

Vision par ordinateur - Reconnaissance Visuelle

Cours 3 : Filtrage et extraction de primitives - partie 1

Céline Hudelot - Mention IA- CentraleSupélec

2022-2023

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

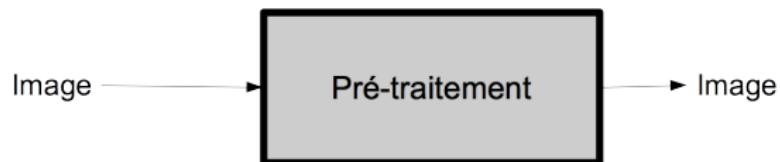
- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Pré-traitement d'une image

Pré-traitement : regroupe l'ensemble des processus visant à améliorer les caractéristiques d'une image.



L'objectif de l'amélioration d'images est de rendre les images plus aptes à l'interprétation humaine ou à celle de la machine.

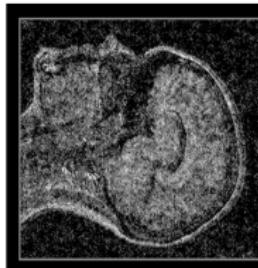
Pré-traitement d'une image

- **Lissage local** : suppression du bruit ou des petites variations présents dans une image. L'intensité d'un pixel est transformée en fonction des intensités sur un **petit voisinage du pixel** (on parle aussi d'amélioration locale).
- **Amélioration d'images** : modification des caractéristiques visuelles de l'image (contraste, ...) pour faciliter son interprétation par l'œil humain (souvent amélioration ponctuelle).
- **Restauration d'images** : suppression des dégradations subies par une image à l'aide de connaissances a priori sur ces dégradations.

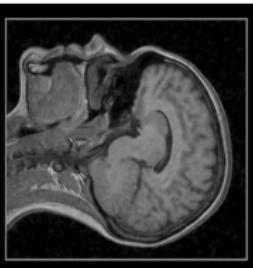
On peut travailler dans le domaine spatial mais aussi dans le domaine fréquentiel (transformée de Fourier).

Pré-traitement d'une image

Exemples de pré-traitements



désbruitage



remplacement de pixels manquants



enhancement de la résolution

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

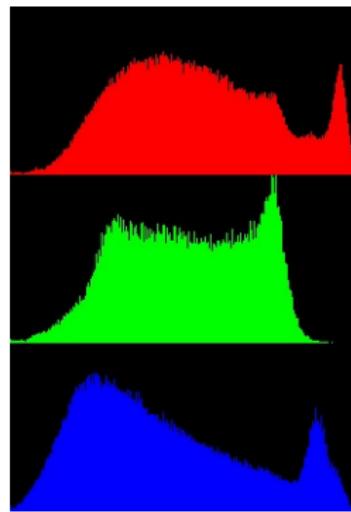
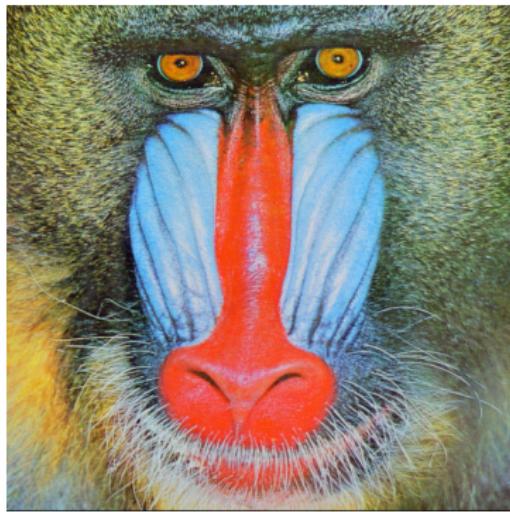
- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Histogramme d'une image

- Distribution des niveaux de gris (ou de couleurs) dans une image.
- $H(k)$ = nombre de pixels ayant la valeur k dans l'image.



Histogramme d'une image

Histogramme normalisé

- Proportion de pixels en fonction du niveau de gris (densité de probabilité)
- H_n donne la probabilité (en termes de fréquence d'occurrence) qu'un pixel ait pour niveau de gris k .
- $x \rightarrow H_n(x) = \frac{H(x)}{nbpixels}$

Histogramme d'une image

Histogramme cumulé

- Histogramme cumulé :

$$H_c(k) = \sum_{i \leq k} H(i)$$

- Histogramme cumulé normalisé :

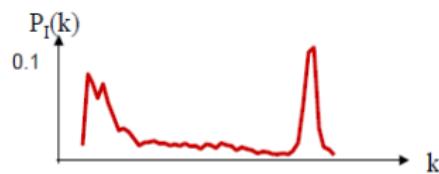
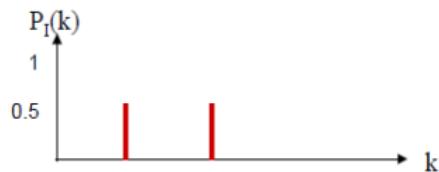
$$H_c(k) = \sum_{i \leq k} H_n(i)$$

- $H_c(k)$ représente la probabilité d'avoir un niveau de gris inférieur ou égal à k .

Histogramme d'une image

Dynamique d'une image $[val_{min}, val_{max}]$:

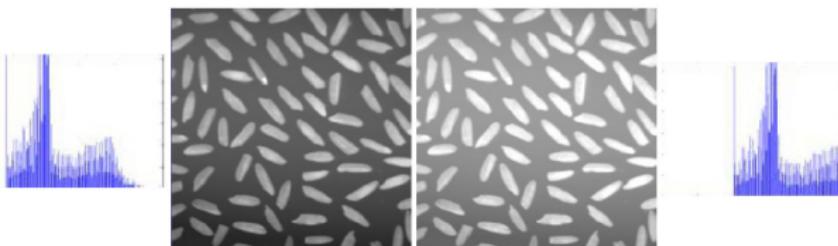
- val_{min} : valeur minimale des niveaux de gris dans l'image.
- val_{max} : valeur maximale des niveaux de gris dans l'image.



Luminance d'une image

Définition

- Moyenne d'intensité de tous les pixels de l'image.
- Pour augmenter la luminance, il suffit de décaler l'histogramme.



Contraste d'une image

Deux définitions

- Ecart type des variations des niveaux de gris.

$$C = \sqrt{\frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f(x, y) - Moy)^2}$$

- Variation entre niveau de gris min et max.

$$C = \frac{\max[f(x, y)] - \min[f(x, y)]}{\max[f(x, y)] + \min[f(x, y)]}$$

Contraste d'une image

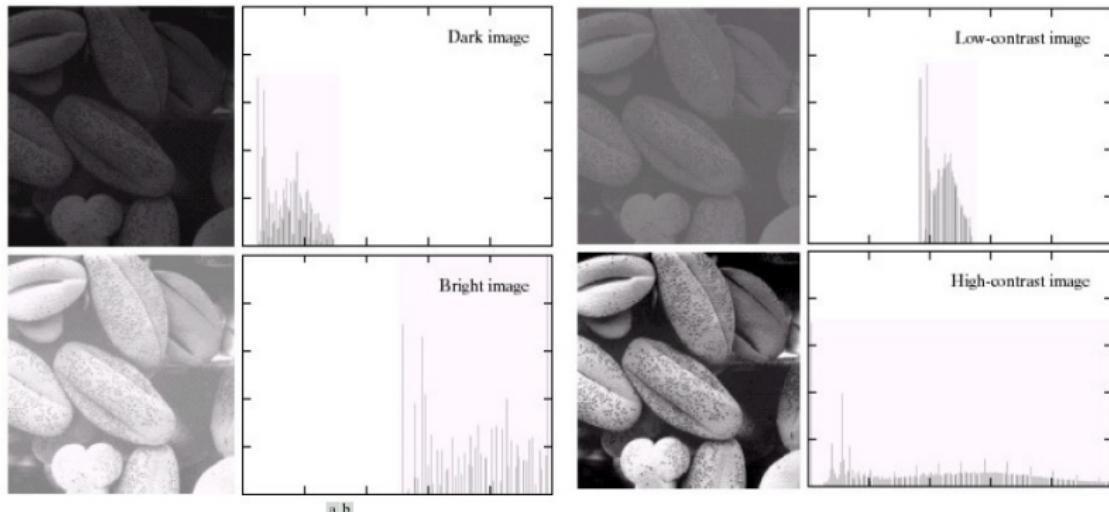


FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Amélioration du contraste

- Transformation linéaire
- Transformation linéaire avec saturation
- Transformation linéaire par morceaux
- Transformation non linéaire
- Egalisation de l'histogramme

Expansion de dynamique

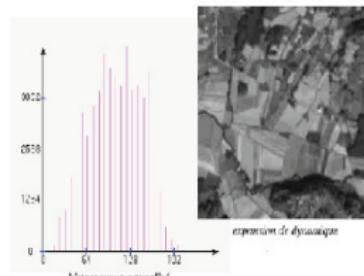
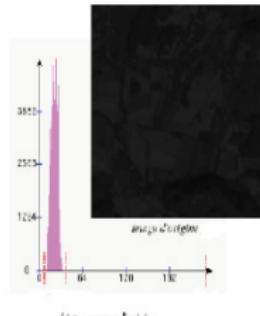
Principe

Transformation des niveaux de gris de telle sorte que l'image utilise toute la dynamique (entre 0 et 255).

- $I_{Exp}(i,j) = (I(i,j) - \min)(\frac{255}{\max - \min})$

Ou de manière générale, expansion à l'intervalle $[0, L - 1]$

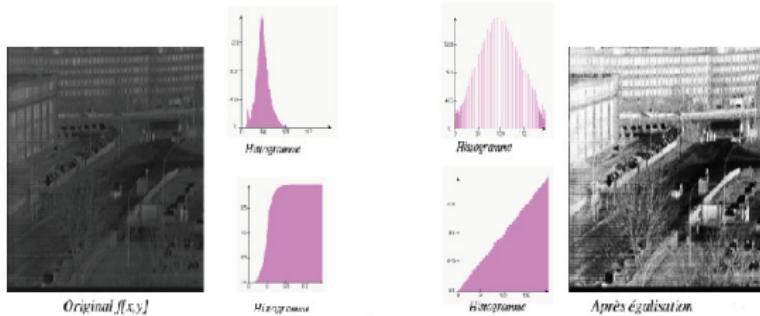
$$I_{Exp}(i,j) = (I(i,j) - \min)(\frac{L - 1}{\max - \min})$$



Egalisation

Principe

- Transformation des niveaux de gris dont le but est d'équilibrer le mieux possible la distribution de pixels dans la dynamique.
- L'histogramme cumulé $h_c(x) = \sum_{z \leq x} h_n(z)$ doit être le plus linéaire possible.
- $I_{Ega}(i,j) = \text{Int}(255 \times h_c(I(i,j)))$ avec Int la fonction qui arrondit à l'entier le plus proche.

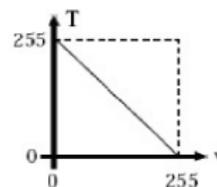
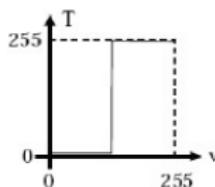
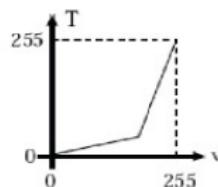
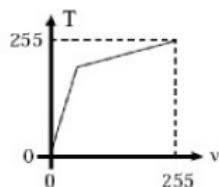


Transformations d'histogramme

Principe algorithmique

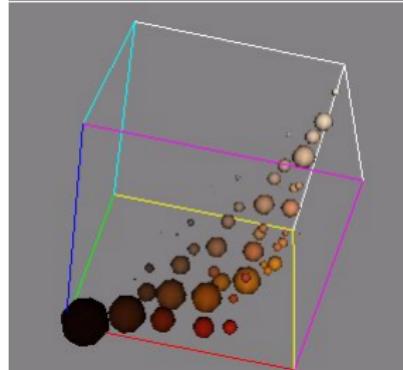
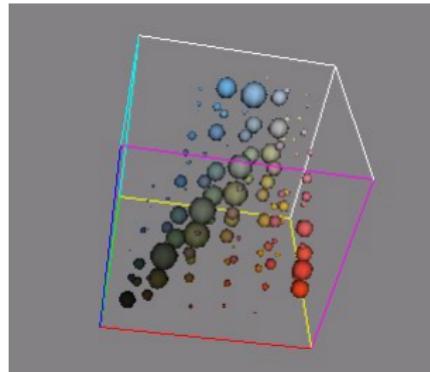
- Parcourir les pixels de l'image.
- Pour chaque pixel (i, j) :
 - ▶ Lire la valeur x .
 - ▶ Remplacer x par $T(x)$
- La fonction $T(\cdot)$ est représentée par son tracé. Le choix de $T(\cdot)$ permet de modifier les caractéristiques d'une image.

Transformations d'histogramme



Histogramme couleur

Histogramme couleur complet en 3D



credit : <http://ij-plugins.sourceforge.net/ij-vtk/color-space/>

Approche par points

- Ces approches sont globalement des **approches par point**, i.e. la valeur de chaque pixel de sortie dépend uniquement de la valeur du pixel d'entrée correspondant (et éventuellement, quelques informations supplémentaires globales).
- Des approches pour améliorer principalement la qualité visuelle de l'image.
- Avec des applications importantes pour l'amélioration des techniques modernes de reconnaissance visuelle (augmentation de données, apprentissage auto-supervisé...)

Applications récentes



Image augmentation

source : <https://github.com/aleju/imgaug>

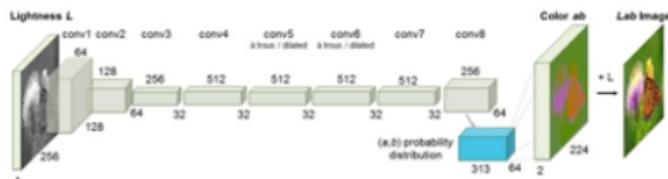


Fig. 2. Our network architecture. Each **conv** layer refers to a block of 2 or 3 repeated **conv** and **ReLU** layers, followed by a **BatchNorm** [30] layer. The net has no **pool** layers. All changes in resolution are achieved through spatial downsampling or upsampling between **conv** blocks.

Colorful image colorization

Self-supervised learning with colorization

source : <https://richzhang.github.io/colorization/>

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Filtrage

Filtrage

Opération qui forme une nouvelle image dont les pixels sont une combinaison des valeurs des pixels d'une image originale.

Objectifs

- Extraire une information utile à partir de l'image :
 - ▶ Caractéristiques : contours, coins, ...
- Modifier ou mettre en avant certaines propriétés de l'image :
 - ▶ Débruitage, super résolution, inpainting, ...

Filtrage

- Parallèle avec le filtrage pour un signal 1D : supprimer du bruit ou sélectionner certaines fréquences.
- En image, on fait souvent référence à la fréquence spatiale.
 - ▶ Fréquence spatiale élevée : détails qui se répètent fréquemment sur un petit nombre de pixels.
 - ▶ Fréquence spatiale faible : variations qui se répètent peu.

Filtrage

Familles de filtres

- Passe-haut : élimine les fréquences basses.
- Passe-bas : élimine les fréquences hautes.
- Passe-bande : ne laisse passer qu'une gamme de fréquences.

Lissage de l'image

- Filtres passe-bas.
- Objectif : atténuation du bruit et des détails (basses fréquences).

Dérivation de l'image

- Filtres passe-haut.
- Objectif : extraction de contours, de points d'intérêts par l'accentuation des détails et des contours (hautes fréquences).

Filtrage

Propriétés générales

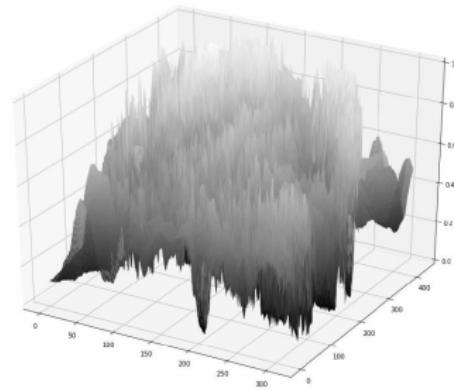
- Domaine de représentation du signal à filtrer :
 - ▶ Spatial
 - ▶ Fréquentiel
- Filtres linéaires - Filtres non linéaires.
- Filtres isotropes - Filtres anisotropes.

Systèmes et filtres

Image comme une fonction

- Une image est une fonction f de \mathbb{R}^2 à \mathbb{R}^M .
- $f[x, y]$ donne **l'intensité** à la position $[x, y]$
- Définie sur un rectangle support, avec un intervalle fini.

$$f: \underbrace{[a, b] \times [c, d]}_{\text{Domain support}} \rightarrow \underbrace{[0, 255]}_{\text{range}}$$



Systèmes et filtres

$$f[n, m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n, m]$$

- On considère une image comme une fonction $f : \mathbb{R}^2 \rightarrow \mathbb{R}^M$.
- Un système S convertit une fonction d'entrée $f[n, m]$ en une fonction de sortie $g[n, m]$.
- Dans le cas des images, $[n, m]$ représente la **position spatiale dans l'image**

Systèmes et filtres

$$f[n, m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n, m]$$

- \mathcal{S} est le système (opérateur), défini comme un mapping qui transforme l'entrée f en la sortie g .

$$g = \mathcal{S}[f] \quad g[n, m] = \mathcal{S}\{f[n, m]\}$$

$$f[n, m] \xrightarrow{\mathcal{S}} g[n, m]$$

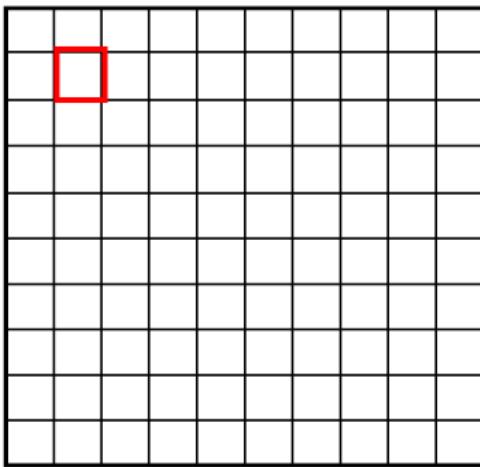
Exemple 1 : Filtre moyenneur

Moyenne mobile 2D sur un voisinage 3×3 .

$f[n, m]$

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0
0	0	0	90	90	90	90	90	0
0	0	0	90	90	90	90	90	0
0	0	0	90	0	90	90	90	0
0	0	0	90	90	90	90	90	0
0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

$g[n, m]$

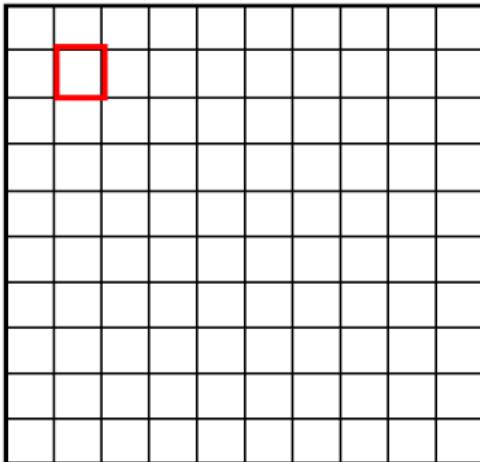


Courtesy of S. Seitz

Exemple 1 : Filtre moyenneur

 $f[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

 $g[n, m]$ 

Courtesy of S. Seitz

Exemple 1 : Filtre moyenneur

 $f[n, m]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

 $g[n, m]$

0	10									

Exemple 1 : Filtre moyenneur

 $f[n, m]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

 $g[n, m]$

			0	10	20					

Exemple 1 : Filtre moyenneur

 $f[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

 $g[n, m]$

Exemple : Filtre moyenneur

$f[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[n, m]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Exemple : Filtre moyenneur

$$g[n, m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k, l]$$

Sum over rows
Sum over columns

$$= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 f[n - k, m - l]$$

Filter "kernel", "mask"

$$\frac{1}{9} \begin{matrix} h \\ \hline \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \end{matrix}$$

Exemple : Filtre moyenneur

En résumé

- Ce filtre transforme chaque valeur de pixel en la moyenne de ses voisins.
- Effet de lissage.

Systèmes et filtres

Les systèmes peuvent être classés selon leurs propriétés :

- Amplitude

- ▶ Additivité
- ▶ Homogénéité
- ▶ Superposition
- ▶ Stabilité
- ▶ Invertibilité

- Spatiales

- ▶ Causalité (les effets ne peuvent pas précéder la cause)
- ▶ Invariance aux déplacements (Shift invariance)
- ▶ Mémoire

Systèmes et filtres

Systèmes invariants aux déplacements (SI : Shift-Invariant)

Si $f[n, m] \xrightarrow{\mathcal{S}} g[n, m]$

alors, \mathcal{S} est invariant aux déplacements (SI) si

$$f[n - n_0, m - m_0] \xrightarrow{\mathcal{S}} g[n - n_0, m - m_0]$$

pour chaque entrée $f[n, m]$ et déplacements n_0 et m_0 .

A faire à la maison !

Est-ce que le filtre moyenneur est invariant aux déplacements ?

Systèmes et filtres linéaires

$$f[n, m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n, m]$$

- Filtrage linéaire :
 - ▶ Forme une nouvelle image dont les pixels sont une somme pondérée des valeurs des pixels originales.
 - ▶ Le même ensemble de poids est utilisé à chaque point.
- Rappel : fonction linéaire
 - ▶ $f(\alpha x) = \alpha f(x)$
 - ▶ $f(x + y) = f(x) + f(y)$

\mathcal{S} est un système linéaire si et seulement si

$$\mathcal{S}\{\alpha f_1[n, m] + \beta f_2[n, m]\} = \alpha \mathcal{S}\{f_1[n, m]\} + \beta \mathcal{S}\{f_2[n, m]\}$$

(Propriété de superposition)

Systèmes et filtres linéaires invariants aux déplacements

Système LSI (Linear Shift Invariant)

- Propriété de superposition

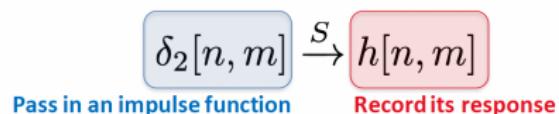
$$\mathcal{S}\{\alpha f_1[n, m] + \beta f_2[n, m]\} = \alpha \mathcal{S}\{f_1[n, m]\} + \beta \mathcal{S}\{f_2[n, m]\}$$

- Invariance aux déplacements

$$f[n - n_0, m - m_0] \xrightarrow{\mathcal{S}} g[n - n_0, m - m_0]$$

Systèmes et filtres linéaires invariants aux déplacements (LSI)

- Un système LSI est complètement spécifié par sa réponse impulsionnelle.

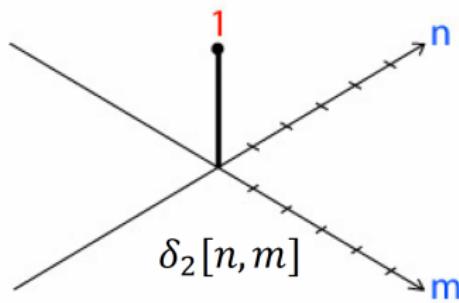


Systèmes et filtres linéaires invariants aux déplacements (LSI)

2D impulse function $\delta_2[n, m]$

- 1 at [0,0].
- 0 everywhere else

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0



Filtre moyenneur et réponse impulsionnelle

- Filtre moyenneur

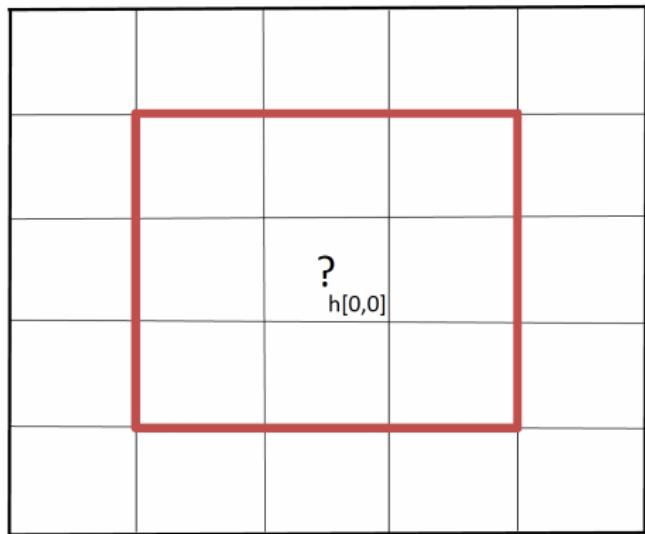
$$f[n, m] \xrightarrow{S} \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 f[n - k, m - l]$$

- On l'utilise pour obtenir l'expression de la réponse impulsionnelle

$$\delta_2[n, m] \xrightarrow{S} h[n, m]$$

$$= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

Filtre moyenneur et réponse impulsionale



$$\begin{aligned}\delta_2[n, m] &\xrightarrow{S} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]\end{aligned}$$

Filtre moyenneur et réponse impulsionale

		$1/9$ $h[0,0]$?	$h[0,1]$

$$\delta_2[n, m] \xrightarrow{S} h[n, m]$$

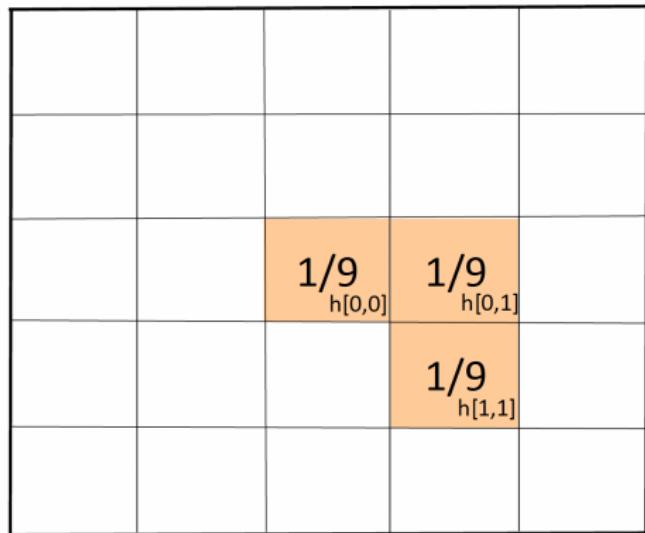
$$= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

Filtre moyenneur et réponse impulsionale

		$1/9_{h[0,0]}$	$1/9_{h[0,1]}$	
			$?_{h[1,1]}$	

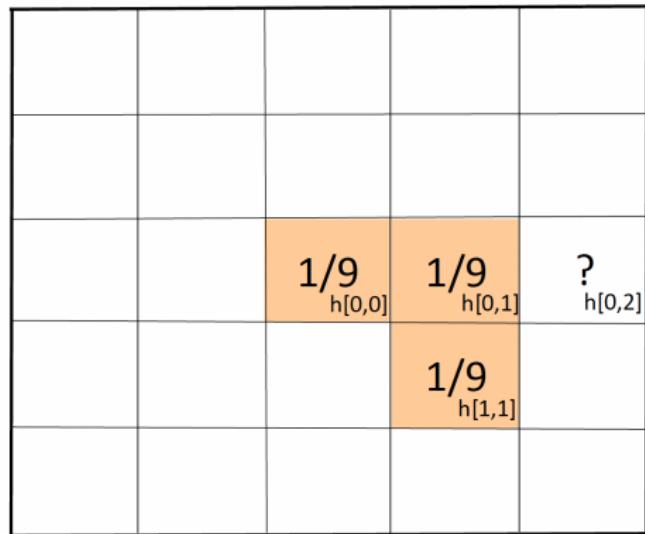
$$\delta_2[n, m] \xrightarrow{S} h[n, m] = \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

Filtre moyenneur et réponse impulsionale



$$\delta_2[n, m] \xrightarrow{S} h[n, m] = \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

Filtre moyenneur et réponse impulsionale



$$\delta_2[n, m] \xrightarrow{S} h[n, m] = \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

Filtre moyenneur et réponse impulsionale

		$1/9$ $h[0,0]$	$1/9$ $h[0,1]$	0 $h[0,2]$
			$1/9$ $h[1,1]$	

$$\begin{aligned}\delta_2[n, m] &\xrightarrow{S} h[n, m] \\ &= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]\end{aligned}$$

Filtre moyenneur et réponse impulsionale

0	0	0	0	0
0	$\frac{1}{9}$ $_{h[-1,-1]}$	$\frac{1}{9}$	$\frac{1}{9}$	0
0	$\frac{1}{9}$	$\frac{1}{9}$ $_{h[0,0]}$	$\frac{1}{9}$ $_{h[0,1]}$	0 $_{h[0,2]}$
0	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$ $_{h[1,1]}$	0
0	0	0	0	0

$$\delta_2[n, m] \xrightarrow{S} h[n, m]$$

$$= \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

Filtre moyenneur et réponse impulsionnelle

Réponse impulsionnelle d'un filtre moyenneur 3×3

$$h[n, m] = \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 \delta_2[n - k, m - l]$$

$$= \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h$$

Systèmes et filtres linéaires invariants aux déplacements (LSI)

- Un système LSI est complètement spécifié par sa réponse impulsionnelle.
 - ▶ Pour chaque entrée g , on peut calculer la sortie g en terme de la réponse impulsionnelle.
 - ▶ Dans la suite, **on a écrit g en fonction de h**
 - ▶ On sait qu'un système LSI respecte la propriété de superposition et l'invariance aux déplacements.
 - ▶ On connaît h

$$\delta_2[n, m] \xrightarrow{S} h[n, m]$$

Systèmes et filtres linéaires invariants aux déplacements (LSI)

Idée principale

Ecrire f comme une somme de réponses impulsionales.

Considérons une image f de taille 3×3

$$\begin{array}{|c|c|c|} \hline f[0,0] & f[0,1] & f[1,1] \\ \hline f[1,0] & f[1,1] & f[1,2] \\ \hline f[2,0] & f[2,1] & f[2,2] \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline f[0,0] & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & f[0,1] & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \dots + \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & f[2,2] \\ \hline \end{array}$$

$$= f[0,0]*\begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + f[0,1]*\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \dots + f[2,2]*\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

$$= f[0,0] \cdot \delta_2[n, m] + f[0,1] \cdot \delta_2[n, m - 1] + \dots + f[2,2] \cdot \delta_2[n - 2, n - 2]$$

Systèmes et filtres linéaires invariants aux déplacements (LSI)

- Plus généralement

$$f[n, m] = \sum_{k=-\infty}^{-\infty} \sum_{l=-\infty}^{-\infty} f[k, l] \cdot \delta_2[n - k, m - l]$$

- Propriété de superposition

$$f[n, m] = \sum_{k=-\infty}^{-\infty} \sum_{l=-\infty}^{-\infty} f[k, l] \cdot \delta_2[n - k, m - l]$$

$$\xrightarrow{\mathcal{S}} \sum_{k=-\infty}^{-\infty} \sum_{l=-\infty}^{-\infty} f[k, l] \cdot \mathcal{S}\{\delta_2[n - k, m - l]\}$$

- Invariance

$$\mathcal{S}\{\delta_2[n - k, m - l]\} = h[n - k, m - l]$$

Systèmes et filtres linéaires invariants aux déplacements (LSI)

- Un système LSI est complètement spécifié par sa réponse impulsionnelle.
- Pour chaque entrée f , on calcule g en termes de sa réponse impulsionnelle

$$f[n, m] \xrightarrow{\mathcal{S}} g[n, m]$$

$$f[n, m] \xrightarrow{\mathcal{S}} \sum_{k=-\infty}^{-\infty} \sum_{l=-\infty}^{-\infty} f[k, l].h[n - k, m - l]$$

- Convolution discrète

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{-\infty} \sum_{l=-\infty}^{-\infty} f[k, l].h[n - k, m - l]$$

Filtrage

Filtrage spatial d'une image : réduction du bruit

Définition du bruit

- Phénomène parasite aléatoire dont les origines sont diverses (capteurs, acquisition, contexte...).
- Pour le filtrage linéaire, on considère que le bruit est additif

$$I_b(i,j) = I(i,j) + b(i,j)$$

- Exemples :
 - ▶ Bruit gaussien : $I_b(x,y) = I(x,y) + \mathcal{N}(0,\sigma)$
 - ▶ Bruit impulsionnel poivre et sel d'ordre n : ajouter n pixels blancs et n pixels noirs aléatoirement dans l'image.
 - ▶ ...

Hypothèse fondamentale pour la réduction du bruit : le signal utile et le bruit ont des composantes fréquentielles différentes
signal utile = basses fréquences ; bruit = hautes fréquences

Filtre linéaire

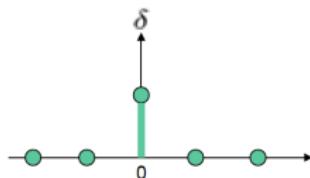
Filtrage spatial linéaire : opération de convolution

- **Filtrage linéaire spatial** : convoluer l'image avec la réponse impulsionnelle du filtre.

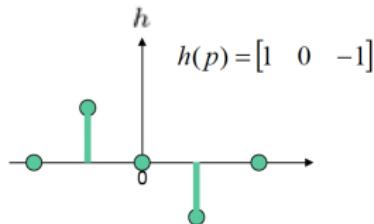
$$g[i, j] = (f * h)[i, j] = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} f[i - m, j - n]h[m, n]$$

En 1D

$$g(p) = \sum_{i=-\infty}^{i=+\infty} f(p - i)h(i)$$



Impulsion unité



Réponse impulsionnelle

Filtre linéaire spatial

Filtrage spatial linéaire 2D discret

- Le produit de convolution d'un signal 2D $f(i, j)$ (une image) avec un filtre h est donné par

$$g[i, j] = (f * h)[i, j] = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} f[i - m, j - n]h[m, n]$$

- Deux situations :
 - ▶ Filtres de Réponse Impulsionnelle $h(m, n)$ finie (RIF)
 - ▶ Filtres de Réponse Impulsionnelle $h(m, n)$ infinie (RII)

Filtre linéaire spatial

Filtrage spatial linéaire 2D discret

- ① Filtres de Réponse Impulsionnelle $h(m, n)$ infinie (RII) : $h(m, n)$ ne s'annule pas.
- ② Filtres de Réponse Impulsionnelle $h(m, n)$ finie (RIF) : $h(m, n)$ est nulle en dehors d'un intervalle entre m et n
 - ▶ On parle de masque de convolution

Filtrage spatial d'une image : mise en oeuvre

Convolution discrète

Le filtrage est une **opération de voisinage** : la valeur d'un pixel est remplacée par la valeur d'une fonction appliquée à ce pixel et à ses voisins.



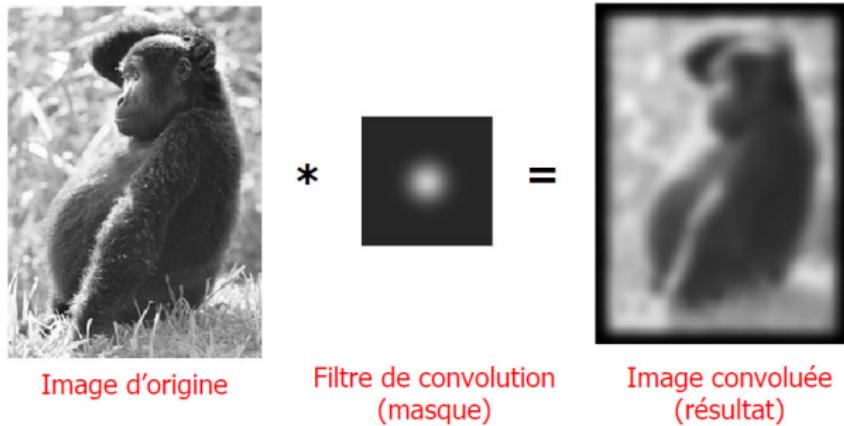
Voisinage à 4 points
noté $V_4(i,j)$



Voisinage à 8 points
noté $V_8(i,j)$

- **Filtres linéaires** : la fonction appliquée est une combinaison linéaire : utilisation de masque de convolution (convolution de l'image avec la réponse impulsionnelle du filtre).

Filtrage d'une image : exemple de convolution 2D



Note : par convention pratique, la taille de l'image résultat est la même que celle de l'image d'origine

FIGURE – Source : Alain Boucher

Filtre linéaire

Implémentation des filtres linéaires

Soit I une image numérique. Soit h une fonction de $[x_1, x_2] \times [y_1, y_2]$ à valeurs réelles (noyau de convolution). La **convolution** de I par h est définie par :

$$(I * h)(x, y) = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h(i, j) I(x - i, y - j)$$

Les nouvelles valeurs sont calculées par le produit scalaire entre le noyau (masque) de convolution et le voisinage du pixel.

Filtre linéaire : principe de calcul

Exemple

- Soit $h(n, m)$ un masque carré de taille $d = 3$ impaire. Alors,

$$f'(i, j) = (f * h)(i, j) = \sum_{n=-1}^1 \sum_{m=-1}^1 f(i - n, j - m)h(n, m)$$

- Si on pose, $g(n, m) = h(-n, -m)$, on obtient alors la somme pondérée de l'image avec g

Filtre linéaire : principe de calcul

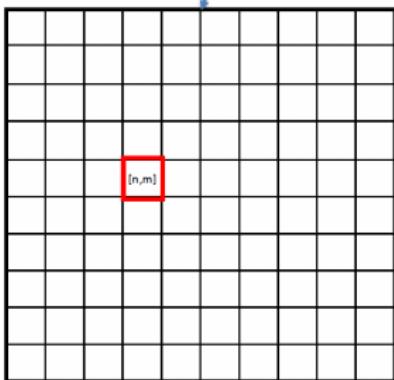
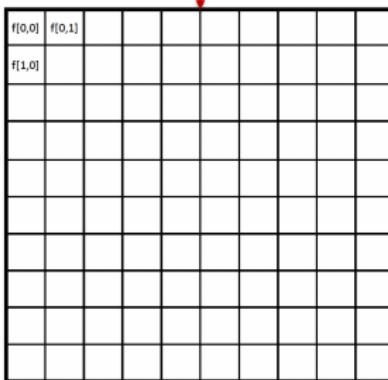
$$h = \begin{pmatrix} w_9 & w_8 & w_7 \\ w_6 & w_5 & w_4 \\ w_3 & w_2 & w_1 \end{pmatrix} g = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

Convolution en (i, j) par h

$$\begin{aligned} f'(i, j) &= w_1 f(i - 1, j - 1) + w_2 f(i - 1, j) + w_3 f(i - 1, j + 1) \\ &\quad + w_4 f(i, j - 1) + w_5 f(i, j) + w_6 f(i, j + 1) \\ &\quad + w_7 f(i + 1, j - 1) + w_8 f(i + 1, j) + w_9 f(i + 1, j + 1) \end{aligned}$$

Filtre linéaire : principe de calcul

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot h[n - k, m - l]$$

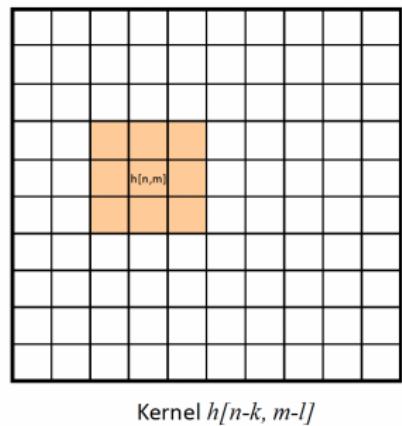
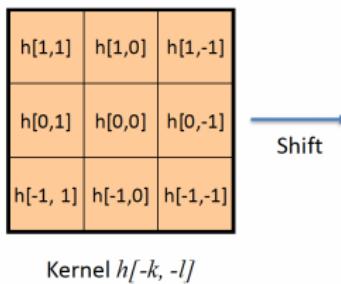
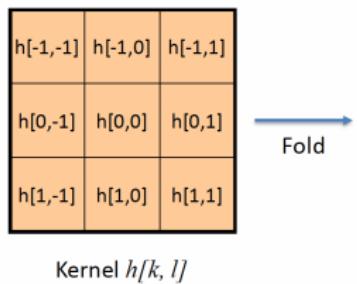
Output $f * h$ 

$h[-1, -1]$	$h[-1, 0]$	$h[-1, 1]$
$h[0, -1]$	$h[0, 0]$	$h[0, 1]$
$h[1, -1]$	$h[1, 0]$	$h[1, 1]$

Kernel $h[k, l]$

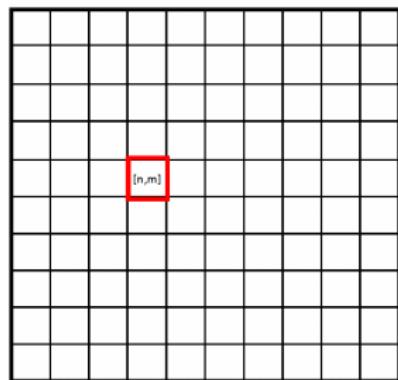
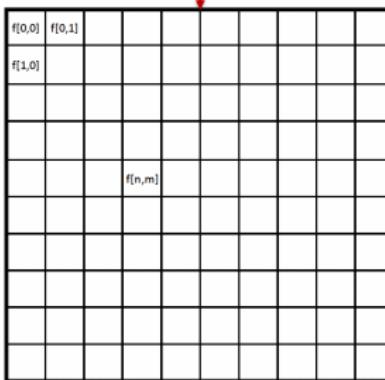
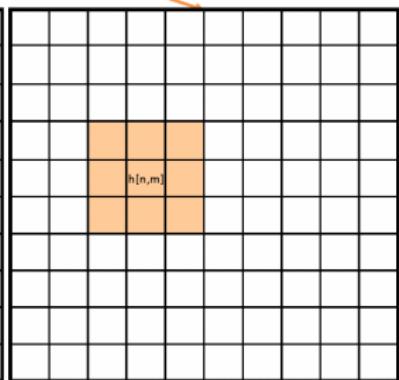
Filtre linéaire : principe de calcul

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot h[n - k, m - l]$$



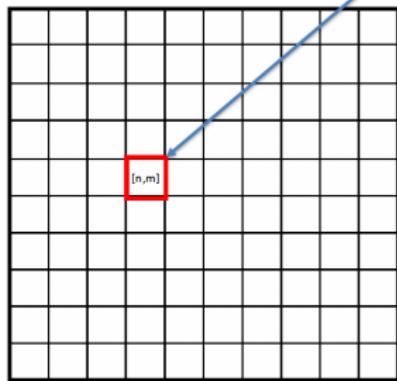
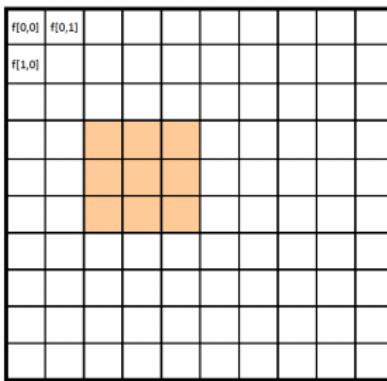
Filtre linéaire : principe de calcul

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot h[n - k, m - l]$$

Output $f * h$ Image $f[k, l]$ Kernel $h[n-k, m-l]$

Filtre linéaire : principe de calcul

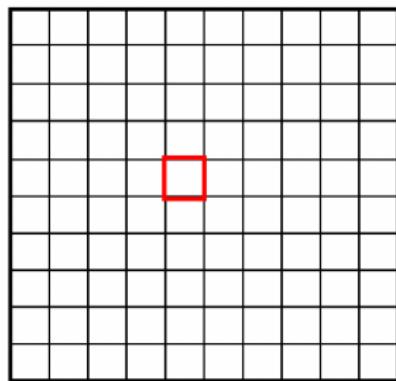
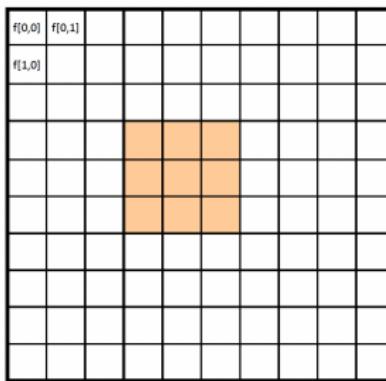
$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot h[n - k, m - l]$$

Output $f * h$ 

Element-wise multiplication
Image $f[k, l] \bullet$ Kernel $h[n-k, m-l]$

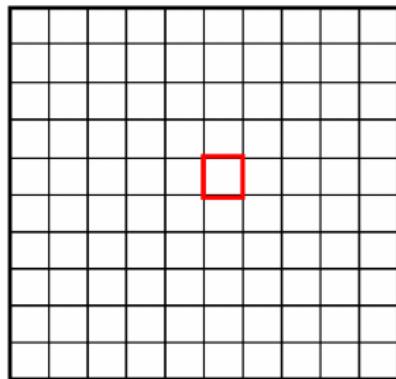
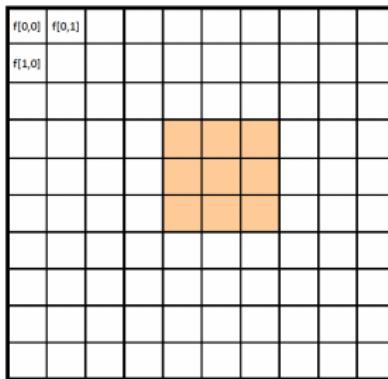
Filtre linéaire : principe de calcul

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot h[n - k, m - l]$$

Output $f * h$ Element-wise multiplication
Image $f[k, l] \bullet$ Kernel $h[n-k, m-l]$

Filtre linéaire : principe de calcul

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot h[n - k, m - l]$$

Output $f * h$ 

Element-wise multiplication
Image $f[k, l] \cdot$ Kernel $h[n-k, m-l]$

Filtre linéaire : principe de calcul

Soit un filtre RIF de réponse impulsionnelle $h(m, n)$. Au pixel $p = f(i, j)$:

- Faire une symétrie centrale du noyau de convolution par rapport à son centre :

$$h(m, n) \Rightarrow h(-m, -n) = g(m, n)$$

- Centrer le filtre sur p en le superposant à l'image.
- Effectuer la somme pondérée entre les pixels de l'image et les coefficients du filtre $g(m, n)$
- Le pixel p dans l'image but (filtrée) aura comme valeur la somme pondérée.

Filtre linéaire : propriétés

On peut toujours représenter un filtre linéaire, invariant spatialement par une convolution. On note :

$$h = f * g$$

- Linéarité (convolution distributive) :

$$f * (g + h) = (f * g) + (f * h)$$

- Invariance spatiale = indépendant de la position du pixel :

$$\tau_x(f * g) = (\tau_x f) * g = f * (\tau_x g)$$

avec :

$$(\tau_x f)(y) = f(y - x)$$

LSI (Linear Shift-Invariant)

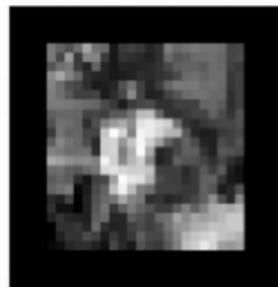
Filtrage d'une image : mise en oeuvre

Problème aux bords

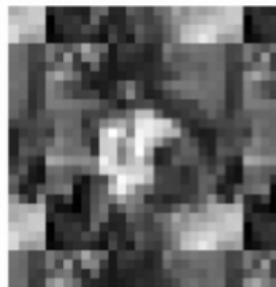
- Mettre à 0 (zero padding) ou à une valeur constante.
- Convolution partielle sur une portion du masque.
- Miroir de l'image : $f(-x, y) = f(x, y)$.
- répétition du bord de l'image : l'image est entourée des mêmes valeurs que sur son bord.
- Pas de solution idéale.

Filtrage d'une image : mise en oeuvre

Problème au bords



zero



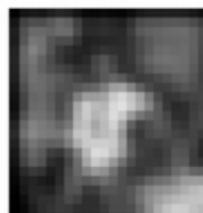
wrap



clamp



mirror



blurred: zero



normalized zero



clamp



mirror

Source : Szeliski

Filtrage d'une image

Filtres isotropes - anisotropes

- En 2D, un filtre est isotrope si le filtrage réalisé est indépendant de l'orientation des structures de l'image.
- Pour une image, un filtre symétrique est isotrope
- Un filtre anisotrope réagira en fonction des directions de luminance de l'image
to détection de contours...

Filtrage d'une image

Filtres linéaires séparables

- Une réponse impulsionnelle g est séparable selon x et y si et seulement si :

$$g(x, y) = g_x(x) \cdot g_y(y)$$

- En terme de filtrage d'une image par convolution

$$h(x, y) = g(x, y) * I(x, y) = g_y(y) * (g_x(x) * I(x, y))$$

- Avantages d'un filtre séparable :

- ▶ On ramène le filtre d'un signal 2D à un filtrage en 1D
- ▶ Réduction du temps de calcul
- ▶ On peut aussi souvent implémenter récursivement le filtrage.

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Filtre moyenneur : filtre linéaire

Principe

- Permet de lisser l'image.
- Remplace chaque pixel par la valeur moyenne de ses voisins.
- Réduction du bruit et des détails *non importants*.
- Rend l'image floue (*blur edges*).
- Peut être appliqué itérativement.
- Filtre passe bas.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

ou $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

3x3

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

5x5

FIGURE – Source : Alain Boucher

Filtre moyenneur : exemple



Moyenne 3x3

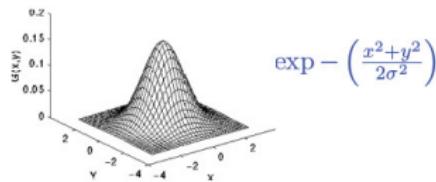


Moyenne 5x5

Filtre gaussien

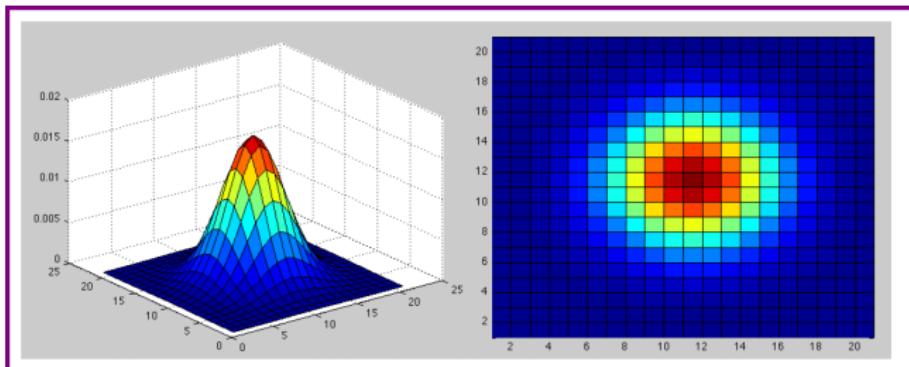
Principe

- Amélioration du filtre moyenneur : moyennage pondéré de l'image en fonction de la distance du pixel voisin.
- Donne plus d'importance aux pixels du centre.
- Approximation discrète d'un noyau gaussien 2D.
- La largeur du filtre est donnée par son écart-type σ .



$$h_{3 \times 3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad h_{5 \times 5} = \frac{1}{246} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Filtre gaussien



- Le noyau gaussien est défini par un ensemble de coefficients qui sont des échantillons de la gaussienne 2D.
- La largeur du filtre est donnée par son écart-type σ , i.e. largeur de part et d'autre du point central :
$$2\text{Ent}^+(3\sigma) + 1$$
- Si σ est petit, le lissage n'a presque pas d'effet.
- Plus σ est grand, plus on réduit le bruit mais plus l'image est floue.

Filtre gaussien : exemple



Original image



Average filter



Gaussian filter

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Filtre median : filtre non linéaire

Principe

- Pour nettoyer le bruit dans une image, il existe mieux que le filtre moyenneur ou le filtre gaussien.
- **Filtre médian.**
- Filtre non linéaire qui ne s'implémente pas comme un produit de convolution.
- La valeur d'un pixel est remplacée par la valeur médiane de son voisinage NxN.
- Préserve l'information de contour en supprimant le bruit impulsif.

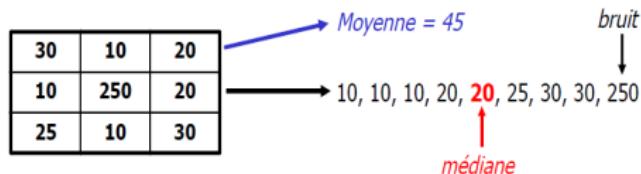


FIGURE – Source : Alain Boucher

Filtre median : exemple

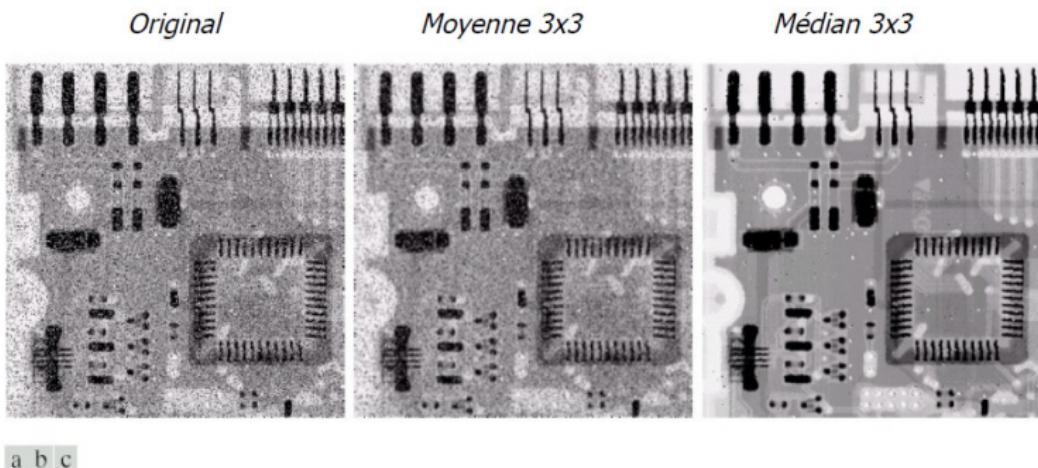


FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

FIGURE – Source : Gonzales & Woods

Filtre median : exemple



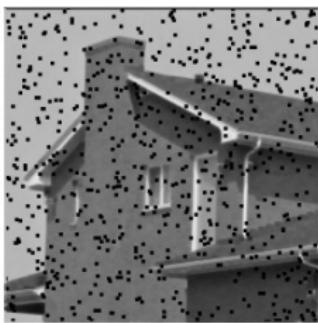
Image initiale



Bruit Poivre & Sel



Moyenne V8



Min V8



Max V8



Médian V8

FIGURE – Source : Alain Boucher

Petite parenthèse sur l'opération de convolution

Les réseaux convolutionnels.

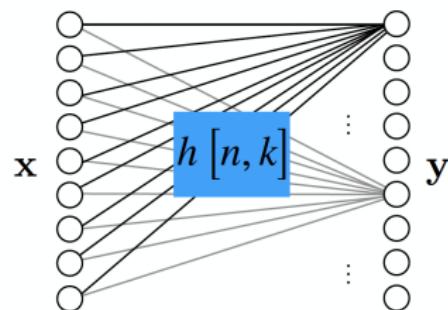
Petite parenthèse sur l'opération de convolution

A linear function f can be written as a matrix multiplication:

$$y = \begin{bmatrix} & & & \\ & h[n,k] & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} & & & \\ & x & & \\ & & & \\ & & & \end{bmatrix}$$

n indexes rows,
k indexes columns

It can also be represented as a fully connected linear neural network



$h[n, k]$ Is the strength of the connection between $x[k]$ and $y[n]$

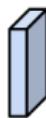
Convolutional neural networks : a convolutional layer

From the course of Li, Karpathy and Johnson in Stanford

32x32x3 image

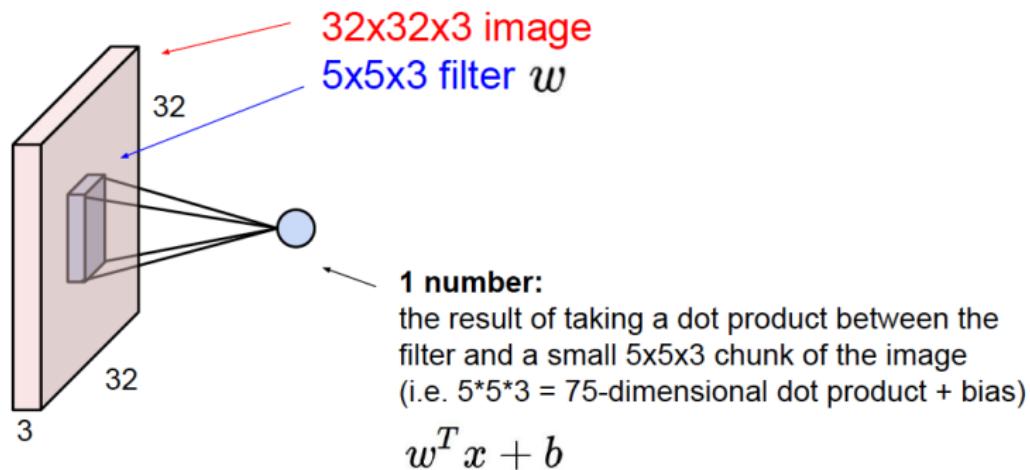


5x5x3 filter

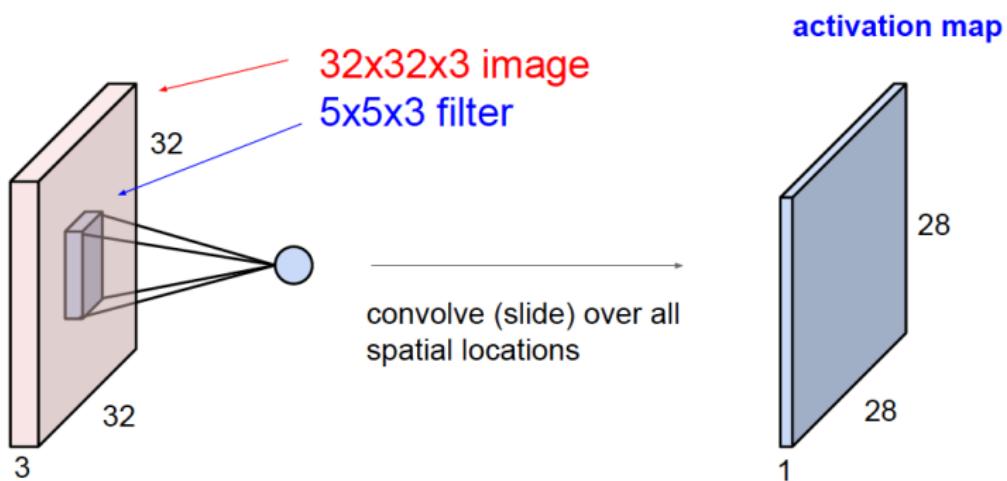


Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolutional neural networks : a convolutional layer

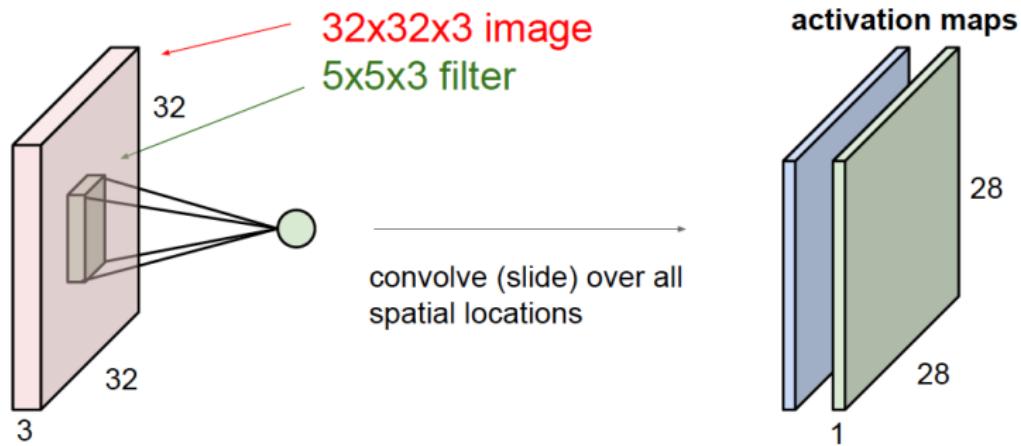


Convolutional neural networks : a convolutional layer



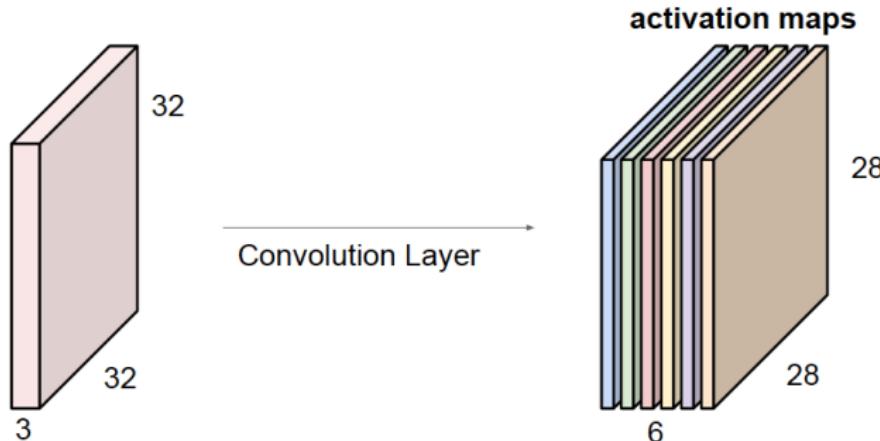
Convolutional neural networks : a convolutional layer

With another filter : the green one



Convolutional neural networks : a convolutional layer

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a “new image” of size $28 \times 28 \times 6$!

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Avant propos

Pourquoi ?

- Etape préliminaire à de nombreuses applications de l'analyse d'images.
- Des indices riches pour l'interprétation de la scène.
- Représentation plus compacte : information de toute l'image résumée dans les contours des différents objets.
- Biologiquement plausible : une des premières étapes dans la vision est de reconnaître des contours.



http://perso.telecom-paristech.fr/~bloch/TDI/poly_contours.pdf

Avant propos

Une tâche difficile

Que sont les contours dans cette image ?



Qu'est-ce qu'un contour ?

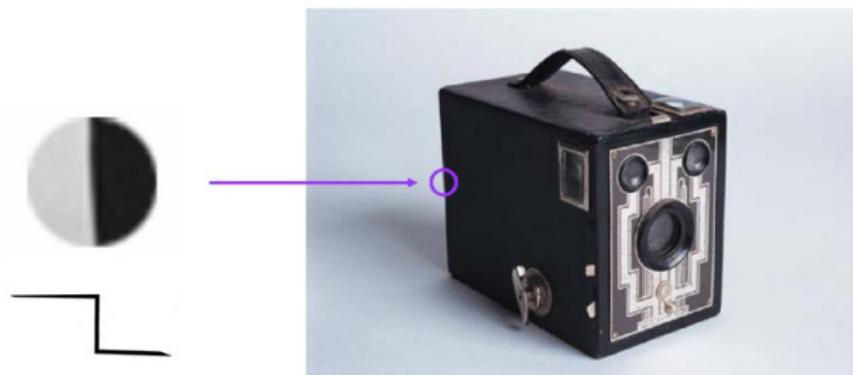


FIGURE – Source : Gluckman & Wong

Une variation brusque d'intensité.

Qu'est-ce qu'un contour ?

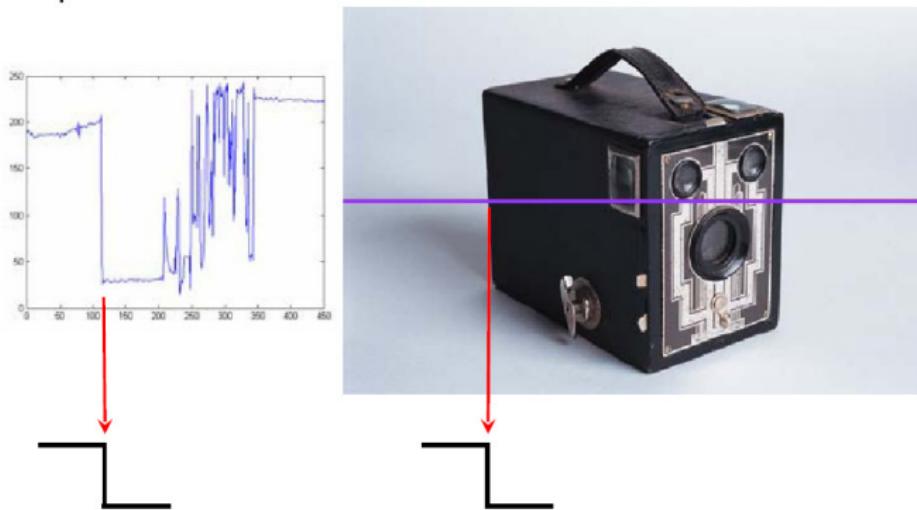


FIGURE – Source : Gluckman & Wong

Qu'est-ce qu'un contour ?

Une discontinuité dans l'image (changements d'intensité)

de profondeur

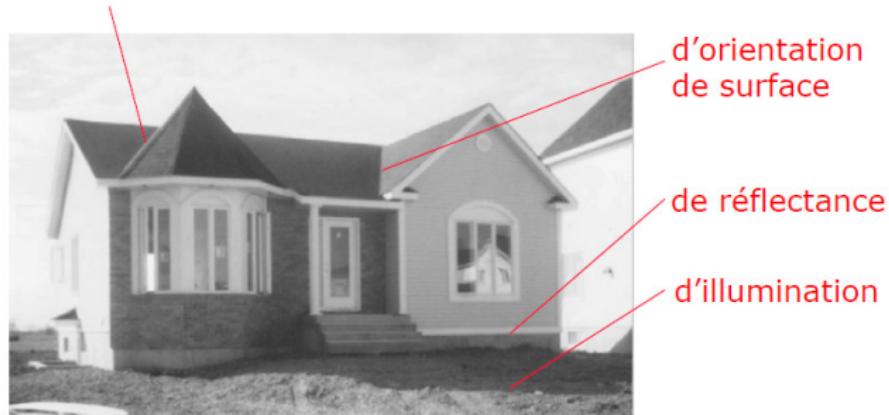


FIGURE – Source : Jacques-André Landry

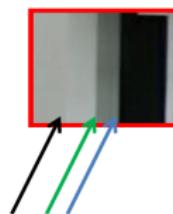
Qu'est-ce qu'un contour ?



FIGURE – Source : D. Hoeim

Qu'est-ce qu'un contour ?

Discontinuité d'orientation de surface



Source: D. Hoiem

FIGURE – Source : D. Hoeim

Qu'est-ce qu'un contour ?

Discontinuité de profondeur



Source: D. Hoiem

FIGURE – Source : D. Hoeim

Qu'est-ce qu'un contour ?

Discontinuité de couleur de surface

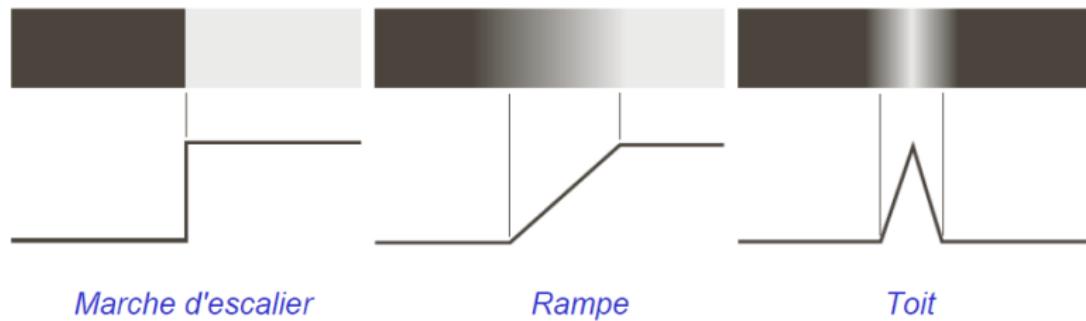


Source: D. Hoeim

FIGURE – Source : D. Hoeim

Les différents types de contours

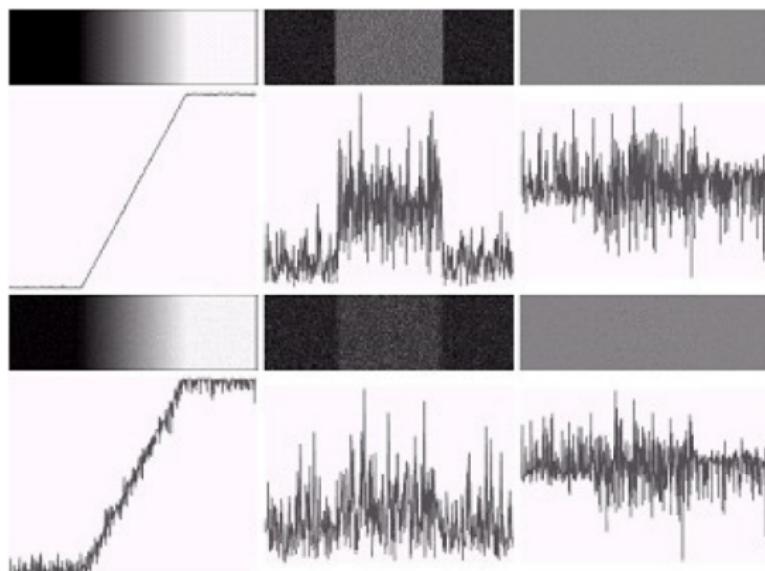
Caractérisés par des discontinuités dans la fonction d'intensité.



Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Mais dans la réalité

Beaucoup de bruit



Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Extraction de contours

Méthode en deux étapes

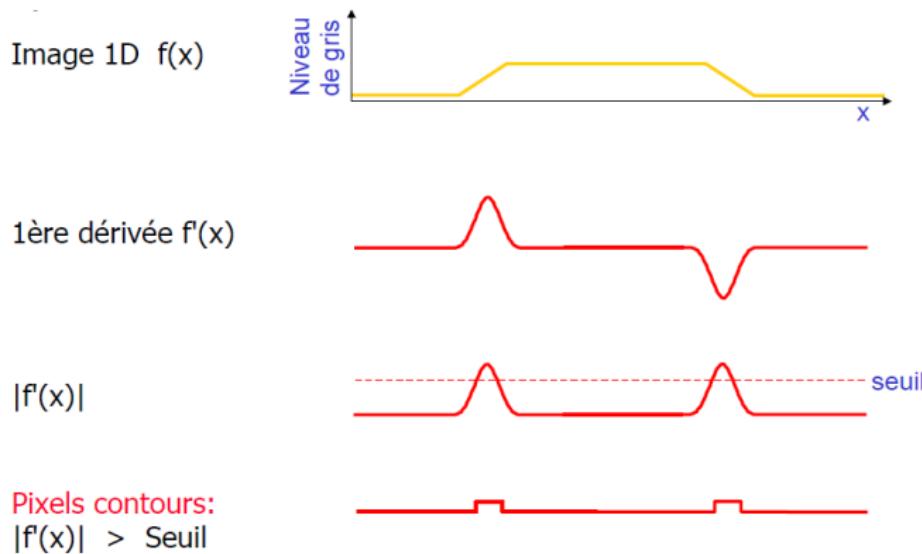
- Première étape : permet de localiser les contours à partir du **calcul du gradient ou du laplacien** dans des directions privilégiées tout en quantifiant l'importance du contour.
- Seconde étape : isoler les contours du reste de l'image à partir d'un **seuillage judicieux**.

Extraction de contours

Principe : deux étapes

- Filtre passe-haut
 - ▶ Approche gradient
 - ▶ Approche laplacien
- Seuillage
 - ▶ Seuillage simple
 - ▶ Seuillage par hysteresis

Dérivée d'une image et contour



Source : Alain Boucher

Etude des dérivées de la fonction d'intensité dans l'image

Gradient de l'image

Rappel

L'image est une fonction :

$$I : S \rightarrow \Omega$$

$$(x, y) \rightarrow I(x, y)$$

Dérivée première de l'image

En deux dimensions, il s'agit du gradient de l'image, vecteur qui représente la variation de la fonction dépendant de plusieurs paramètres par rapport à la variation de ces paramètres.

$$\vec{\nabla} I(x, y) = \begin{pmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{pmatrix}$$

Gradient de l'image : modèle de contour

- Image $I(x, y)$. Contour = lieu des fortes variations de I .
- Gradient de I :

$$\vec{\nabla}I = \vec{G} = \begin{pmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{pmatrix}$$

- Module du gradient de I :

$$G = \|\vec{\nabla}I\| = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}$$

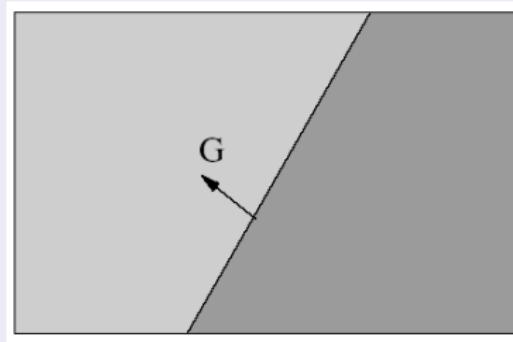
- Direction du gradient de I :

$$\vec{g} = \frac{\vec{\nabla}I}{\|\vec{\nabla}I\|}$$

Gradient de l'image : modèle de contour

Définition du contour

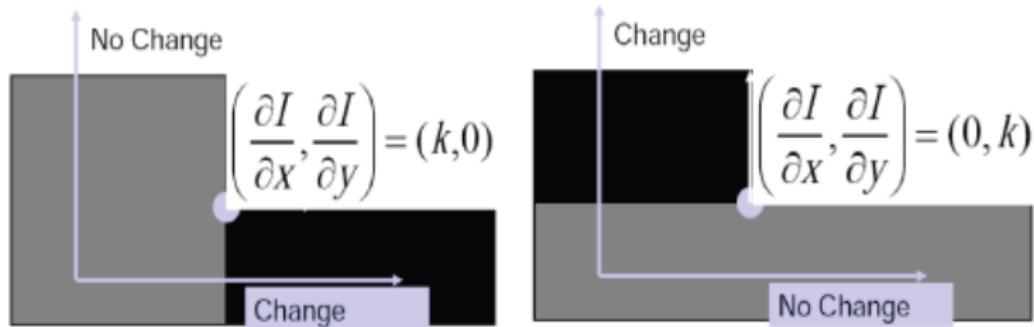
Lieux des maxima du gradient dans la direction \vec{g} .



\vec{g} est normal à la surface définie par $I(x, y)$

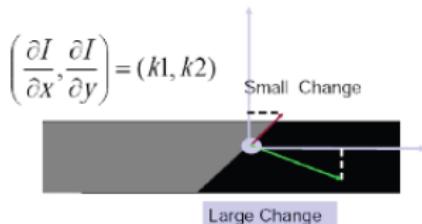
Gradient de l'image : modèle de contour

Contour vertical et contour horizontal



Gradient de l'image : norme et direction

Contour oblique



Le gradient est un vecteur perpendiculaire au contour. Son amplitude mesure la force du contour.

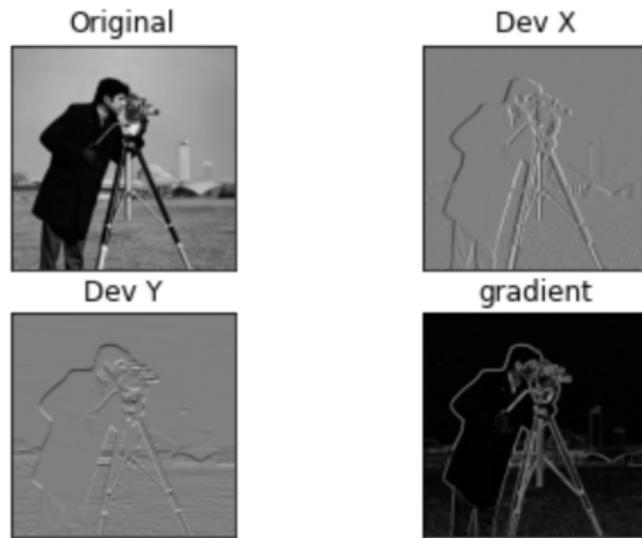
Norme

$$\|\vec{\nabla}\| = \sqrt{\partial I_x^2 + \partial I_y^2}$$

Direction

$$\theta = \arctan \left(\frac{\partial I_x}{\partial I_y} \right)$$

Gradient de l'image : exemple



Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Gradient de l'image : détection de contour

Seuillage du gradient : les points de contour dans une image sont caractérisés par des [extrema locaux du gradient](#).

Approche 1 : Principe

- ① Approximations discrètes du gradient par différences finies (convolution)
- ② Calcul de la norme du gradient en tous les points de l'image.
- ③ Maxima locaux dans la direction du gradient
- ④ Suppression de non maxima
- ⑤ Seuillage
- ⑥ Chaînage

Approche du 1er ordre : approximation du gradient

Gradient de l'image : Implémentation - Dérivée discrète

Approximations discrètes du gradient par différences finies

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

ou :

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

avec, en général $n = 1$

Calcul des dérivées par convolution de l'image avec un masque de différences.

Gradient de l'image : Implémentation - Dérivée discrète

Approximation du gradient

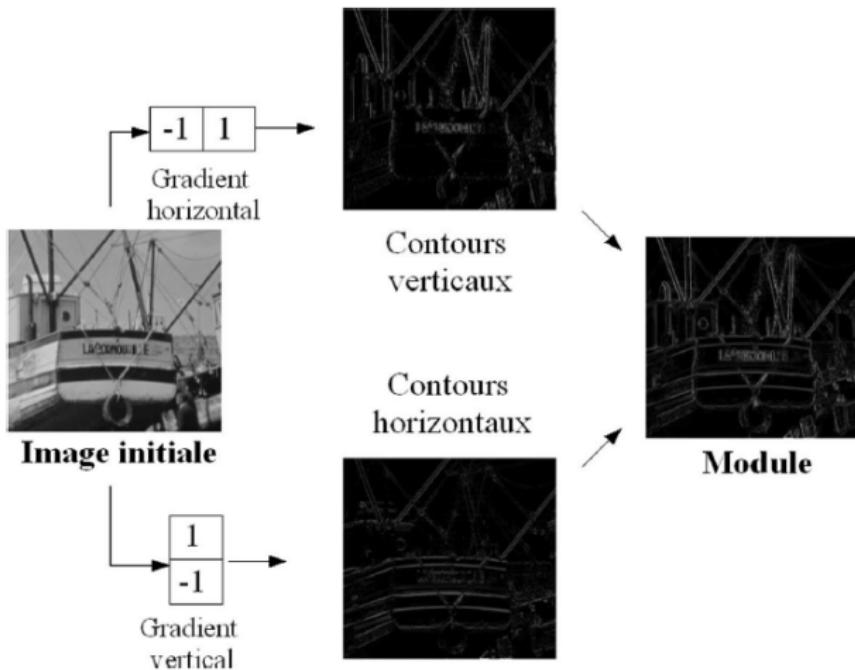
- $\nabla_x I \approx I(x, y) - I(x - 1, y)$ ($\nabla_y I \approx I(x, y) - I(x, y - 1)$)
- $\nabla_x I \approx I(x + 1, y) - I(x - 1, y)$

$$\begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array} \quad \begin{array}{|c|} \hline -1 \\ \hline 1 \\ \hline \end{array}$$

- Ou :

$$\begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline \end{array} \quad \begin{array}{|c|} \hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array}$$

Gradient de l'image : Implémentation - Dérivée discrète



©Jean-Hugh THOMAS

Filtre de Roberts

Principe

- Une première approximation de la dérivée première d'une image est proposée par Roberts en 1965.
- Gradients à $\frac{\pi}{4}$:
 - ▶ $\nabla_1 I \approx I(x+1, y+1) - I(x, y)$
 - ▶ $\nabla_2 I \approx I(x, y+1) - I(x+1, y)$
- Utilisation de 2 masques de convolution pour les 2 directions de la dérivée

1	0	0	1
0	-1	-1	0

- Forte sensibilité au bruit en raison de la taille des masques.
 - ▶ Filtre déivateur est passe-haut
 - ▶ Bruit = hautes fréquences (variation brusque)

Autres filtres pour la détection de contour

Lissage puis dérivée de l'image

Filtre de Prewitt

Filtre moyenneur + dérivée

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Filtre de Sobel (1972)

Filtre gaussien + dérivée

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

La détection des contours est moins sensible au bruit.

Détection de contours : exemple



Roberts



Prewitt



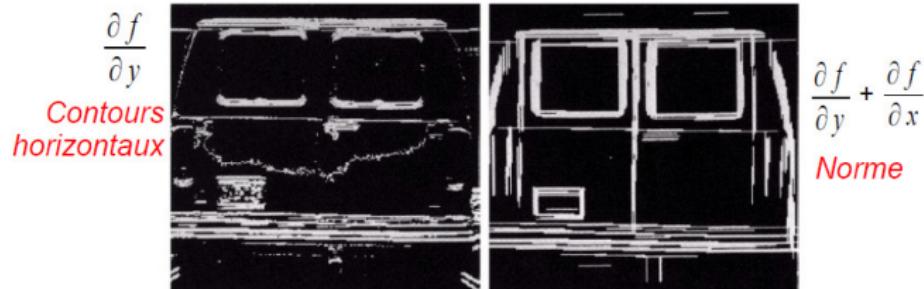
Sobel

Gradient de l'image : exemple

a
b
c
d

FIGURE 10.16

- (a) Input image.
- (b) G_y component of the gradient.
- (c) G_x component of the gradient.
- (d) Result of edge linking. (Courtesy of Perceptics Corporation.)



Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Détection de contours : seuillage

Influence du seuillage sur la détection



Sans seuillage



Avec $S = 25$



Avec $S = 60$

Détection de contours : seuillage

Le seuil peut être déterminé à partir de l'histogramme (cumulé) du module du gradient. Avantage : ce seuil est alors *adapté* au contenu de l'image.

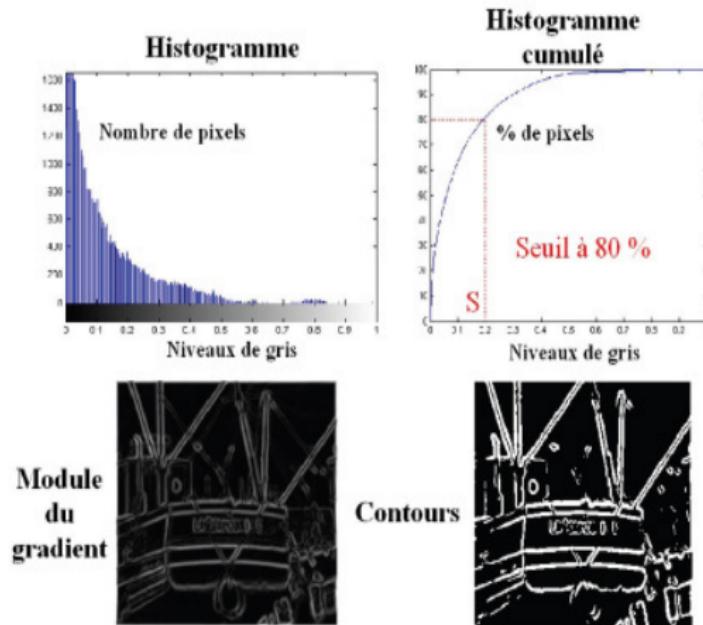


FIGURE – D'après F. Arnaud

Gradient de l'image : détection de contour

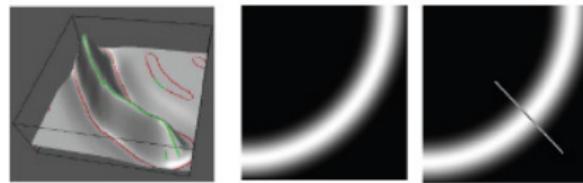
Post traitement

- ① Elimination des non maxima locaux de la norme du gradient dans sa direction.
- ② Seuillage par hysteresis.

Gradient de l'image : détection de contour

Extraction des extrema locaux dans la direction du gradient

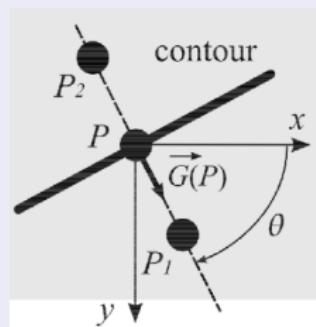
- Détermination pour un pixel p donné, des valeurs du gradient sur la droite passant par p et de direction celle de son gradient.
- On vérifie que le gradient en p est bien localement maximum sur cette droite.



Gradient de l'image : détection de contour

Extraction des extrema locaux dans la direction du gradient

- On recherche les maxima locaux dans la direction du gradient.
- P est un maximum local $\Rightarrow \|\vec{G}(P)\| > \|\vec{G}(\vec{P}_1)\|$ et $\|\vec{G}(P)\| > \|\vec{G}(\vec{P}_2)\|$



Gradient de l'image : détection de contour

Extraction des extrema locaux dans la direction du gradient : en pratique

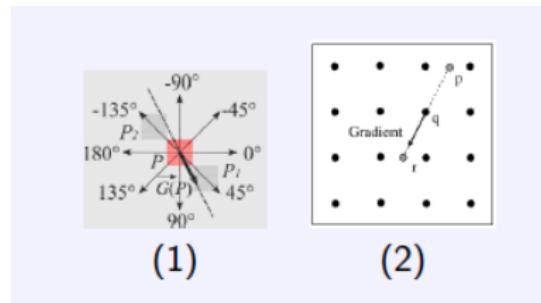
Suppression des non-maxima

- Discretisation de la direction

- ▶ On arrondit la direction à $\frac{\pi}{4}$ (8 voisins).
- ▶ On cherche les 2 pixels voisins dans la direction du gradient.

- Interpolation

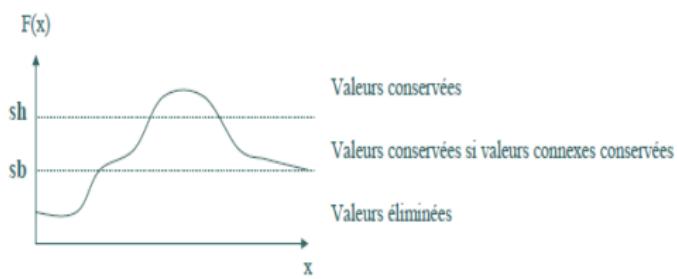
- ▶ On calcule la valeur subpixels des deux voisins.
- ▶ On interpolate la valeur de la norme du gradient en ces points.



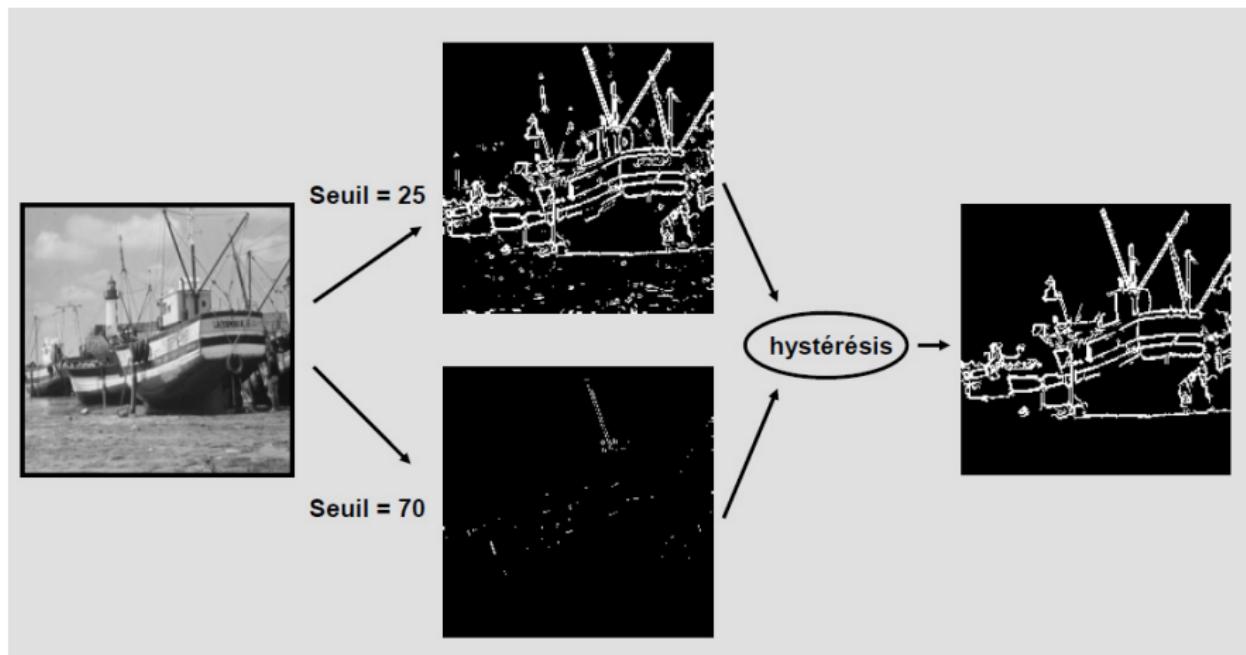
Seuillage par hysteresis

Idée

- Garder les contours les plus forts en essayant d'assurer leur continuité.
- Utilisation d'un seuil bas (s_b) et d'un seuil haut (s_h).
- Le seuil bas met en évidence des contours moins forts dans l'image.
- Ces contours moins forts ne sont conservés que si ils sont situés dans le voisinage des contours forts, i.e. si le pixel donné est connecté, par un chemin constitué de pixels dont la norme du gradient est supérieure à s_b , à un pixel pour lequel la norme du gradient est supérieure à s_h .



Seuillage par hysteresis



Détection de contours : Laplacien

Principe

- Utilisation de la dérivée seconde de l'image.
- Les contours correspondent aux passages à zéro de la dérivée seconde.
- Opérateur laplacien :

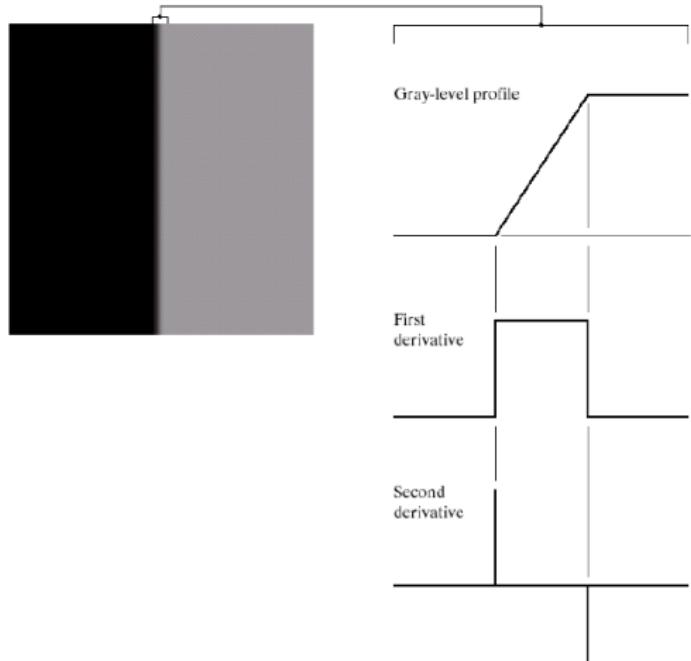
$$\Delta I = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Détection de contours : Laplacien

a b

FIGURE 10.6

- (a) Two regions separated by a vertical edge.
(b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.



Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Détection de contours : Laplacien

Approximations du laplacien par différences finies

- démonstration : série de Taylor en $I(x \pm h, y \pm h)$
- $\Delta I \approx I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) - 4I(x,y)$
- $\Delta I \approx I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) + I(x+1,y+1) + I(x-1,y-1) + I(x+1,y-1) + I(x-1,y+1) - 8I(x,y)$
- Plusieurs approximations discrètes du Laplacien :

0	1	0	ou	1	1	1
1	-4	1		1	-8	1
0	1	0		1	1	1

- Une seule matrice de convolution.
- Symétrique en rotation.

Détection de contours : Laplacien

Original

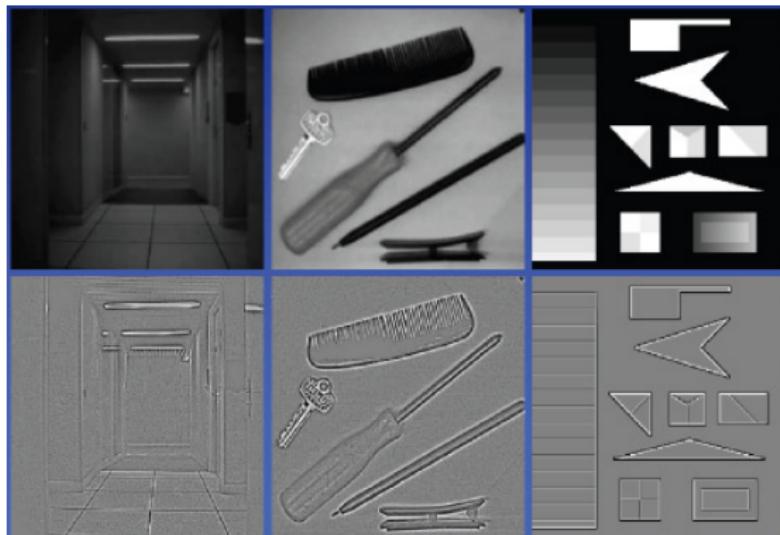


Laplacian



Image du laplacien : contours \Leftrightarrow transitions entre valeurs positives (blanc) et négatives (noir), avec un seuil sur l'amplitude.

Détection de contours : Laplacien



Détection de contours : Laplacien

	Δ_1	
Δ_2	Δ	Δ_3
	Δ_4	

Détection des passages par zéro du laplacien.

En pratique

- Si $\Delta > 0$ et l'un des autres $\Delta_i \leq 0$ ou $\Delta < 0$ et $\Delta_i \geq 0$ alors on considère qu'il y a changement de signe.
- **OU** Prendre une fenêtre centrée 3×3 sur le pixel (i,j) : le passage par 0 est détecté si $\max(I_L) > 0$, $\min(I_L) < 0$ et $\max(I_L) - \min(I_L) > S$.

Détection de contours : Laplacien



Détection de contours : Laplacien

Sensible au bruit \Rightarrow Nécessité de lisser par un filtrage passe-bas.

Lissage passe-bas

- Convolution de l'image avec un masque laplacien : $I * g * \Delta$
- Cas d'un masque gaussien : $g(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$

$$I * g * \Delta = I * (\Delta * g) = I * \Delta g$$

Les opérateurs différentiels linéaires commutent avec la convolution :

$$\Delta * g = \Delta g = \frac{4}{\sqrt{2\pi}\sigma} \left(\frac{x^2 + y^2}{2\sigma^2} - 1 \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- On convole directement I avec le masque défini par Δg : **Laplacian of Gaussian (LoG)**

Détection de contours : Laplacien

Laplacian of Gaussian

- LoG : filtre du système de vision humaine.
- LoG peut être approximé par DoG = Difference of Gaussians.

$$DoG(x, y, \sigma_1, \sigma_2) = g(x, y, \sigma_1) - g(x, y, \sigma_2)$$

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- **Approches optimales**
- Détection de formes

Détection de contours : approches optimales

Filtrage optimal

- En 1D : modélisation du contour par une marche d'escalier.
- Avec du bruit : $C(x) = A\Theta(x) + n(x)$, avec n bruit blanc gaussien.
- On cherche le filtre déivateur h tel que :

$$h * C = \Theta$$

Détection de contours : approches optimales

Dans le cas monodimensionnel, on suppose que la détection est effectuée en convoluant le signal par un filtre de réponse impulsionnel h . Les contours sont les extrêmas de la sortie du filtre (contours de type marche, bruit blanc).

Critères de Canny (1983)

Critères pour la détermination d'un filtre optimal

- **Détection** : l'opérateur doit garantir une bonne détection, i.e. une réponse forte même à de faibles contours. (critère Σ)
- **Localisation** : le contour doit être localisé avec précision (critère Λ).
- **Réponse unique** : un contour doit provoquer une seule réponse de l'opérateur.

On cherche h dérivateur qui maximise Σ et Λ sous la contrainte d'unicité de la réponse

Détection de contours : approches optimales

Principe

- Expression des critères par l'optimisation conjointe de 3 fonctionnelles permettant de définir le filtre linéaire optimal pour la détection d'un contour idéal (marche d'escalier).
- Hypothèse d'un bruit additif indépendant du signal.
- Équation différentielle dont la solution est de la forme (filtre réponse impulsionnelle finie) :

$$h(x) = a_1 e^{\alpha x} \cos(\omega x) + a_2 e^{\alpha x} \sin(\omega x) + a_3 e^{-\alpha x} \cos(\omega x) + a_4 e^{-\alpha x} \sin(\omega x)$$

Détection de contours : approches optimales

Résolution de l'équation différentielle permettant d'optimiser les critères de Canny

Les différentes approches

- Filtre gaussien et dérivée (1980, solution approchée).
- Filtre de Shen-Castan (1986, solution exacte).
- Filtre de Deriche (1987, solution exacte).

Détection de contours : approches optimales

Filtre gaussien

- Filtre de lissage gaussien dont la dérivée constitue une approximation de la solution du filtre optimal (Canny)
- Filtre de lissage gaussien a pour réponse impulsionnelle :
$$h(x) = c \exp^{-x^2/2\sigma^2}$$
- Dérivée :
$$h'(x) = -c \frac{x}{\sigma^2} \exp^{-x^2/2\sigma^2} =$$
 solution du filtre optimal (Canny).
- Passage en 2D : action de 2 filtres croisés :

$$f_x(x, y) = -x \exp^{\frac{-x^2}{2\sigma^2}} \exp^{\frac{-y^2}{2\sigma^2}}$$

$$f_y(x, y) = -y \exp^{\frac{-x^2}{2\sigma^2}} \exp^{\frac{-y^2}{2\sigma^2}}$$

J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8 :679-714, 1986.

http://www.limsi.fr/Individu/vezien/PAPIERS_ACS/canny1986.pdf



Détection de contours : approches optimales

Détecteur de Canny

- Filtrer l'image avec les dérivées de gaussienne.
- Trouver la magnitude et l'orientation du gradient.
- Suppression des non-maximas (fenêtre glissante)
- Assemblage et seuillage :
 - ▶ Définition de deux seuils : bas et haut.
 - ▶ Le seuil haut est utilisé pour démarrer le contour et le seuil bas pour le poursuivre.

Détection de contours : approches optimales

Détecteur de Canny

Effet de σ .

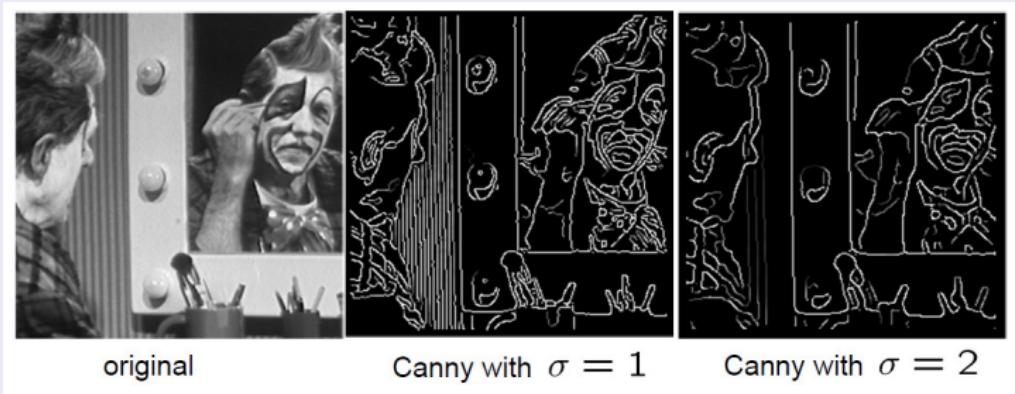


FIGURE – Source S. Seitz

- σ grand permet la détection de contours épais.
- σ petit permet la détection de contours fins.

Détection de contours : approches optimales

Filtre de Shen Castan

- Optimisation d'un critère incluant la détection et la localisation.
- Solution de la forme $h(x) = ce^{-\alpha|x|}$
- c est choisi de façon à normaliser le filtre $c = \frac{1-e^{-\alpha}}{1+e^{-\alpha}}$
- α détermine la largeur du filtre, plus α est petit, plus le lissage est important.
- Filtre de dérivation :

$$h'(x) = \begin{cases} de^{-\alpha|x|} & \text{si } x \geq 0 \\ -de^{-\alpha|x|} & \text{sinon} \end{cases}$$

avec $d = 1 - e^{-\alpha}$ choisi pour normaliser le filtre.

Détection de contours : approches optimales

Filtre de Deriche

- Filtre de lissage dont la dérivée est la solution exacte de l'équation de Canny étendue aux filtres à supports infinis.
- $$h(x) = k(\alpha|x| + 1)e^{-\alpha|x|}$$
- avec
$$k = \frac{(1-e^{-\alpha})^2}{1+2\alpha e^{-\alpha}-e^{-2\alpha}}$$
- $$h'(x) = -k'xe^{-\alpha|x|} \text{ avec } k' = \frac{(1-e^{-\alpha})^2}{e^{-\alpha}}$$
- Filtres directionnels (anisotropes).

Détection de contours : approches optimales

Filtre de Deriche : exemple



Figure 1: (a) : image originale, (b) : gradient en x (filtre de Deriche), (c) gradient en y (filtre de Deriche), (d) extrêmes locaux de la norme du gradient dans la direction du gradient.



Figure 2: Scellages des extrêmes locaux pour différentes valeurs du paramètre α du filtre de Deriche : (a) $\alpha = 0.5$, (b) $\alpha = 1.5$.

- Influence de α : facteur d'échelle.
- Faible α : peu de robustesse au bruit, bonne localisation
- Fort α : bonne robustesse au bruit, mauvaise localisation
- Adapter α au rapport signal sur bruit de l'image.

Détection de contours

Une première étape seulement pour la segmentation.

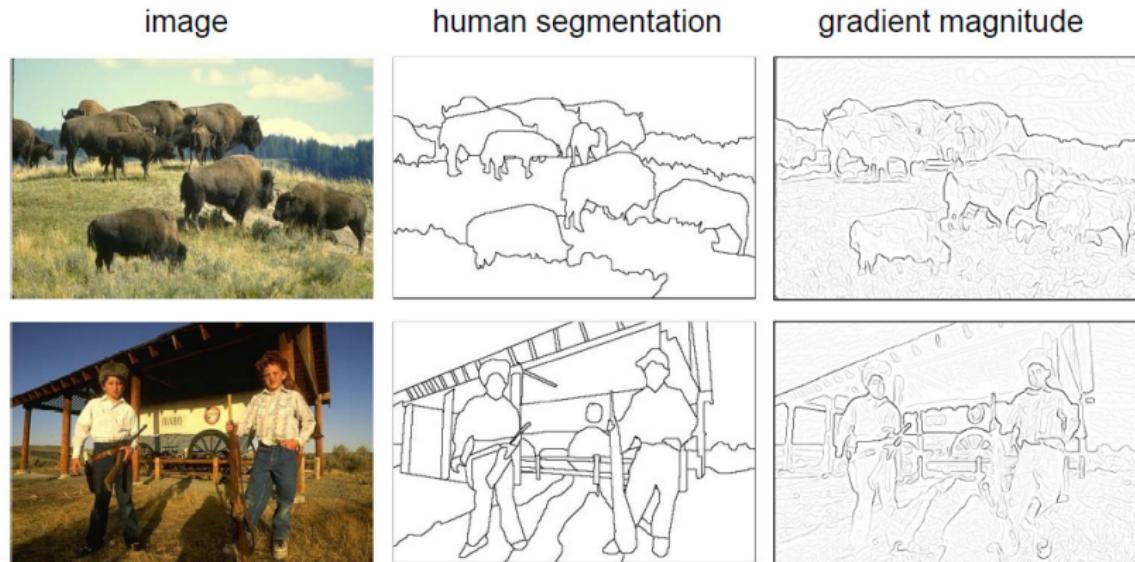


FIGURE – Source Fei Fei Li

Plan

1 Avant Propos : pré-traitement

2 Notions de base

3 Filtrage

- Quelques filtres linéaires
- Quelques filtres non-linéaires

4 Détection de contours

- Approches discrètes
- Approches optimales
- Détection de formes

Détection de contours : Transformée de Hough

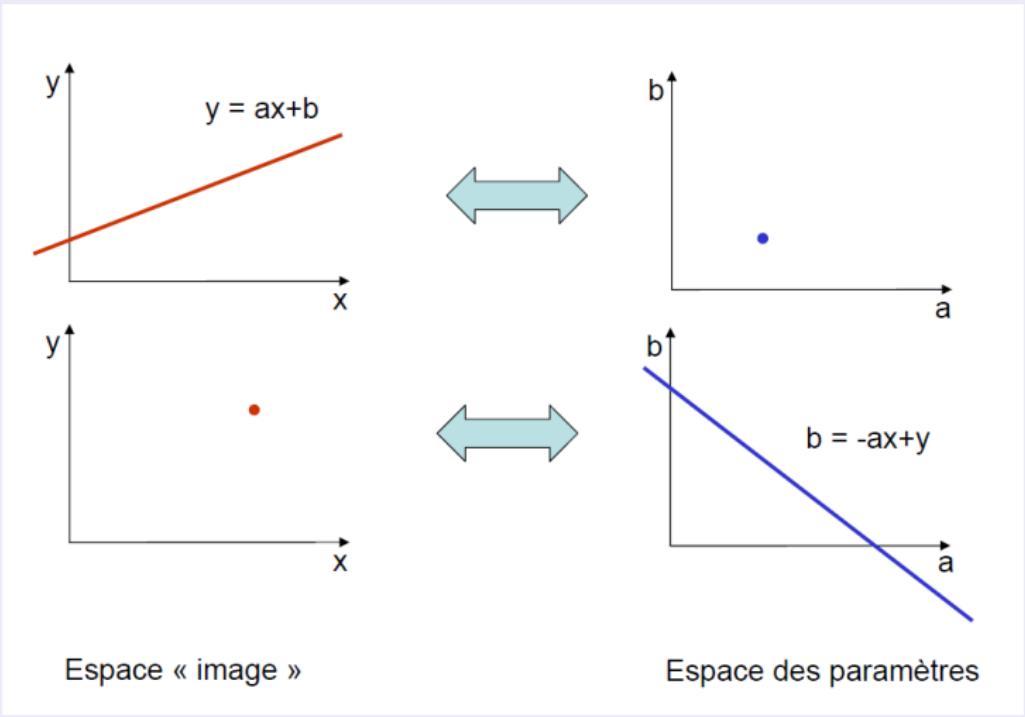
Par exemple pour la détection de droites

Principe

- Technique de reconnaissance de formes inventée en 1962 par Paul Hough.
- Approche globale pour détecter des contours continus, i.e. pas seulement des pixels contours mais le contour au complet.
- Passage du plan x-y au plan paramétrique a-b et procédure de vote.

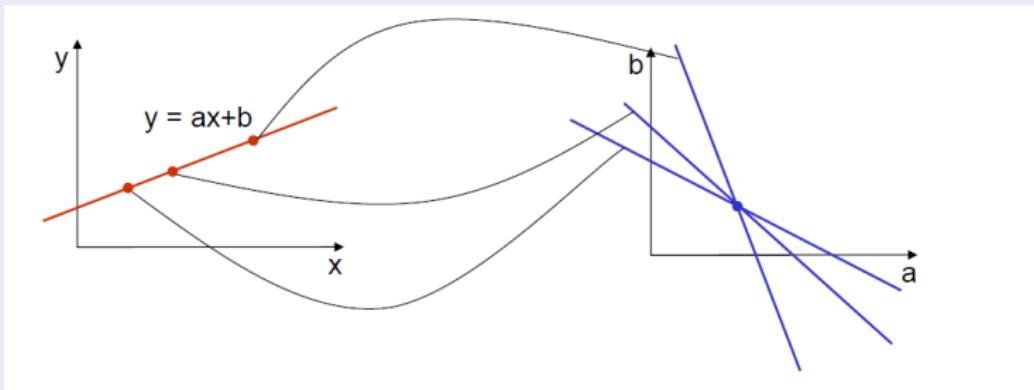
Détection de contours : Transformée de Hough

Principe



Détection de contours : Transformée de Hough

Principe



Les images de tous les points d'une droite se coupent en (a, b) dans le domaine de Hough.

Détection de contours : Transformée de Hough

Principe pour la détection de droite

- ① Appliquer une détection de contours.
- ② Discretiser le plan des paramètres (a,b).
- ③ Initialiser un accumulateur.
- ④ Pour chaque point de contour :
 - ▶ Déterminer sa droite image dans l'espace des paramètres.
 - ▶ Incrémenter l'accumulateur sur les points de cette droite
 - ★ Recherche des maxima : paramètres (a,b) de la droite. Les points de croisement des droites dans le plan $a-b$ indiquent les vraies droites existantes dans le plan $x-y$.

Détection de contours : Transformée de Hough

- **Problème** : l'espace (a, b) n'est pas adéquat (car non borné $a \rightarrow \infty, b \rightarrow \infty$)
- **Solution** : représentation sous forme polaire (ρ, θ) , $\rho = x\cos\theta + y\sin\theta$

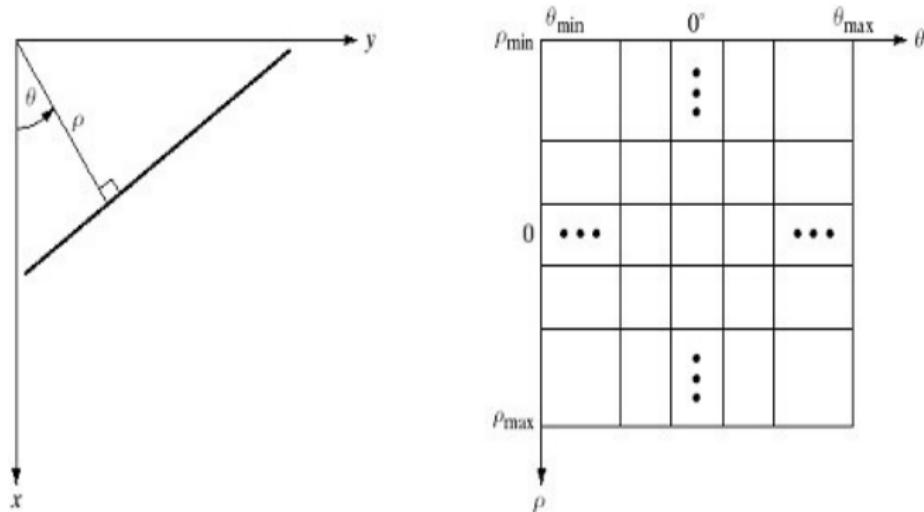


FIGURE – Source : A. Boucher

Détection de contours : Transformée de Hough

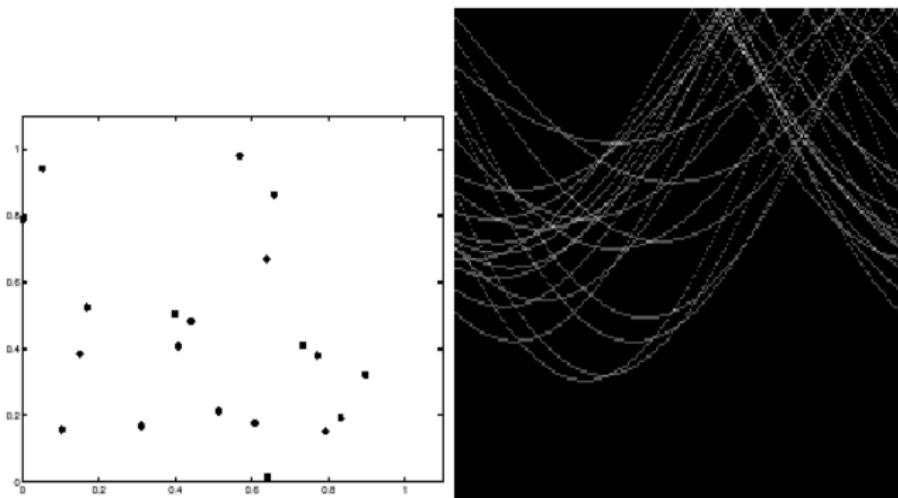


FIGURE – Source : A. Boucher

La transformée de points aléatoires ne donne rien de précis.

Détection de contours : Transformée de Hough

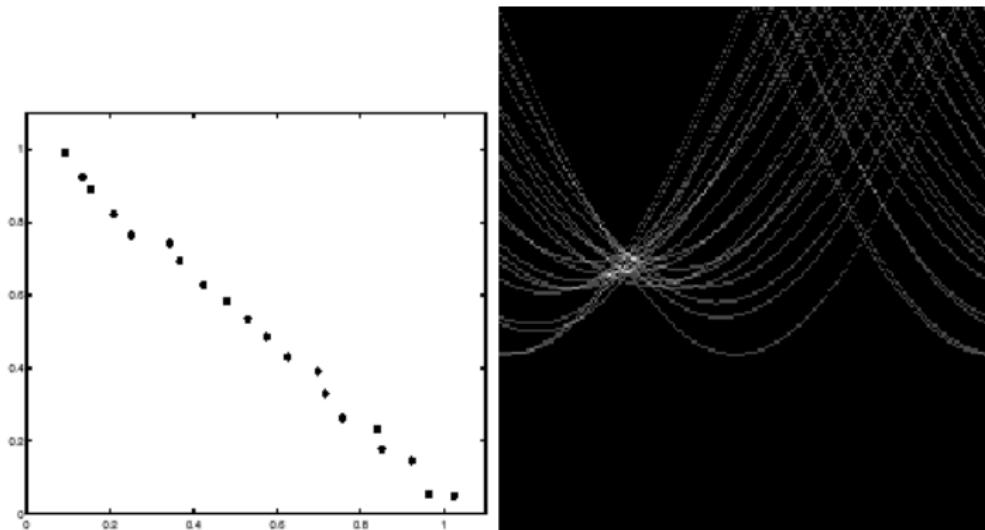


FIGURE – Source : A. Boucher

La transformée de points alignés permet de retrouver la droite.

Détection de contours : Transformée de Hough

Exemple

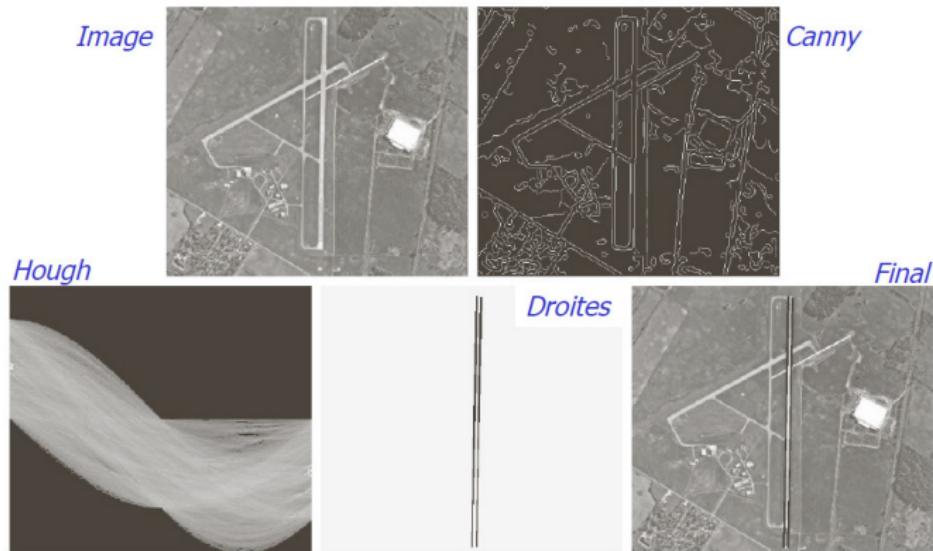


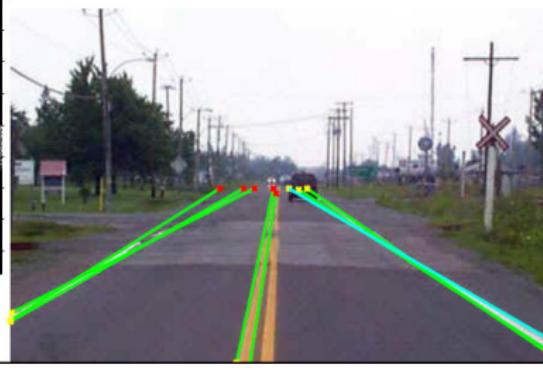
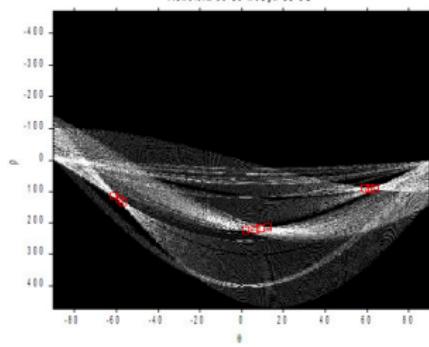
FIGURE – Source : Gonzales & Woods

Détection de contours : Transformée de Hough

Exemple



Transformée de Hough du CD



En résumé

- L'histogramme, un outil de base pour de nombreux pré-traitements.
- De nombreux pré-traitements sont aussi modélisés comme une opération de filtrage.
- Détection de contours :
 - ▶ Recherche des discontinuités dans l'image.
 - ▶ Gradient et Laplacien d'une image.
 - ▶ Approches optimales.