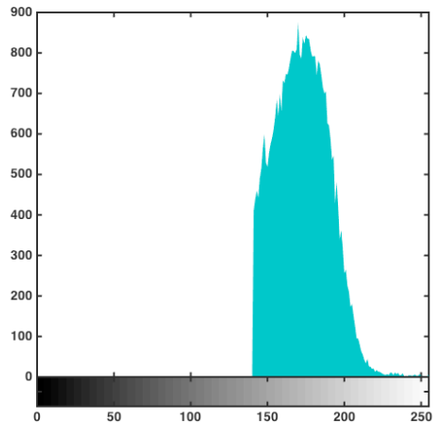


Pixels de la classe 0



Pixels de la classe 1





Source :

©Marie-Flavie Auclair-Fortier

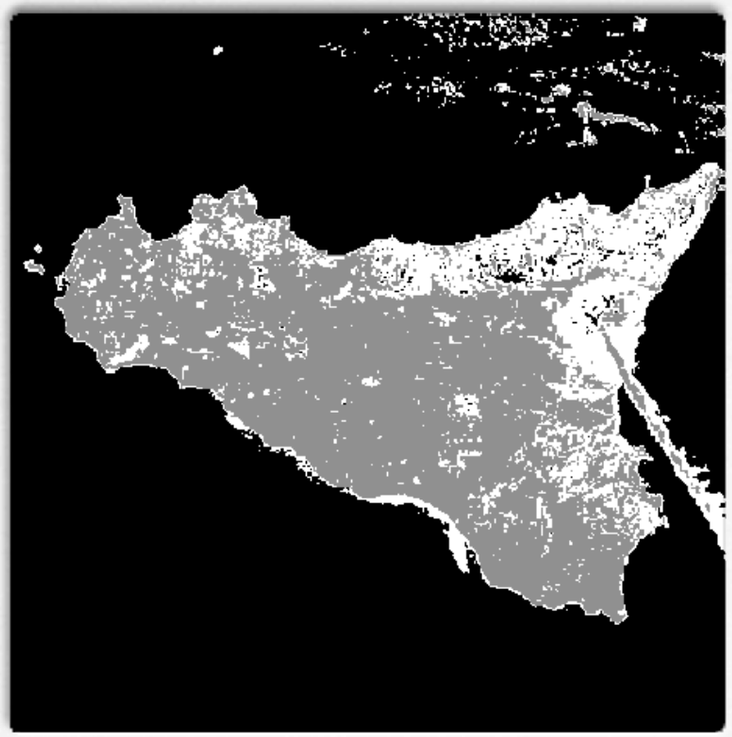
4. RÉGIONS

1. Segmentation à base de classification

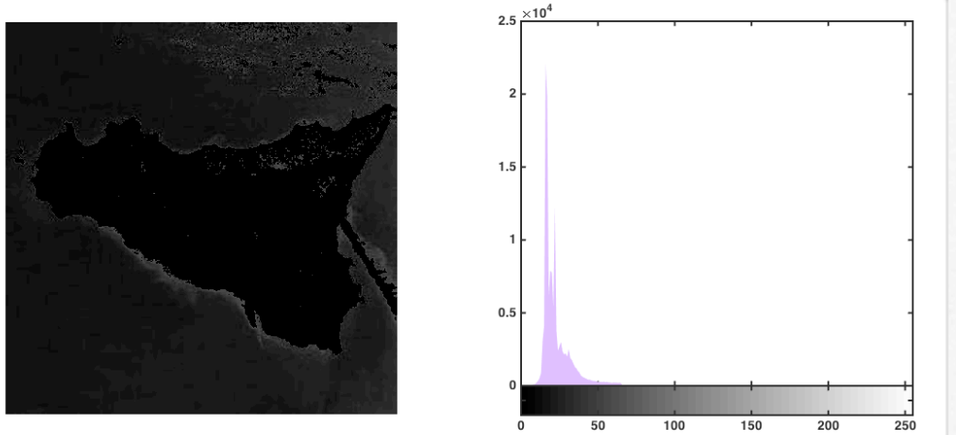
b) *K*-moyennes

- 3 classes

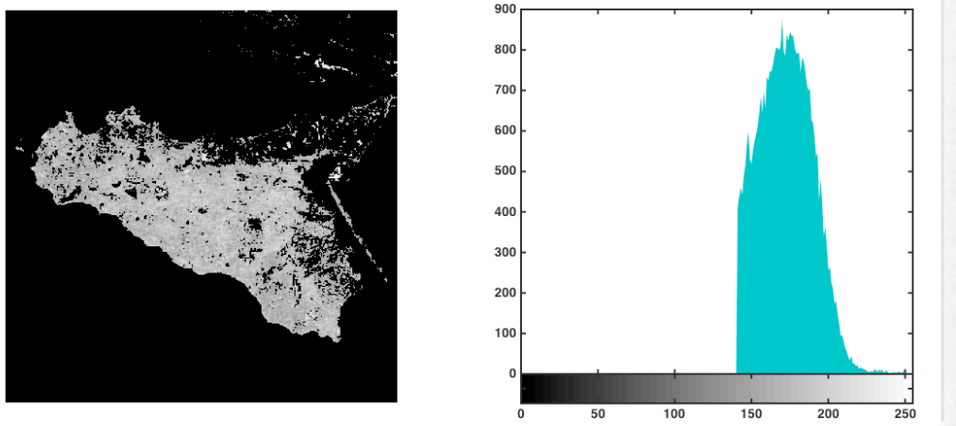
Image segmentée



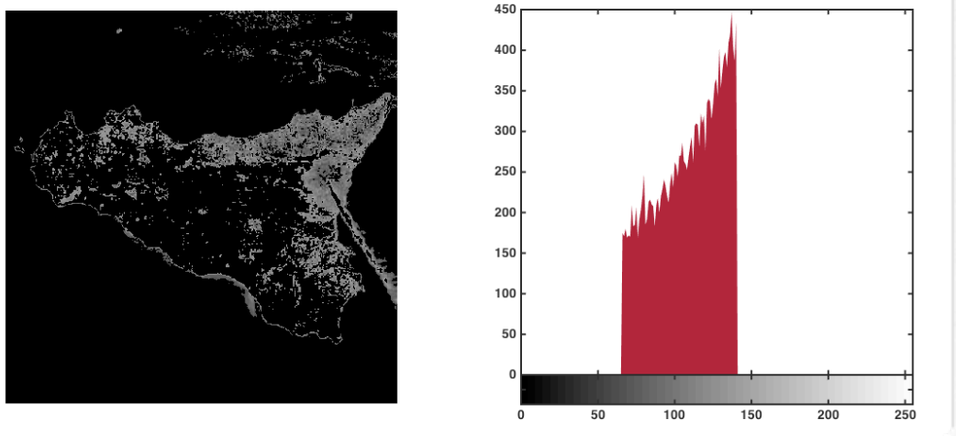
Pixels de la classe 0



Pixels de la classe 1



Pixels de la classe 2



Source :

©Marie-Flavie Auclair-Fortier

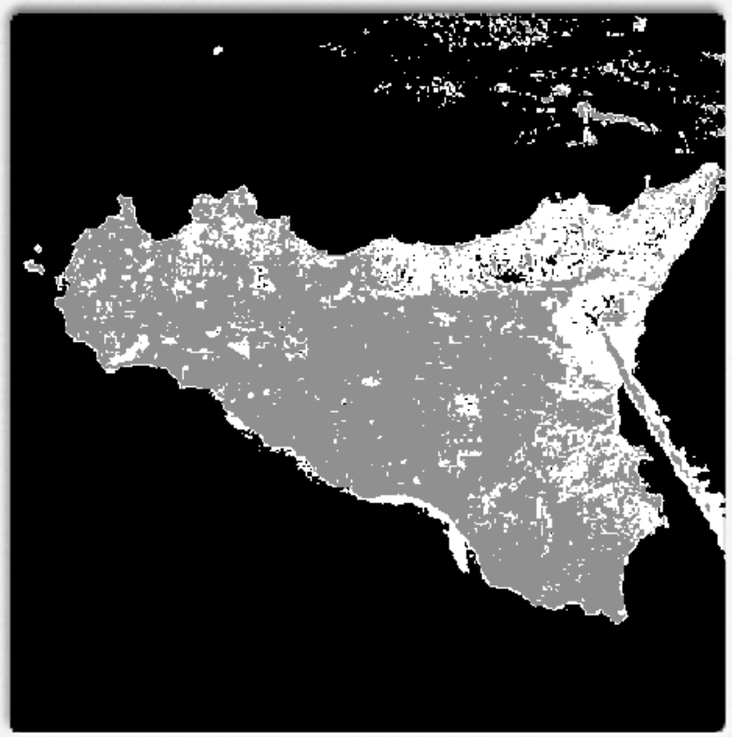
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

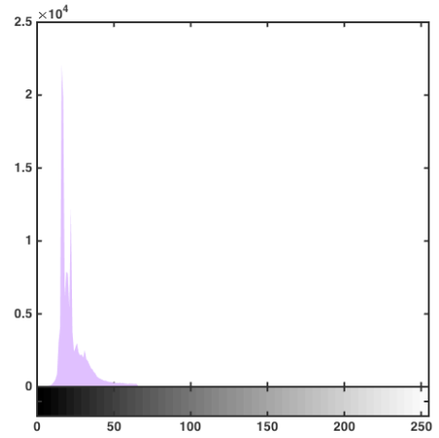
- 3 classes

Image segmentée

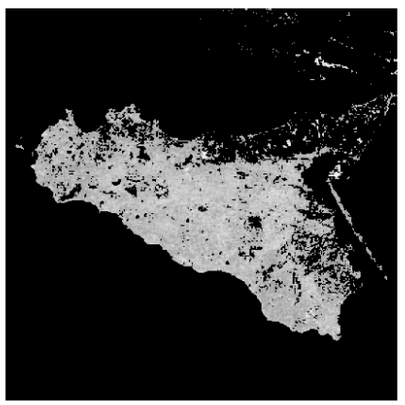


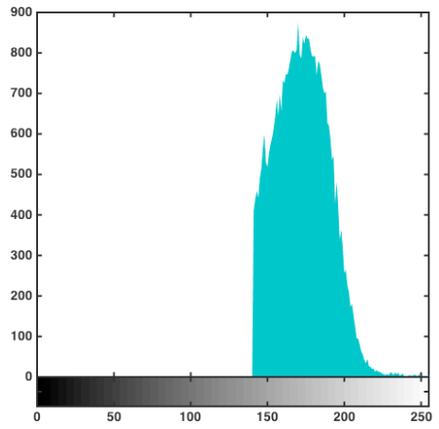
Classe i	moyenne μ_i
0	22.61
1	172.41
2	108.76

Pixels de la classe 0

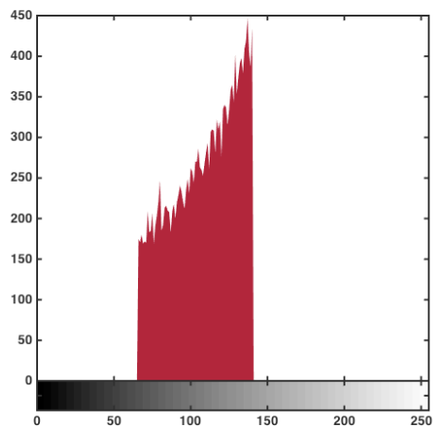


Pixels de la classe 1





Pixels de la classe 2



Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- 2 classes

Algorithme : Classification par la moyenne (2 classes)

Entrées : $f[m, n]$ → Image d'origine

μ_0 et μ_1 → moyennes des deux classes

Sorties : $c[m, n]$ → Segmentation en deux classes (0 ou 1)

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

- 2 classes

Algorithme : Classification par la moyenne (2 classes)

Entrées : $f[m, n]$ → Image d'origine

μ_0 et μ_1 → moyennes des deux classes

Sorties : $c[m, n]$ → Segmentation en deux classes (0 ou 1)

1	`Pour chaque pixel` $[m, n]$

Source :

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

- 2 classes

Algorithme : Classification par la moyenne (2 classes)

Entrées : $f[m, n]$ → Image d'origine

μ_0 et μ_1 → moyennes des deux classes

Sorties : $c[m, n]$ → Segmentation en deux classes (0 ou 1)

1	Pour chaque pixel $[m, n]$
	Si $(f[m, n] - \mu_0)^2 < (f[m, n] - \mu_1)^2$
	$c[m, n] = 0$

Source :

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- 2 classes

Algorithme : Classification par la moyenne (2 classes)

Entrées : $f[m, n]$ → Image d'origine

μ_0 et μ_1 → moyennes des deux classes

Sorties : $c[m, n]$ → Segmentation en deux classes (0 ou 1)

1	Pour chaque pixel $[m, n]$
	Si $(f[m, n] - \mu_0)^2 < (f[m, n] - \mu_1)^2$
	$c[m, n] = 0$
	Sinon

Source :

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- 2 classes

Algorithme : Classification par la moyenne (2 classes)

Entrées : $f[m, n]$ → Image d'origine

μ_0 et μ_1 → moyennes des deux classes

Sorties : $c[m, n]$ → Segmentation en deux classes (0 ou 1)

```
1   Pour chaque pixel [m, n]
        Si (f[m, n] − μ0)^2 < (f[m, n] − μ1)^2
            c[m, n] = 0
        Sinon
            c[m, n] = 1
```

Source :

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

Algorithme 4.9 : Classification par les k-moyennes

Entrées :	$f[m, n]$	→ Image d'origine
	k	→ Nombre de classes
Sorties :	$r[m, n]$	→ Carte des k classes

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

Algorithme 4.9 : Classification par les k-moyennes

Entrées :	$f[m, n]$	→ Image d'origine
	k	→ Nombre de classes
Sorties :	$r[m, n]$	→ Carte des k classes

1	`Pour toutes les classes` $i \in \{1, \ldots, k\}$

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

Algorithme 4.9 : Classification par les k-moyennes

Entrées : $f[m, n]$ → Image d'origine

k → Nombre de classes

Sorties : $r[m, n]$ → Carte des k classes

1	Pour toutes les classes $i \in \{1, \ldots, k\}$
	Initialiser la moyenne de la classe (peut être au hasard ou supervisé)
2	Tant que les moyennes changent

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

Algorithme 4.9 : Classification par les k-moyennes

Entrées : $f[m, n]$ → Image d'origine

k → Nombre de classes

Sorties : $r[m, n]$ → Carte des k classes

1	Pour toutes les classes $i \in \{1, \ldots, k\}$
	Initialiser la moyenne de la classe (peut être au hasard ou supervisé)
2	Tant que les moyennes changent
	Pour chaque pixel $[m, n]$

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) K-moyennes

Algorithme 4.9 : Classification par les k-moyennes

Entrées : $f[m, n]$ → Image d'origine

k → Nombre de classes

Sorties : $r[m, n]$ → Carte des k classes

1	Pour toutes les classes $i \in \{1, \ldots, k\}$	
	Initialiser la moyenne de la classe (peut être au hasard ou supervisé)	
2	Tant que les moyennes changent	
	Pour chaque pixel $[m, n]$	
	Assigner le pixel à la classe i telle que $i = \text{argmin}\,[(f[m, n] - \mu_i)^2]$	// le $i \in \{1, \ldots, k\}$ pour lequel l'erreur quadratique est minimale
3	Recalculer la moyenne de chaque classe	

Source :

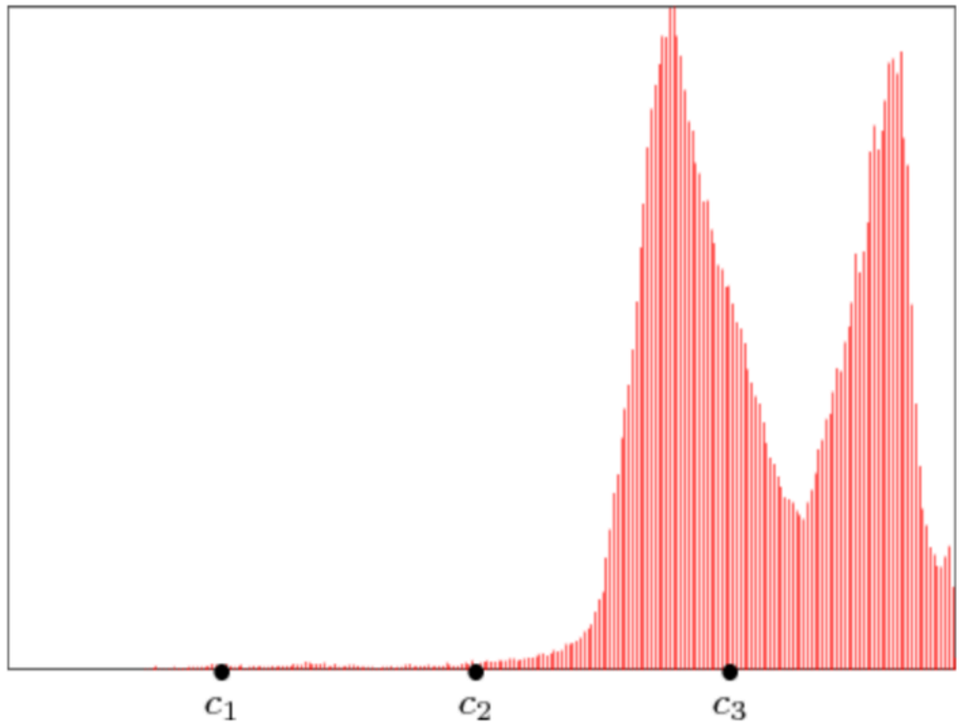
©Marie-Flavie Auclair-Fortier

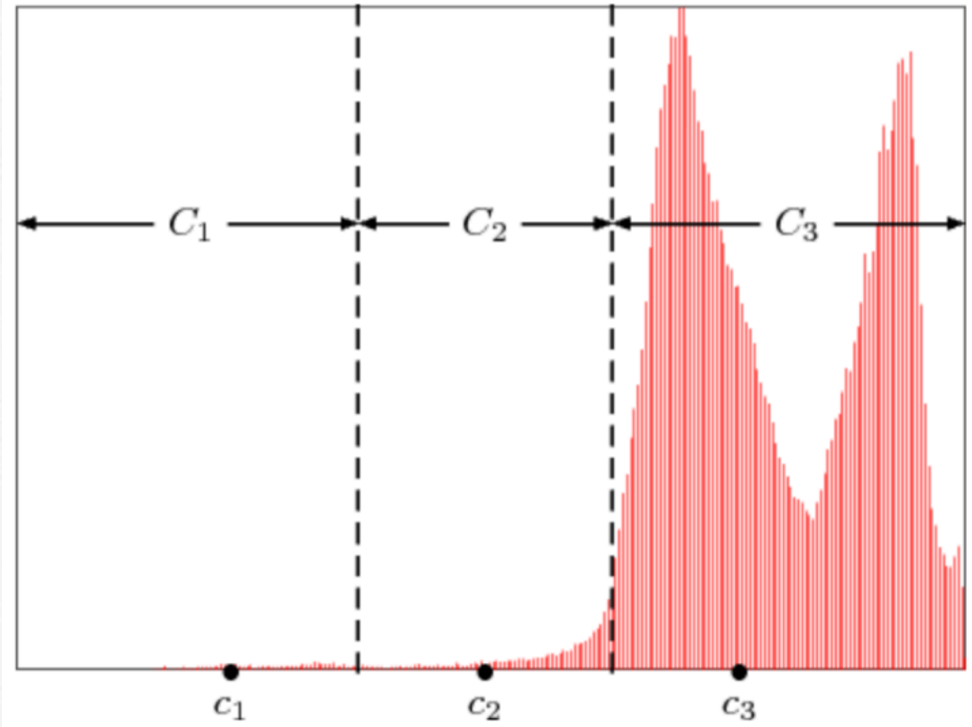
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- Initialisation des centroïdes, égale répartition





Source : Anne Vialard LaBRI, Université Bordeaux

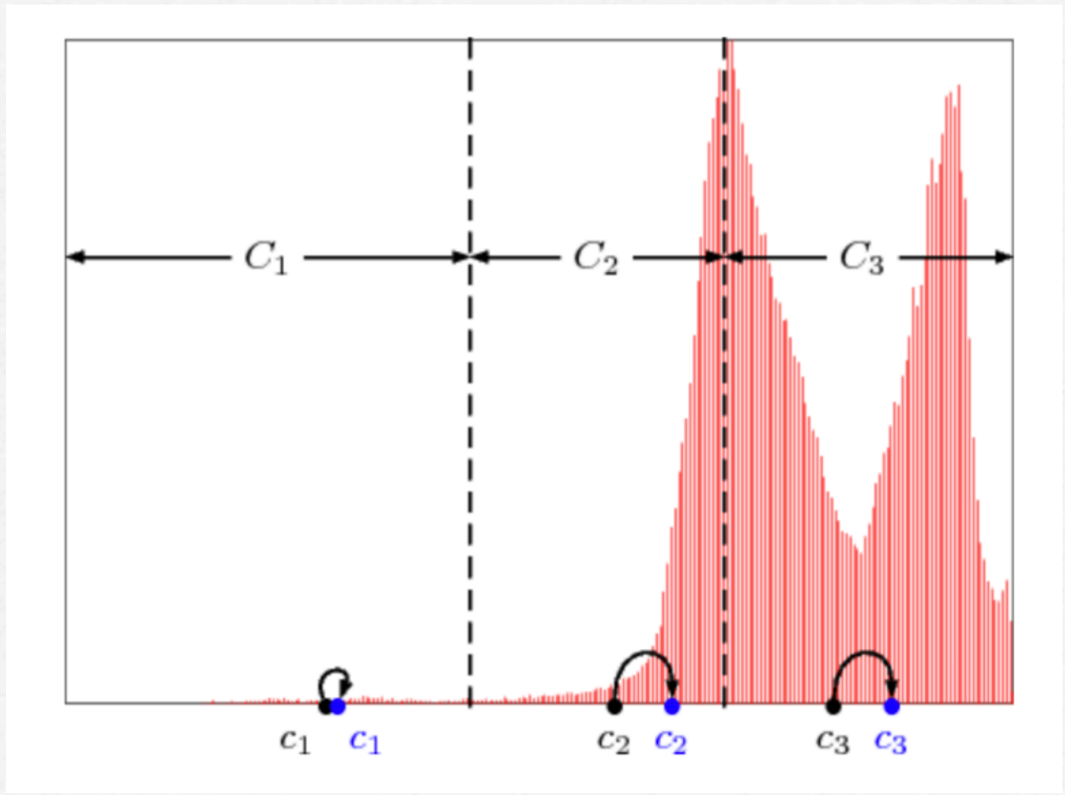
©Marie-Flavie Auclair-Fortier

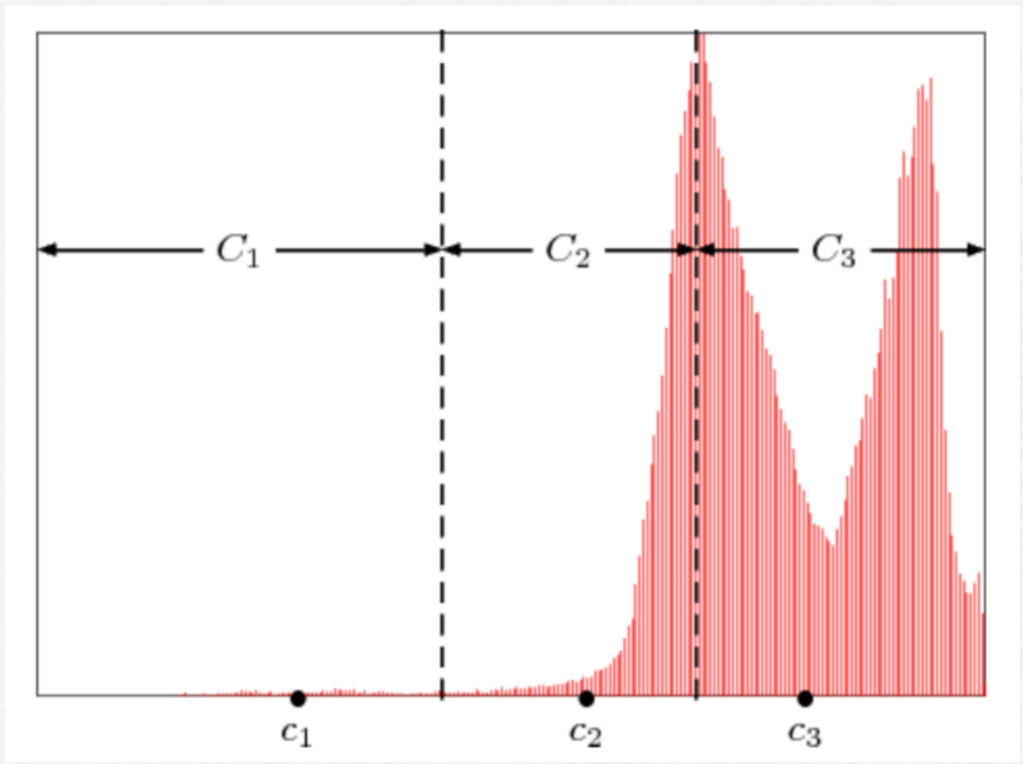
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- Itération 1



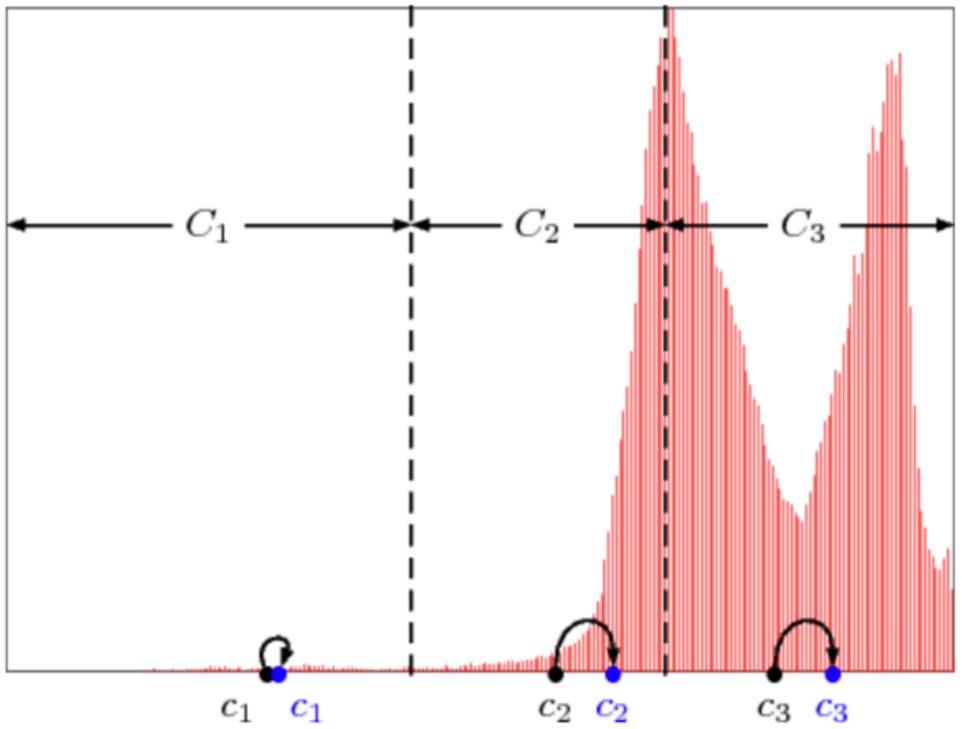


Source : Anne Vialard LaBRI, Université Bordeaux

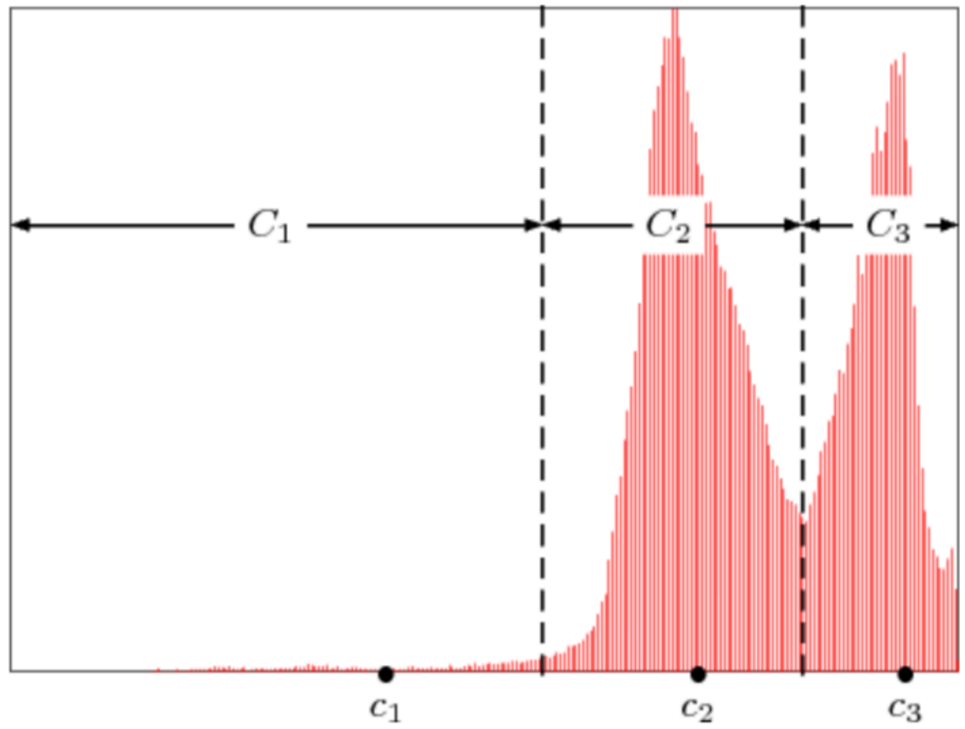
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes



Itération 2



Segmentation finale

Source : Anne Vialard LaBRI, Université Bordeaux

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ $f[m, n]$ peut être une fonction vectorielle

Remarque

- la moyenne est utilisée comme couleur représentative de la région

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ $f[m,n]$ peut être une fonction vectorielle
    - ➡ $f[m,n]$ est alors un vecteur de caractéristiques

Remarque

- la moyenne est utilisée comme couleur représentative de la région

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ $f[m,n]$ peut être une fonction vectorielle
    - ➡ $f[m,n]$ est alors un vecteur de caractéristiques
      - *Exemple* : $f[m,n] = (R,G,B)[m,n]$

Remarque

- la moyenne est utilisée comme couleur représentative de la région

Source :

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ $f[m,n]$ peut être une fonction vectorielle
    - ➡ $f[m,n]$ est alors un vecteur de caractéristiques
      - *Exemple* : $f[m,n] = (R,G,B)[m,n]$
    - ➡ Les μ_i sont aussi des vecteurs (centroïdes)

Remarque

- la moyenne est utilisée comme couleur représentative de la région

Source :

4. RÉGIONS

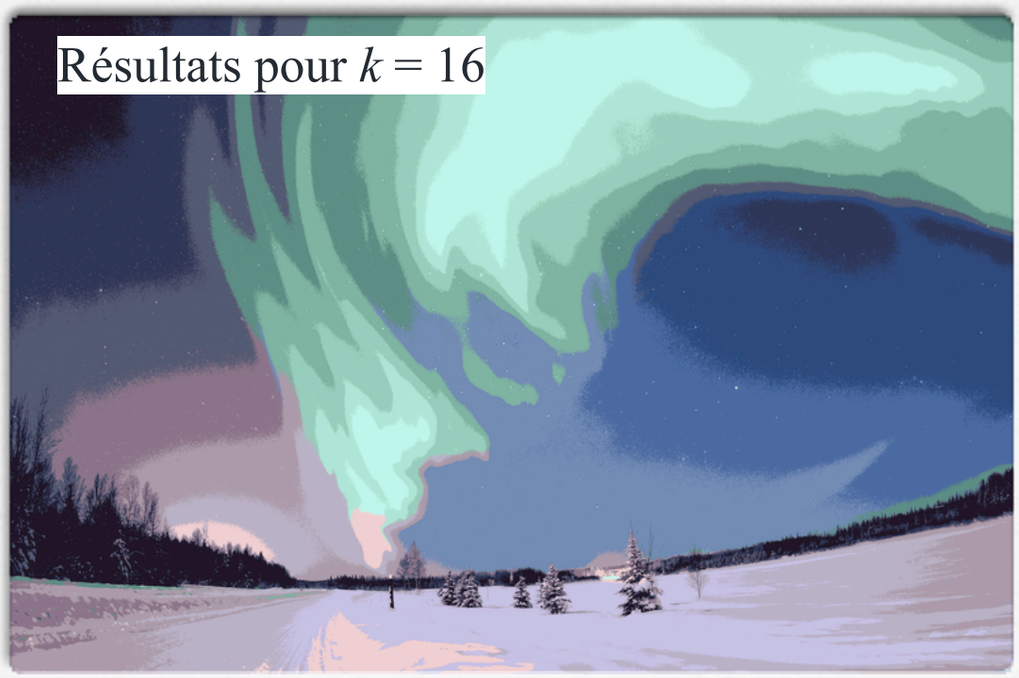
1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ $f[m,n]$ peut être une fonction vectorielle
    - ➡ $f[m,n]$ est alors un vecteur de caractéristiques
      - *Exemple* : $f[m,n] = (R,G,B)[m,n]$
    - ➡ Les μ_i sont aussi des vecteurs (centroïdes)
    - ➡ Dans ce cas, on assigne $i = \text{argmin} \left| f[m,n] - \mu_i \right|^2$

Remarque

- la moyenne est utilisée comme couleur représentative de la région



Source :

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ On peut également utiliser d'autres caractéristiques pour segmenter une image
    - ➡ Les caractéristiques liées à la texture

Source : PM Jodoin

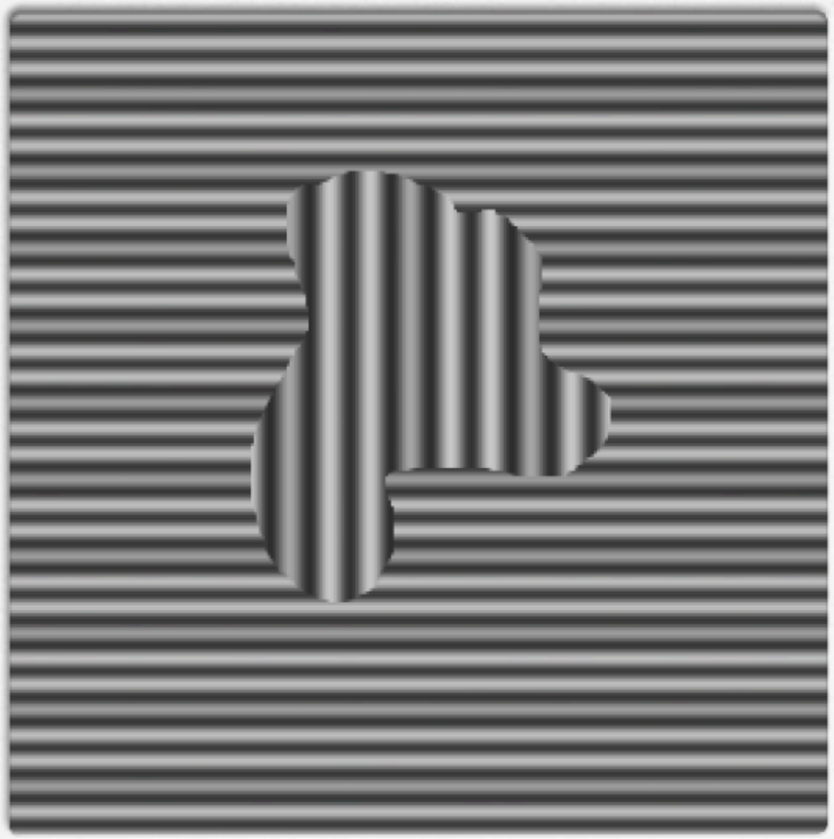
©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ On peut également utiliser d'autres caractéristiques pour segmenter une image
    - ➡ Les caractéristiques liées à la texture



Exemple :

- Il est impossible de segmenter cette image en 2 classes sur la base exclusive des niveaux de gris car l'objet central possède le même histogramme que le reste de l'image.

Source : PM Jodoin

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

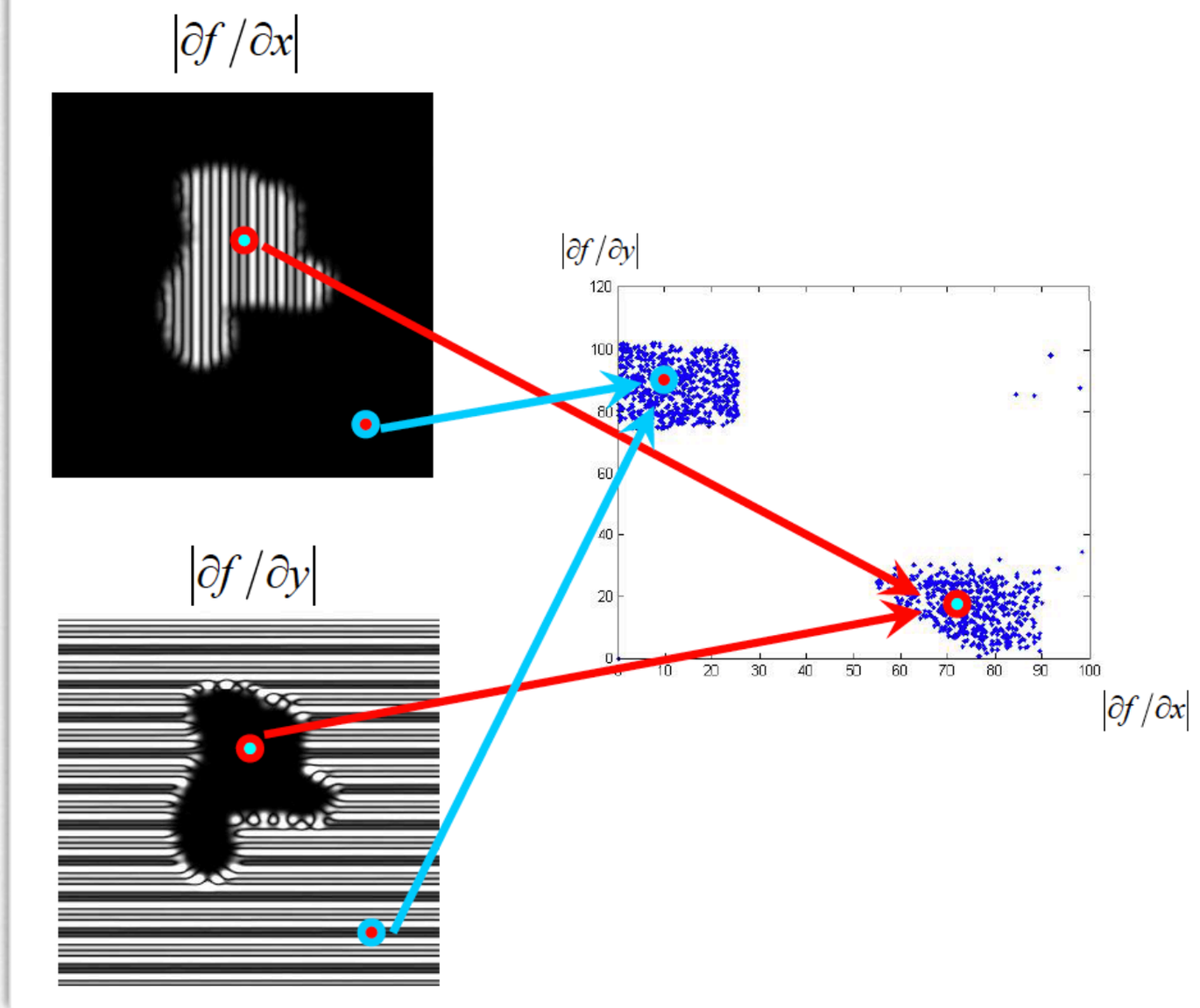
1. Segmentation à base de classification

b) *K*-moyennes

- *Remarques* :
  - ✓ On peut également utiliser d'autres caractéristiques pour segmenter une image
    - ➡ Les caractéristiques liées à la texture

Exemple :

- Utiliser un vecteur de caractéristiques formé des dérivées partielles



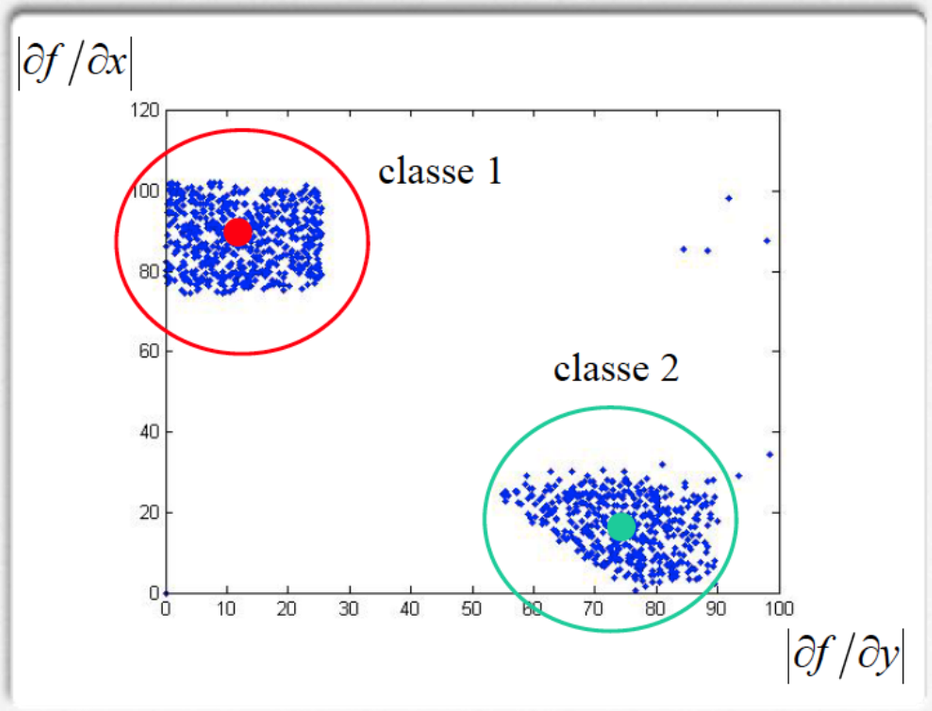
Source : PM Jodoin

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarques*
  - ✓ On peut également utiliser d'autres caractéristiques pour segmenter une image
    - ➡ Les caractéristiques liées à la texture.



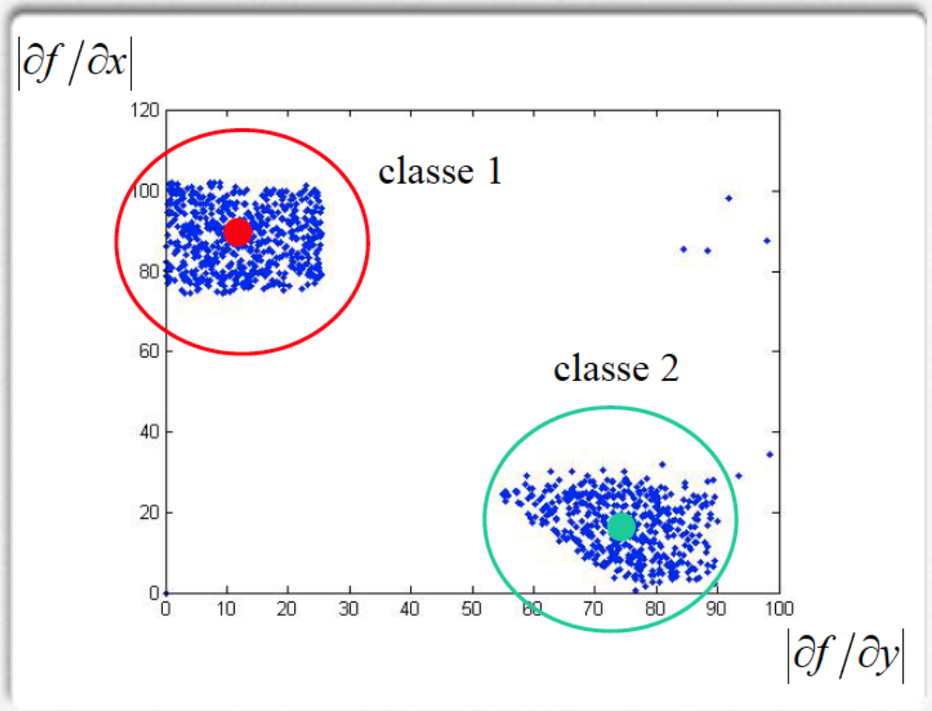
Source : PM Jodoin

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarques*
  - ✓ On peut également utiliser d'autres caractéristiques pour segmenter une image
    - ➡ Les caractéristiques liées à la texture.



Exemple :

- Maintenant que chaque pixel est associé à un point 2D, il est facile de segmenter l'image à l'aide de l'algorithme des *K*-moyennes.

Source : PM Jodoin

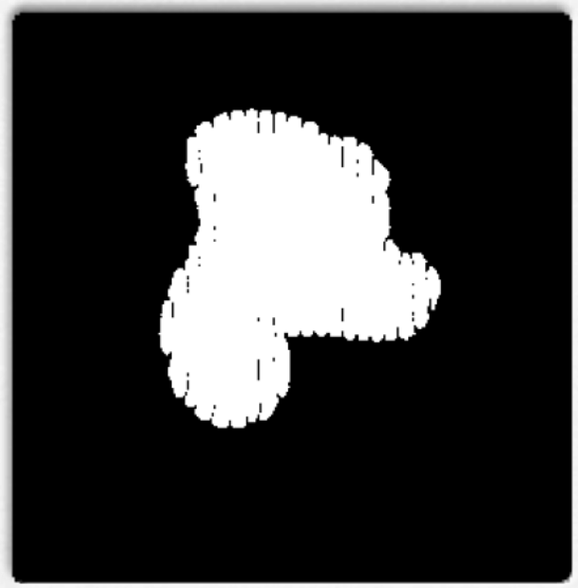
©Marie-Flavie Auclair-Fortier

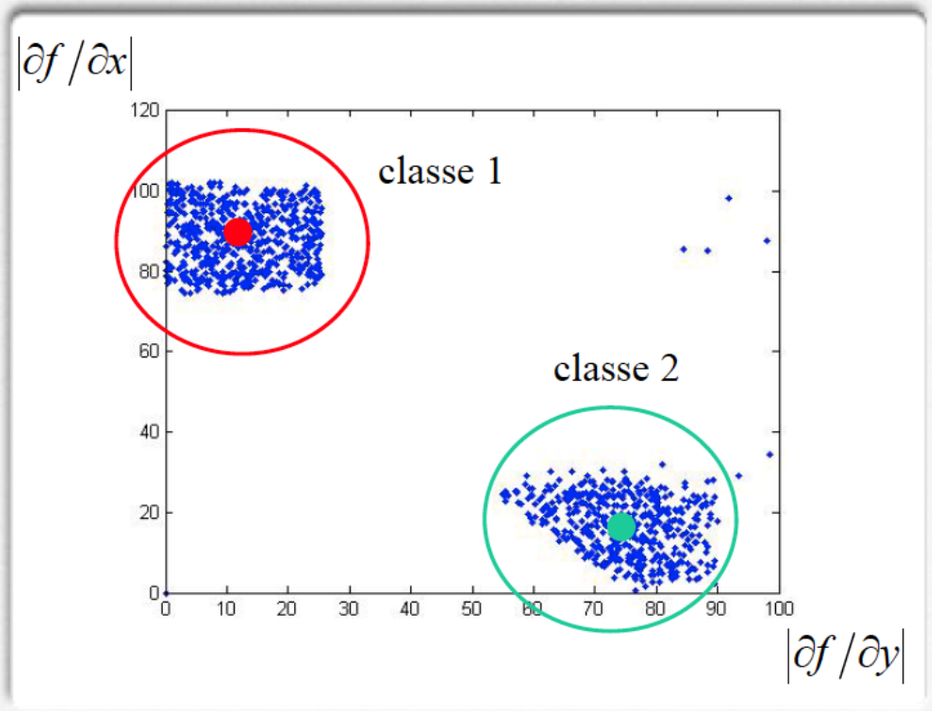
4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarques*
  - ✓ On peut également utiliser d'autres caractéristiques pour segmenter une image
    - ➡ Les caractéristiques liées à la texture.





Exemple :

- Maintenant que chaque pixel est associé à un point 2D, il est facile de segmenter l'image à l'aide de l'algorithme des *K*-moyennes.

Source : PM Jodoin

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ Il n'est pas assuré que cet algorithme converge vers la meilleure solution

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ Il n'est pas assuré que cet algorithme converge vers la meilleure solution
    - ➡ dépendant de l'initialisation, l'algorithme peut converger vers des résultats différents, voire même aberrants. Pour résoudre ce problème, on peut lancer l'algorithme plusieurs fois avec différents paramètres de départ et ne garder que la meilleure solution.

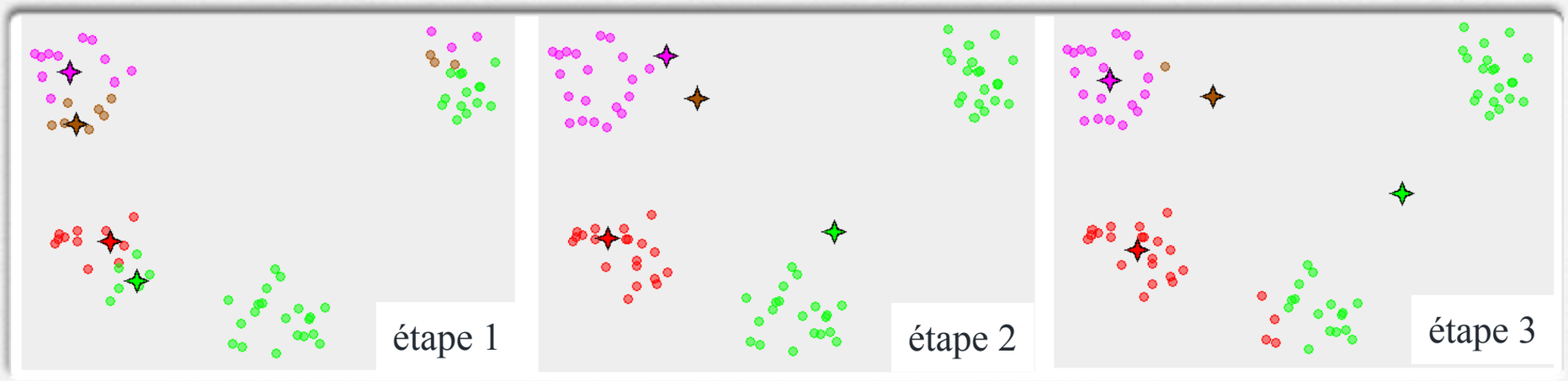
Source :

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ Il n'est pas assuré que cet algorithme converge vers la meilleure solution



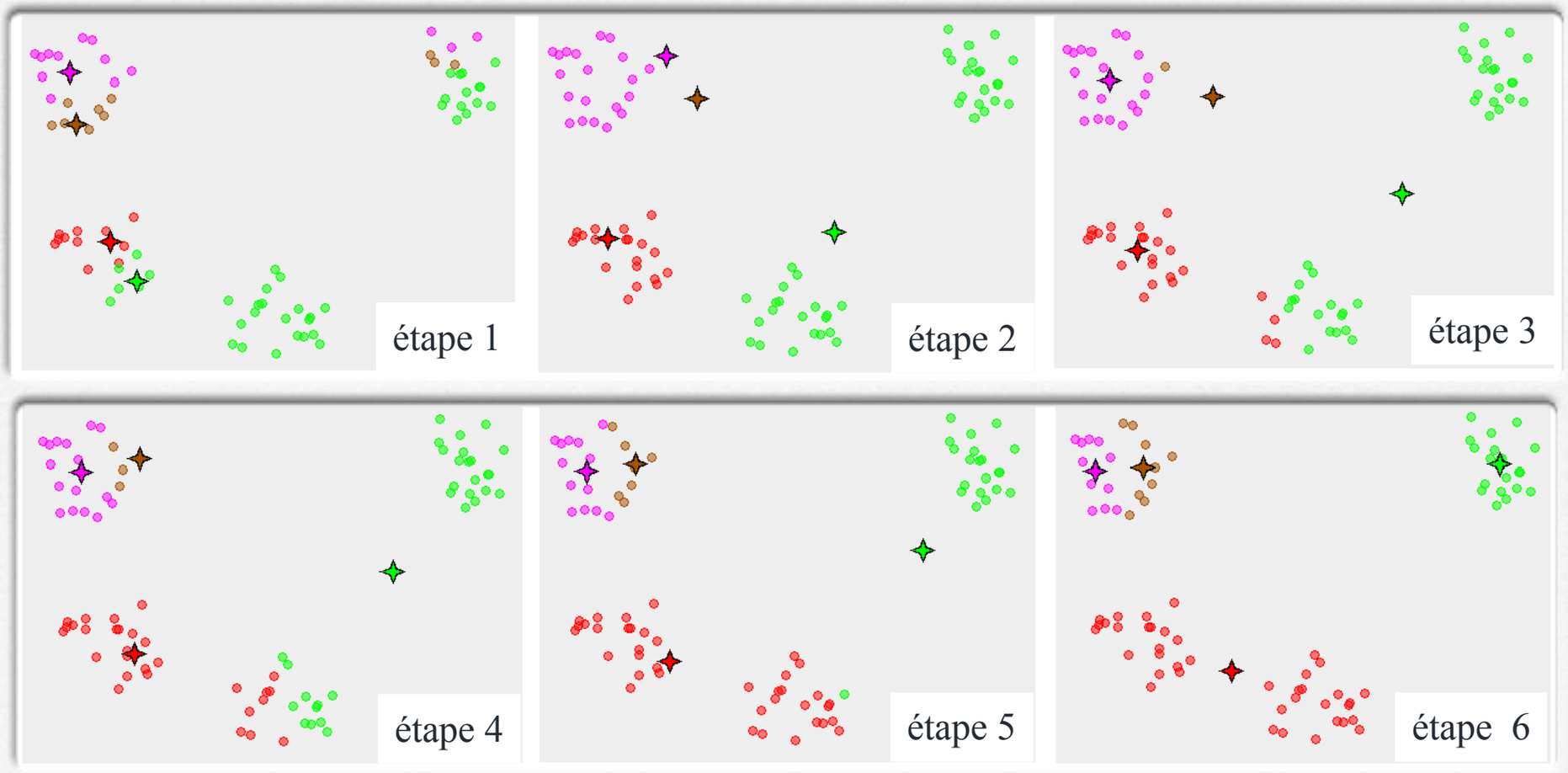
Tiré des notes du cours : ECE662: Statistical Pattern Recognition and Decision Making Processes, M. Boutin, 2008

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ Il n'est pas assuré que cet algorithme converge vers la meilleure solution



Tiré des notes du cours : ECE662: Statistical Pattern Recognition and Decision Making Processes, M. Boutin, 2008

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ Il faut connaître le bon nombre de classes :
    - ➡ Il existe des algorithmes d'apprentissage de k

Image originale | $k = 2$ | $k = 3$ | $k = 10$

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

b) *K*-moyennes

- *Remarque* :
  - ✓ Il faut connaître le bon nombre de classes :
    - ➡ Il existe des algorithmes d'apprentissage de k

Image originale | $k = 2$ | $k = 3$ | $k = 10$

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

- Inconvénients
  - ✓ Aboutit généralement à une sursegmentation c'est-à-dire à un bien plus grand nombre de régions que d'objets présents dans l'image.
  - ✓ Classification : faible prise en compte de la localisation des pixels dans l'image, du contexte, de la notion de voisinage
    - ➡ Nécessité de méthodes itératives parfois coûteuses en temps de calcul
  - ✓ Dans de nombreuses applications : nombre de classes non connu ! Nécessité d'effectuer une classification non-supervisée.

Source :

©Marie-Flavie Auclair-Fortier

4. RÉGIONS

1. Segmentation à base de classification

- Inconvénients
  - ✓ Aboutit généralement à une sursegmentation c'est-à-dire à un bien plus grand nombre de régions que d'objets présents dans l'image.
  - ✓ Classification : faible prise en compte de la localisation des pixels dans l'image, du contexte, de la notion de voisinage
    - ➡ Nécessité de méthodes itératives parfois coûteuses en temps de calcul
  - ✓ Dans de nombreuses applications : nombre de classes non connu ! Nécessité d'effectuer une classification non-supervisée.

Source :

©Marie-Flavie Auclair-Fortier