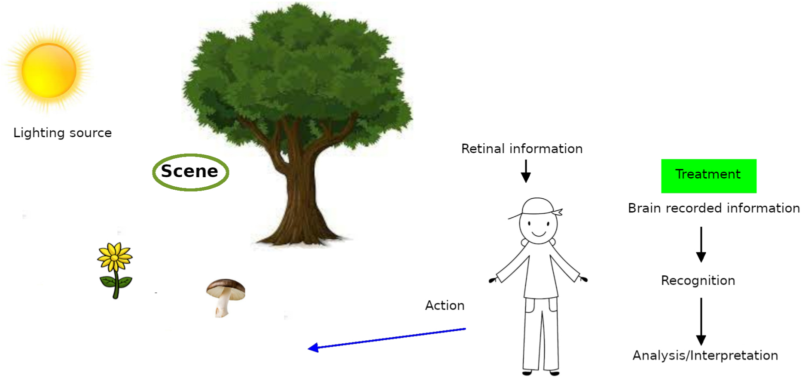
Image, Modélisation et Rendu
Parcours Multimédia, Département SN
Parcours IATI, Département 3EA
Des transformations à l'analyse d'images
Partie I

Sylvie Chambon
schambon@enseeiht.fr

28 janvier 2025

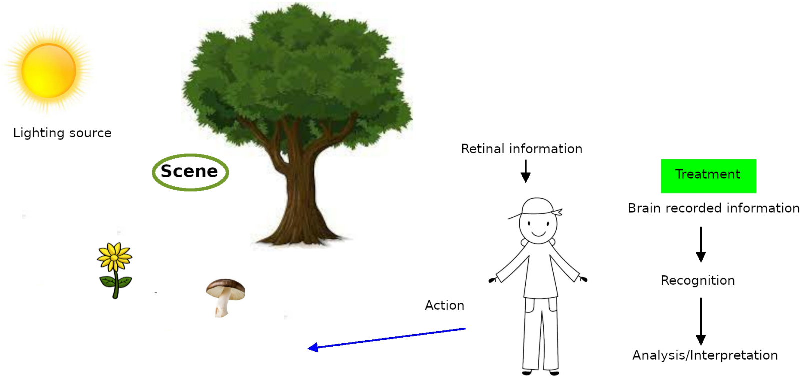
Introduction

Vision humaine



Introduction

Vision humaine



- **Performances**
  - Scène perçue ≈ 1 méga pixels en couleur et en intensité
  - Temps d'identification d'une image ≈ 100ms
  - Nombre de scènes mémorisées ≈ 100000

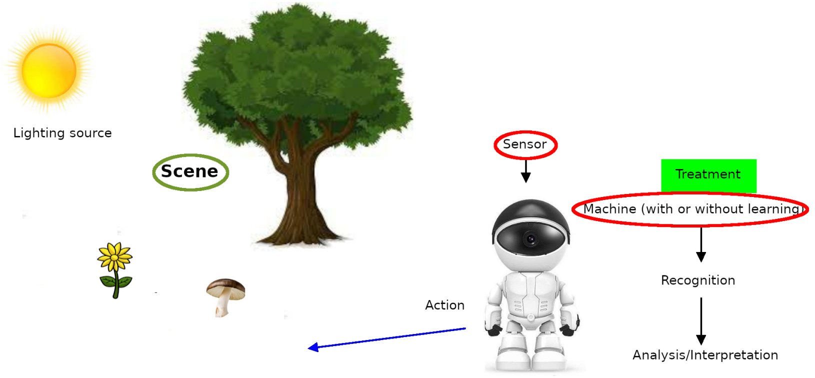
Introduction

Objectifs/Buts de la vision artificielle

- **Objectifs :** Atteindre les performances du système humain
- **Buts :** Réaliser des tâches humaines difficiles ou fastidieuses par des machines

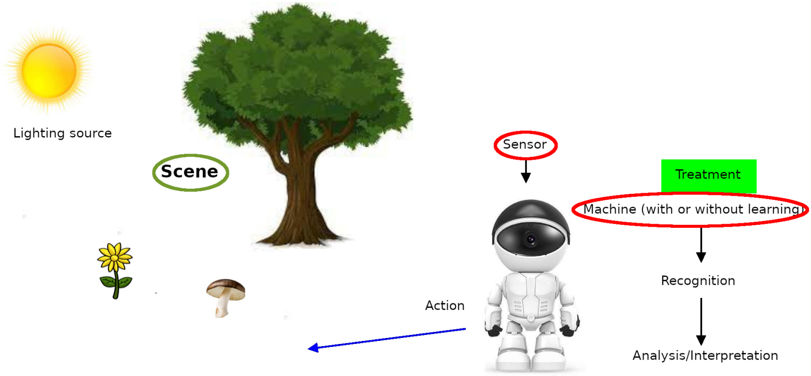
Introduction

Traitement d'images/Vision par ordinateur



Introduction

Traitement d'images/Vision par ordinateur



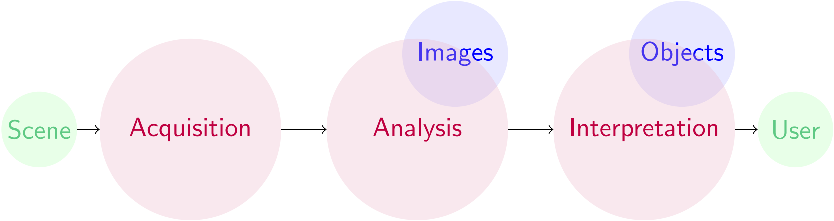
- **Dispositif :** scène + source lumineuse + capteur
- **Chaîne traitement :** acquisition, traitement (reconnaissance + interprétation), décision

Introduction

Traitement d'images/Vision par ordinateur

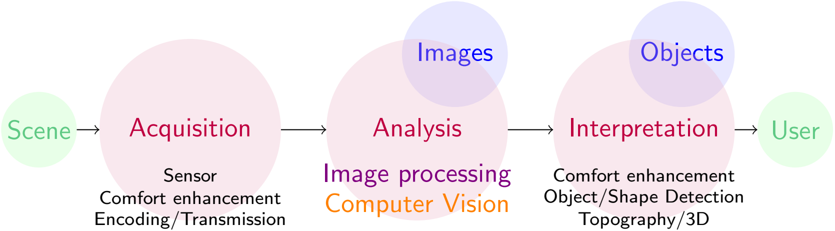
Introduction

Traitement d'images/Vision par ordinateur



Introduction

Traitement d'images/Vision par ordinateur



Introduction

Traitement d'images/Vision par ordinateur

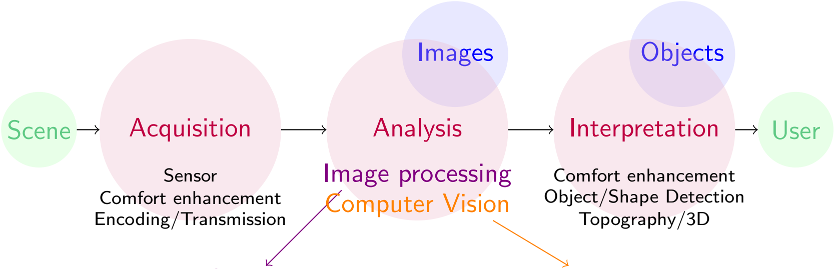


Image Transformation

Low Level (Near acquisition)

Compression / Quantization

Restoration / Enhancement

Transformation / Filtering

High Level (Near analysis)

Edge detection

Points of interest detection

Segmentation

Image comprehension

3D reconstruction

Calibration

Matching

Reconstruction

Shape recognition

Scene understanding

Introduction

Traitement d'images/Vision par ordinateur

Scene → Acquisition → Analysis → Interpretation → User

Images | Objects

Acquisition
Sensor
Comfort enhancement
Encoding/Transmission

Analysis
Image processing
Computer Vision

Interpretation
Comfort enhancement
Object/Shape Detection
Topography/3D

Image Transformation

Low Level (Near acquisition)

Compression / Quantization

Restoration / Enhancement

Transformation / Filtering

High Level (Near analysis)

Edge detection

Points of interest detection

Segmentation

Image comprehension

3D reconstruction

Calibration

Matching

Reconstruction

Shape recognition

Scene understanding

Quelle place pour l'apprentissage dans ce cours ?

- Apprentissage supervisé

Def. Programmes capables de réaliser une tâche sans la coder explicitement

- Aprentissage par l'expérience de la tâche

Ssi il existe une mesure de performance qui augmente avec l'expérience [Mitchell 1997]

- Deux éléments pour faire de l'apprentissage
  1. Une base d'apprentissage = base de donées annotées
  2. Un prédicteur qui minimise la différence

  entre les étiquettes rélles et les étiquettes prédites

- **Algorithmes connus**
  - Arbre de décisions [Quinlan 1986]
  - Forêts aléatoires [Breiman 2001]
  - Réseaux de neurones [McCulloch 1943]

Pourquoi les réseaux de neurones sont populaires depuis 2010 ?

- **Neurone**: Proche du concept en biologie
- **Perceptron**: plusieurs entrées, codées dans un vecteur $\mathbf{x}$
  pondérés, avec les poids $\mathbf{w}$
  une seule réponse y [Rosenblatt 1958]:

$$y = f(\mathbf{w}^T \mathbf{x} + b), \qquad (1)$$

avec b, le bias, f, la fonction d'activation

Rq: Modèle perceptron = Pas de résolution de problème non-linéaire [Marvin 1969]

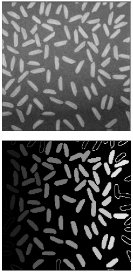
- Avec [Rumelhart 1986]: prise en compte de plusieurs couches
  - Couche d'entrée : les données
  - Couche de sortie : le résultat
  - Couches intermédiaires = couches cachées
- **Apprentissage profond** = au moins 2 couches cachées
- LeNet network [Lecun 1998] : 5 couches
- AlexNet network [Krizhevsky 2012]: dizaines de couches

Pourquoi les réseaux de neurones sont populaires depuis 2010 ?

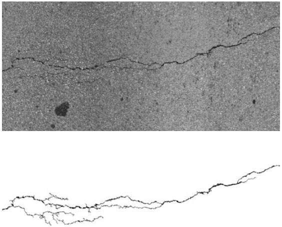
- **Difficultés des approches basées réseaux de neurones**
  1. Coûteux
  2. Bases de données annotées conséquentes
- **Ce qui a permis leur utilisation**
  1. Plus facile d'acquérir des données (capteurs)
  2. Accès facilité à des données existantes (partage et transfert)
  3. Annotation simplifiée (plateforme de crowdsourcing)
  4. Augmentation des puissances de calculs
  5. Meilleure compréhension et utilisation des fonctions d'activation
     sigmoide (pas efficace)
     reLU rectified Linear Unit [Nair 2010]
- Travaux connus [LeCun 2015] , [Goodfellow 2016]

Quand utiliser une approche par apprentissage profond ?

Est-ce que mon problème est bien défini ?
Est-ce que les objets que j'étudie ont des caractéristiques explicites ?



Détection de contours



Sélection par recherche de chemins minimaux

Objectifs d'apprentissage

(1) **Connaître les outils fondamentaux pour analyser et transformer une image**

1. Savoir étudier la dynamique des niveaux de gris (histogramme)
2. Connaître les outils de convolution et de filtrage
3. Savoir utiliser les filtres de convolution

(2) **Découvrir et apprendre les éléments liés à la détection de contours**

1. Apprendre à construire un filtre de convolution pour calculer la dérivée
2. Connaître l'algorithme classique de détection de contours.

(3) **Découvrir et apprendre différentes techniques de segmentation**

1. Savoir distinguer les approches fond/forme des approches pour plus de deux régions
2. Apprendre et distinguer les différentes catégories de méthodes de segmentation
3. Connaître en détails et savoir coder la méthode des k-moyennes

Objectifs d'apprentissage

(4) **Savoir évaluer qualitativement et quantitativement un résultat de segmentation**
   1. Comprendre la notion de segmentation de référence
   2. Connaître les différentes mesures d'évaluation

(5) **Comprendre les méthodes de sur-segmentation appelées superpixels**
   1. Comprendre la différence entre segmentation et sur-segmentation
   2. Savoir analyser les différents éléments d'une méthode de sur-segmentation : attributs et propriétés utilisées ainsi que méthode de construction
   3. Coder une approche connue : SLIC

(6) Partie pratique : **Concevoir une chaîne de traitement complète pour segmenter une image**

Objectifs d'apprentissage : Conclusion

Vous devez être capable d'analyser une image donnée en :

- Identifiant les difficultés de la scène en terme d'extraction de contours ou de segmentation
- Proposant des solutions que vous savez justifier

Temps de travail

- Introduction : 10 minutes
- Chapitre 1 : 1 ou 2h
- Chapitre 2 : 1 ou 2h
- Chapitre 3 : 1 ou 2h
- Chapitre 4 : 1 ou 2h
- Chapitre 5 et 6 (en bonus) : 1h

Introduction

Notations/Définitions

Binaire	Niveau de gris	Couleur	3D
			

Introduction

Représentation d'une image : du continu vers le discret

En continu:

$$\begin{aligned} I \ : \ \mathbb{R}^n &\longrightarrow \ \mathbb{R}^m \\ \mathbf{p} &\longmapsto \ I(\mathbf{p}), \end{aligned}$$

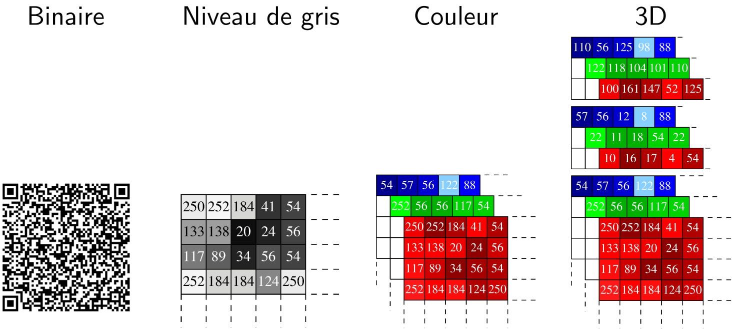
Introduction

Représentation d'une image : du continu vers le discret

Type	Représentation	Machine	Oeil humain
BINAIRE	$I : \mathbb{R}^2 \longrightarrow [0;1]$	noir = 0, blanc = 1	
NIVEAU DE GRIS	$I : \mathbb{R}^2 \longrightarrow [0,255]$	256 niveaux (1 octet)	64 niveaux
COULEUR	$I : \mathbb{R}^2 \longrightarrow [0,255] \times [0,255] \times [0,255]$	16 millions de couleurs (3 octets)	350000 couleurs
3D	$I : \mathbb{R}^3 \longrightarrow [0,255]$ OU $[0,255] \times [0,255] \times [0,255]$		

Introduction

Représentation d'une image : du continu vers le discret



Introduction

Notations

Nous utilisons souvent le terme norme et en particulier nous utilisons les normes L_p définies par :

$$\mathrm{L}_P((x_1, y_1), (x_2, y_2)) = \left(|x_1 - x_2|^P + |y_1 - y_2|^P\right)^{\frac{1}{P}}. \tag{2}$$

Nous avons donc L_1 qui est la somme des différences en valeur absolue et qui est également appelée *city-block*. La norme L_2 est la distance euclidienne.

Introduction

Suite des notations

- I : une image qui à chaque point/pixel $\mathbf{p}$ associe un niveau de gris noté $I(x, y)$;
- $I_x, I_y, I_{xx}, I_{yy}, I_{xy}$: images des dérivées premières et secondes de l'image ;
- $I_x(x, y), \ldots, I_{xy}(x, y)$: valeurs des dérivées premières et secondes au point $\mathbf{p}$;
- Pour simplifier, on notera I_x pour $I_x(x, y)$ ...
- Lissage : à définir
- Padding : à définir

Tranformations d'images

- Types de transformation
- Transformations ponctuelles
- Transformations locales

Détection de contours

- Définitions et notion de dérivée
- Algorithme de détection basique
- Calcul de dérivées premières
- Calcul des dérivées secondes

Plan de la présentation

Objectifs

- Améliorer la visualisation
- Extraire une information de plus haut niveau : analyse, voire interprétation, de la scène étudiée

Classement des types de transformation (1)

Nous parlons ici des filtres utilisés, et non des images acquises. Il n'y a donc aucune image spatiale ou fréquentielle.

- Locales/globales
- Spatiales : histogrammes, filtrage, convolution

  versus fréquentielles : Fourier

Classement des types de transformation (2)

	SPATIALE	FRÉQUENTIELLE	MORPHOLOGIQUE
Ponctuelle	*Look Up Table / Histogrammes*	-	-
Locale	Filtrage/Convolution	Filtrage/Convolution	Morphologie mathématique
Globale	Filtrage/Convolution	Fourier	-

Classement des types de transformation (2)

	SPATIALE	FRÉQUENTIELLE	MORPHOLOGIQUE
Ponctuelle	*Look Up Table* / **Histogrammes**	-	-
Locale	**Filtrage/Convolution**	Filtrage/Convolution	Morphologie mathématique
Globale	Filtrage/Convolution	Fourier	-

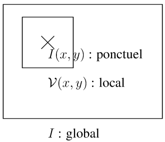
Classement des types de transformation

- Passe-bas : atténue le bruit et les détails (lissage)
- Passe-haut : accentue les contours et les détails (amplifie le bruit)
- Passe-bande : compromis

Quelques notations (1)

I' : image transformée, T : transformation

- **Ponctuelle :** $I^{'}(x,y) = T(I(x,y))$
- **Locale :** $I^{'}(x,y) = T(\mathcal{V}(x,y))$ avec $\mathcal{V}(x,y)$, voisinage de $\mathbf{p}(x,y)$ (notion de **connexité**)
- **Globale :** $I^{'}(x,y) = T(I)$



En global, le résultat est bien une image (Transformée de Fourier).

Quelques notations (2)

Notion de voisinage de p – Notion de connexité

- Ensemble de pixels $\mathbf{p}^{'}$ à une distance d donnée

  Tels qu'il existe un chemin de $\mathbf{p}$ à $\mathbf{p}^{'}$ tel que les pixels qui le constituent appartiennent à $\mathcal{V}(x, y)$.

Quelques notations (2)

Notion de voisinage de p – Notion de connexité

- Ensemble de pixels $\mathbf{p}^{'}$ à une distance d donnée

  Tels qu'il existe un chemin de $\mathbf{p}$ à $\mathbf{p}^{'}$ tel que les pixels qui le constituent appartiennent à $\mathcal{V}(x, y)$.
- d peut être
  - une distance euclidienne (L_2)
  - de type *city-block* (L_1)
  - ou encore *chess board* ou distance de Chebyshev

    $L_\infty = \max\left((|x - x^{'}|, |y - y^{'}|\right)$

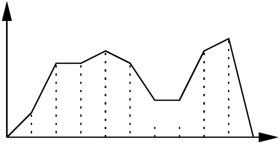
Plan de la présentation

Transformations ponctuelles

Notions d'histogramme

Transformations ponctuelles

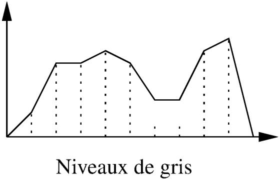
Notions d'histogramme



Histogramme

Transformations ponctuelles

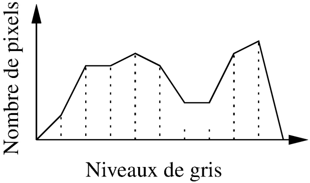
Notions d'histogramme



Histogramme

Transformations ponctuelles

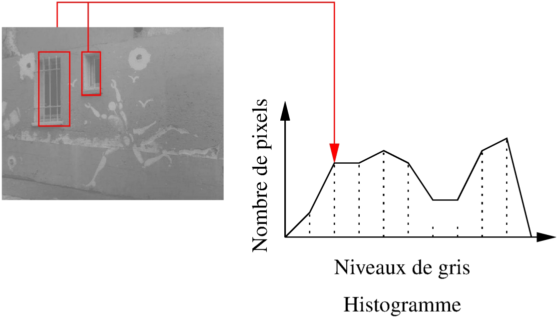
Notions d'histogramme



Histogramme

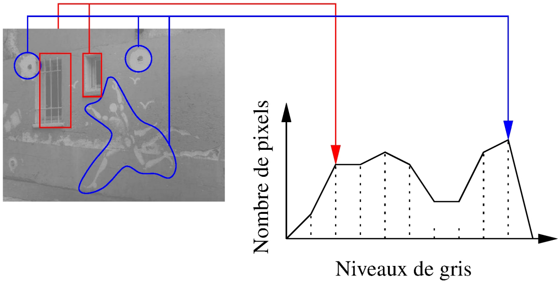
Transformations ponctuelles

Notions d'histogramme



Transformations ponctuelles

Notions d'histogramme



Histogramme

Transformations ponctuelles

Définition 1 : Histogramme

Un histogramme fournit le nombre de pixels pour chaque niveau de gris, sachant que l'image possède N pixels :

$$\begin{aligned} H \ : \ [0, N_{\max}] &\longrightarrow \ [0, N] \\ i &\longmapsto \ H(i) \text{ où } H(i) = \#\{(x,y) | I(x,y) = i\} \end{aligned}$$

Transformations ponctuelles

Utilisation de l'histogramme

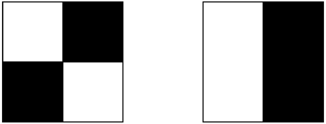
- Permet d'étudier la dynamique de l'image (le **contraste**)
- Donne une vue d'ensemble de la **distribution des niveaux de gris**
- Permet de mettre en évidence les **populations significatives dans l'image**

Transformations ponctuelles

Utilisation de l'histogramme

- Permet d'étudier la dynamique de l'image (le **contraste**)
- Donne une vue d'ensemble de la **distribution des niveaux de gris**
- Permet de mettre en évidence les **populations significatives dans l'image**
- **Par contre, l'histogramme ne permet pas de localiser les populations d'intérêt**

Image 1 Image 2



Transformations ponctuelles

Définition 2 : Histogramme normalisé

$$H_n(i) = \frac{H(i)}{N}$$

Cadre probabiliste : densité de probabilité

Transformations ponctuelles

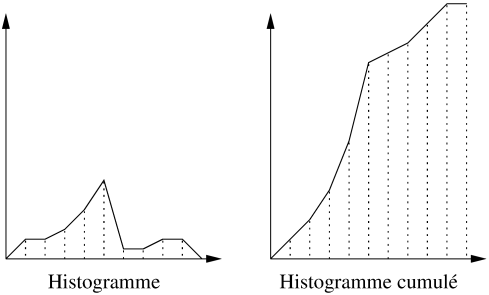
Définition 3 : Histogramme cumulé (1)

Nombre de pixels inférieurs à chaque niveau de gris

$$\begin{aligned} H_c \ : \ [0, N_{\max}] &\longrightarrow \ [0, N] \\ i &\longmapsto \ H_c(i) \text{ où } H_c(i) = \sum_{k=0}^{i} H(k) \end{aligned}$$

Transformations ponctuelles

Histogramme cumulé



Transformations ponctuelles

Définition 3 : Histogramme cumulé (2)

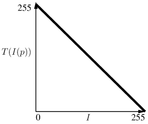
- **Propriétés**
  - Toujours croissant
  - $H_c(N_{\max}) = \sum_{k=0}^{N_{\max}} H(k) = N$
- **Propriétés de transition**

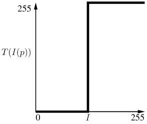
$$\begin{cases} H_c(0) = H(0) \\ H_c(i) = H_c(i-1) + H(i) \quad \forall i \in [1, N_{\max}] \end{cases}$$

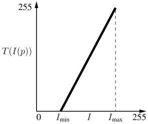
$$\begin{cases} H(0) = H_c(0) \\ H(i) = H_c(i) - H_c(i-1) \quad \forall i \in [1, N_{\max}] \end{cases}$$

Transformations ponctuelles

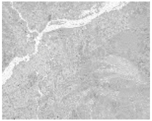
Transformations d'histogrammes







Image



Inversion

Binarisation

Étalement

Comment choisir le seuil de binarisation ?

Transformations ponctuelles

Transformations d'histogrammes

Inversion

$T(i) = N_{\max} - i$, avec $N_{\max}$, le niveau de gris maximal

Transformations ponctuelles

Transformations d'histogrammes

Inversion

$T(i) = N_{\text{max}} - i$, avec N_{max}, le niveau de gris maximal

Seuillage/Binarisation

$$T(i) = \begin{cases} 1 & \text{Si } i > S \\ 0 & \text{Sinon.} \end{cases}$$, avec S, un seuil à choisir

Transformations ponctuelles

Approche Otsu pour déterminer le seuil (1)

- **Objectif :** Déterminer automatiquement le seuil k séparant au mieux deux classes/objets présents C_0 et C_1
- **Principe utilisé :** Minimiser la variance au sein de chaque classe
- **Propriété utilisée :** Lien entre variance totale, variance intra-classe et variance inter-classe

Transformations ponctuelles

Approche Otsu pour déterminer le seuil (2)

- **Notations**
  - μ_0, μ_1 : moyennes des classes
  - w_0, w_1 : poids des classes
  - μ_T : moyenne des niveaux de gris sur l'ensemble de l'image
- **Définitions importantes**

$$w_0\mu_0 + w_1\mu_1 = \mu_T \quad \text{avec } w_0 + w_1 = 1 \tag{3}$$

$$\sigma^2_{intra} = w_0\sigma_0^2 + w_1\sigma_1^2 \tag{4}$$

$$\sigma^2_{inter} = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 = w_0 w_1(\mu_1 - \mu_0)^2 \tag{5}$$

Transformations ponctuelles

Approche Otsu pour déterminer le seuil (3)

Propriété importante

$$\sigma^2_{intra} + \sigma^2_{inter} = \sigma^2_T, \tag{6}$$

Conséquence : **minimiser la variance intra classe revient à maximiser la variance inter classe**

$$\begin{aligned} k &= argmin_k w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 \\ &= argmax_k w_0 w_1(\mu_1 - \mu_0)^2 \end{aligned} \tag{7}$$

Transformations ponctuelles

Approche Otsu pour déterminer le seuil (4)

- Algorithme simple : parcourir l'ensemble des seuils possibles
- Algorithme rapide : 255 niveaux de gris à parcourir
- Astuce algorithmique : modifier les moyennes et variances successives par un simple jeu d'ajouts/suppressions des valeurs ajoutées/supprimées

Transformations ponctuelles

Approche par mélange de gaussienne pour déterminer le seuil (1)

- Une image bimodale peut être vue comme l'estimation de la fonction densité de probabilité de l'intensité : $p(x)$.
- Densité = superposition de deux densités unimodales correspondant aux deux régions à segmenter : $p(x) = P_1 p_1(x) + P_2 p_2(x)$, avec p_1 et p_2, deux densités de probabilité gaussiennes, centrées respectivement en μ_1 et μ_2 (avec $\mu_1 < \mu_2$), et d'écart type respectif σ_1 et σ_2.
- $P_1 + P_2 = 1$.

Transformations ponctuelles

Approche par mélange de gaussienne pour déterminer le seuil (2)

- Un seuil T peut être choisi de manière à séparer les deux modes.
- Erreur de classification :

$$E(T) = P_2 E_1(T) + P_1 E_2(T),$$

avec E_1, la probabilité de se tromper en classant 2 en 1, et *vice versa*

- Minimisation atteinte lorsque : $P_1 E_1(T) = P_2 E2(T)$.
- Résultat classique pour une même variance σ :

$$S = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 + \mu_2} \ln\left(\frac{P_2}{P_1}\right).$$

conséquence de l'expression de minimisation en utilisant la définition de la fonction gaussienne

Transformations ponctuelles

Étalement/Modification de la dynamique

1. On a : $i \in [I_{\min}, I_{\max}]$.

Transformations ponctuelles

Étalement/Modification de la dynamique

1. On a : $i \in [I_{\min}, I_{\max}]$.
2. On souhaite : $i \in [0, N_{\max}]$.

Transformations ponctuelles

Étalement/Modification de la dynamique

1. On a : $i \in [I_{\min}, I_{\max}]$.
2. On souhaite : $i \in [0, N_{\max}]$.
3. On suppose une transformation affine : $T(i) = ai + b$.

Transformations ponctuelles

Étalement/Modification de la dynamique

1. On a : $i \in [I_{\min}, I_{\max}]$.
2. On souhaite : $i \in [0, N_{\max}]$.
3. On suppose une transformation affine : $T(i) = ai + b$.
4. On sait que $T(I_{\min}) = 0$ et $T(I_{\max}) = N_{\max}$.
   D'où (1) $aI_{\min} + b = 0$ et (2) $aI_{\max} + b = N_{\max}$
   (1) - (2) donne $a = \frac{N_{\max}}{I_{\max} - I_{\min}}$

Transformations ponctuelles

Étalement/Modification de la dynamique

1. On a : $i \in [I_{\min}, I_{\max}]$.
2. On souhaite : $i \in [0, N_{\max}]$.
3. On suppose une transformation affine : $T(i) = ai + b$.
4. On sait que $T(I_{\min}) = 0$ et $T(I_{\max}) = N_{\max}$.
   D'où (1) $aI_{\min} + b = 0$ et (2) $aI_{\max} + b = N_{\max}$
   (1) - (2) donne $a = \frac{N_{\max}}{I_{\max} - I_{\min}}$
5. On obtient : $T(i) = \frac{N_{\max}}{I_{\max} - I_{\min}}(i - I_{\min})$.

Transformations ponctuelles

Recadrage de la dynamique : propriétés

- Même raisonnement par morceaux (type d'objets)
- Étalement préserve la forme de l'histogramme
- Ré-hausse le contraste tout en préservant le degré d'importance/de saillance de chaque objet présent dans la scène
- **L'histogramme étalé obtenu est clairsemé car le nombre de niveaux de gris ne change pas.**

Transformations ponctuelles

Recadrage de la dynamique : illustrations

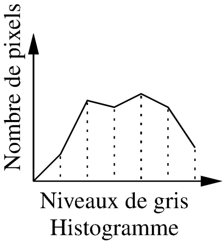
Image originale

Transformations ponctuelles

Recadrage de la dynamique : illustrations

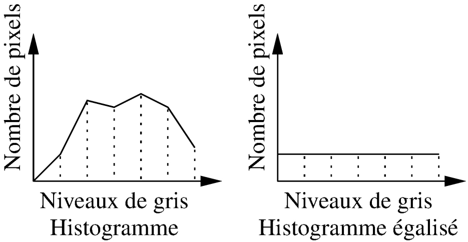
Transformations ponctuelles

Égalisation : principe



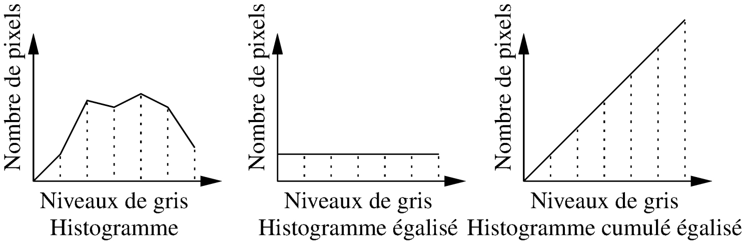
Transformations ponctuelles

Égalisation : principe



Transformations ponctuelles

Égalisation : principe



Transformations ponctuelles

Égalisation : définition

Transformation de l'histogramme cumulé

1. T telle que : $H_c^{'}(T(i)) = H_c(i)$ où $H_c^{'}$ histogramme souhaité

Transformations ponctuelles

Égalisation : définition

Transformation de l'histogramme cumulé

1. T telle que : $H_c^{'}(T(i)) = H_c(i)$ où $H_c^{'}$ histogramme souhaité
2. De plus, $\boxed{H_c^{'}(i) = \frac{N}{N_{\max}} \times (i+1)}$ où N est le nombre de pixels de l'image et $N_{\max}$ le nombre de niveaux de gris.

Transformations ponctuelles

Égalisation : définition

Transformation de l'histogramme cumulé

1. T telle que : $H_c^{'}(T(i)) = H_c(i)$ où $H_c^{'}$ histogramme souhaité
2. De plus, $\boxed{H_c^{'}(i) = \frac{N}{N_{\max}} \times (i+1)}$ où N est le nombre de pixels de l'image et $N_{\max}$ le nombre de niveaux de gris.
3. Ainsi, $H_c^{'}(T(i)) = \frac{N}{N_{\max}} \times (T(i)+1)$.

Transformations ponctuelles

Égalisation : définition

Transformation de l'histogramme cumulé

1. T telle que : $H_c^{'}(T(i)) = H_c(i)$ où $H_c^{'}$ histogramme souhaité
2. De plus, $\boxed{H_c^{'}(i) = \frac{N}{N_{\max}} \times (i+1)}$ où N est le nombre de pixels de l'image et $N_{\max}$ le nombre de niveaux de gris.
3. Ainsi, $H_c^{'}(T(i)) = \frac{N}{N_{\max}} \times (T(i)+1)$.
4. On en déduit que $\frac{N}{N_{\max}}(T(i)+1) = H_c(i)$

Transformations ponctuelles

Égalisation : définition

Transformation de l'histogramme cumulé

1. T telle que : $H_c^{'}(T(i)) = H_c(i)$ où $H_c^{'}$ histogramme souhaité
2. De plus, $\boxed{H_c^{'}(i) = \frac{N}{N_{\max}} \times (i+1)}$ où N est le nombre de pixels de l'image et $N_{\max}$ le nombre de niveaux de gris.
3. Ainsi, $H_c^{'}(T(i)) = \frac{N}{N_{\max}} \times (T(i)+1)$.
4. On en déduit que $\frac{N}{N_{\max}}(T(i)+1) = H_c(i)$
5. D'où : $T(i) = \begin{cases} 0 & \text{Si } H_c(i) < \frac{N_{\max}}{N} \\ \frac{N_{\max} H_c(i)}{N} - 1 & \text{Sinon} \end{cases}$

Transformations ponctuelles

Égalisation : définition

Transformation de l'histogramme cumulé

1. T telle que : $H_c^{'}(T(i)) = H_c(i)$ où $H_c^{'}$ histogramme souhaité
2. De plus, $\boxed{H_c^{'}(i) = \frac{N}{N_{\max}} \times (i+1)}$ où N est le nombre de pixels de l'image et $N_{\max}$ le nombre de niveaux de gris.
3. Ainsi, $H_c^{'}(T(i)) = \frac{N}{N_{\max}} \times (T(i)+1)$.
4. On en déduit que $\frac{N}{N_{\max}}(T(i)+1) = H_c(i)$
5. D'où : $T(i) = \begin{cases} 0 & \text{Si } H_c(i) < \frac{N_{\max}}{N} \\ \frac{N_{\max} H_c(i)}{N} - 1 & \text{Sinon} \end{cases}$

Problème : Cette transformation n'est pas définie si $\boxed{H_c(i) < \frac{N}{N_{\max}}}$.

Transformations ponctuelles

Égalisation : définition

Transformation de l'histogramme cumulé

1. T telle que : $H_c^{'}(T(i)) = H_c(i)$ où $H_c^{'}$ histogramme souhaité
2. De plus, $\boxed{H_c^{'}(i) = \frac{N}{N_{\max}} \times (i+1)}$ où N est le nombre de pixels de l'image et $N_{\max}$ le nombre de niveaux de gris.
3. Ainsi, $H_c^{'}(T(i)) = \frac{N}{N_{\max}} \times (T(i)+1)$.
4. On en déduit que $\frac{N}{N_{\max}}(T(i)+1) = H_c(i)$
5. D'où : $T(i) = \begin{cases} 0 & \text{Si } H_c(i) < \frac{N_{\max}}{N} \\ \frac{N_{\max} H_c(i)}{N} - 1 & \text{Sinon} \end{cases}$

Problème : Cette transformation n'est pas définie si $\boxed{H_c(i) < \frac{N}{N_{\max}}}$.

Conséquence : L'histogramme égalisé ne sera pas égal à l'histogramme théorique attendu, c'est-à-dire que tous les éléments de l'histogramme égalisés ne seront pas égaux. .

Transformations ponctuelles

Égalisation : propriétés

- À l'inverse de l'étalement, l'égalisation réhausse le contraste en faisant apparaître chaque population avec la même importance (c'est ce que nous avons voulu dire par chaque objet a le même niveau de contraste).
- **L'histogramme obtenu est clairsemé car le nombre de niveau de gris utilisé n'est pas augmenté, voire, il est diminué.**
- **La transformation calculée n'étant pas définie pour certaines valeurs de niveaux de gris, certains niveaux de gris sont perdus et nous obtenons donc des valeurs qui ne sont pas strictement égales d'un niveau de gris à l'autre.**

Transformations ponctuelles

Égalisation : illustrations et comparaisons

Image originale

Transformations ponctuelles

Égalisation : illustrations et comparaisons

Étalement de la dynamique

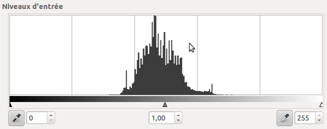
Transformations ponctuelles

Égalisation : illustrations et comparaisons

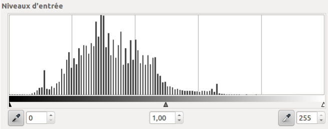
Égalisation

Transformations ponctuelles

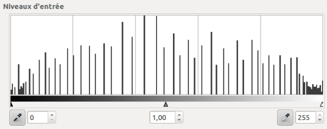
Égalisation : illustrations et comparaisons



Histogramme de l'image d'origine



Étalement de la dynamique



Égalisation

Transformations locales

Filtrage

- Amélioration de l'image
- Interprétation de l'image

Plan de la présentation

Tranformations d'images
- Types de transformation
- Transformations ponctuelles
- Transformations locales

Détection de contours

Transformations locales

Filtres linéaires : convolution

- **Principe :** le niveau de gris du pixel devient une somme pondérée des niveaux de gris de ses voisins

Transformations locales

Filtres linéaires : convolution

- **Principe :** le niveau de gris du pixel devient une somme pondérée des niveaux de gris de ses voisins
- **En continu**

$$I^{'}(x,y) = (f * I)(x,y) \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x^{'},y^{'})I(x-x^{'},y-y^{'})dx^{'}dy^{'}. \tag{8}$$

Transformations locales

Filtres linéaires : convolution

- **Principe :** le niveau de gris du pixel devient une somme pondérée des niveaux de gris de ses voisins
- **En continu**

$$I^{'}(x,y) = (f * I)(x,y) \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x^{'},y^{'}) I(x-x^{'},y-y^{'}) dx^{'} dy^{'}. \tag{8}$$

- **En discret**

$$I^{'}(x,y) = \sum_{x^{'}=-k}^{k} \sum_{y^{'}=-l}^{l} F(k-x^{'},l-y^{'}) I(x+x^{'},y+y^{'}). \tag{9}$$

Transformations locales

Notion de masque / de filtre

- Un masque ou un filtre = une région de taille strictement inférieure à l'image
- Ce masque ou ce filtre va nous permettre d'étudier l'image en parcourant toutes (ou une partie) des régions qui la composent qui ont exactement la même taille que ce masque ou ce filtre.
- Dans toutes les approches listées ensuite il est indipensable de stocker 2 éléments : l'image de départ et l'image de calcul qui deviendra l'image résultat.

Transformations locales

Filtres linéaires : convolution en pratique

Somme des poids = 1 pour conserver la dynamique entre 0 et 255

250	151	198
254	152	117
195	251	223

Image

Transformations locales

Filtres linéaires : convolution en pratique

Somme des poids = 1 pour conserver la dynamique entre 0 et 255

250	151	198
254	152	117
195	251	223

Image

0	1	0
1	-4	1
0	1	0

Masque de convolution

Transformations locales

Filtres linéaires : convolution en pratique

Somme des poids = 1 pour conserver la dynamique entre 0 et 255

250	151	198
254	152	117
195	251	223

Image

0	1	0
1	-4	1
0	1	0

Masque de convolution

$$250 \times 0 + 151 \times 1 + 198 \times 0 \ldots = 165$$

Transformations locales

Filtres linéaires : convolution en pratique

Somme des poids = 1 pour conserver la dynamique entre 0 et 255

Image

250	151	198
254	152	117
195	251	223

Masque de convolution

0	1	0
1	-4	1
0	1	0

Résultat →

Image transformée

?	?	?
?	165	?
?	?	?

$$250 \times 0 + 151 \times 1 + 198 \times 0 \ldots = 165$$

On calcule donc : $151 + 254 + 117 + 251 - 1524 \times 4 = 165$.

Transformations locales

Filtres linéaires : convolution, voisinage et bordure

- **Notion de voisinage** : critère de similarité (niveaux de gris, couleur) ou critère d'adjacence
- **Stratégie aux bords de l'image** : quel impact sur les résultats ?
  - option `valid` en `matlab` : bords non traités
    Image résultat sera plus petite
  - option `same` en `matlab` : Ajout d'une bordure nulle ← revient à ajouter une bordure noire
    Image résultat de même taille
  - Calculs effectués de manière circulaire
    Image résultat de même taille
  - option `full` en `matlab` : Ajout d'une bordure nulle de la taille du filtre
    Image résultat plus grande

Transformations locales

Filtres linéaires : convolution, filtre moyenneur

$$F = \frac{1}{N_l \times N_c} \mathbf{1}_{N_l \times N_c}$$

Transformations locales

Filtres linéaires : convolution, filtre gaussien

1D

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

2D

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

avec μ, μ_x, μ_y : moyennes et σ : écart type

Transformations locales

Filtres linéaires : convolution, filtre gaussien

1D

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

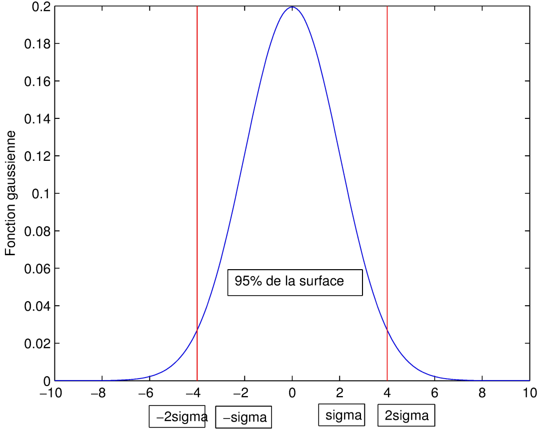
2D

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

avec μ, μ_x, μ_y : moyennes et σ : écart type
Souvent $\mu, \mu_x, \mu_y = 0$ et σ dépend de la taille du masque et $\sigma = \frac{N_m}{4}$ où
N_m : taille du masque

Transformations locales

Filtres linéaires : convolution, filtre gaussien



Transformations locales

Filtres linéaires : convolution, exemple de filtres gaussiens (1)

- 1D avec une taille de 5 et en 2D avec une taille de 3

$$F_5 = \begin{pmatrix} f(-2) & f(-1) & f(0) & f(1) & f(2) \end{pmatrix}$$

$$F_{3,3} = \begin{pmatrix} f(-1,-1) & f(-1,0) & f(-1,1) \\ f(0,-1) & f(0,0) & f(0,1) \\ f(1,-1) & f(1,0) & f(1,1) \end{pmatrix}$$

Transformations locales

Filtres linéaires : convolution, exemple de filtres gaussiens (2)

- Filtres simplifiés

$$F_5 = \begin{pmatrix} f(2) & f(1) & f(0) & f(1) & f(2) \end{pmatrix}$$

$$F_{3,3} = \begin{pmatrix} f(1,1) & f(0,1) & f(1,1) \\ f(0,1) & f(0,0) & f(0,1) \\ f(1,1) & f(0,1) & f(1,1) \end{pmatrix}$$

- Filtres normalisés

$$F_5^N = \frac{1}{f(0) + 2f(1) + 2(f2)} F_5 \text{ et } F_{3,3}^N = \frac{1}{f(0,0) + 4f(0,1) + 4f(1,1)} F_{3,3}$$

Transformations locales

Filtres linéaires : convolution, moyenneur : comportement

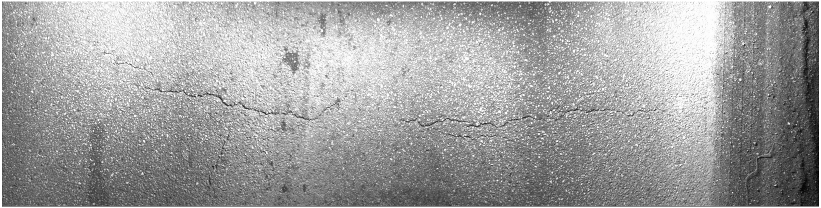
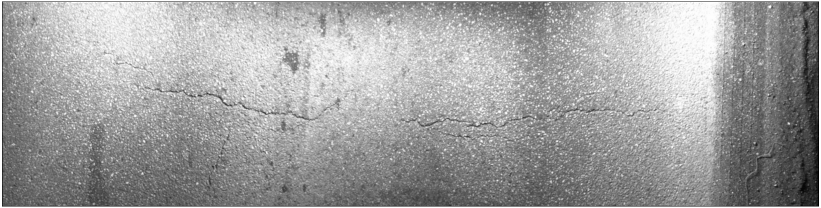


Image originale



Filtre 3×3

Transformations locales

Filtres linéaires : convolution, moyenneur : comportement

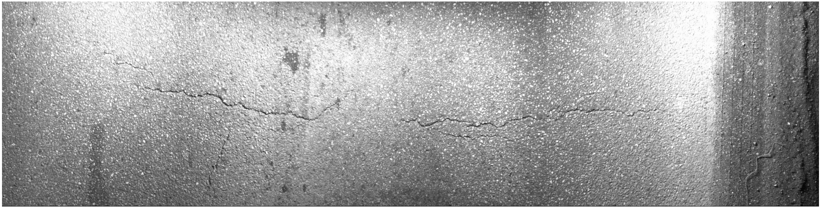
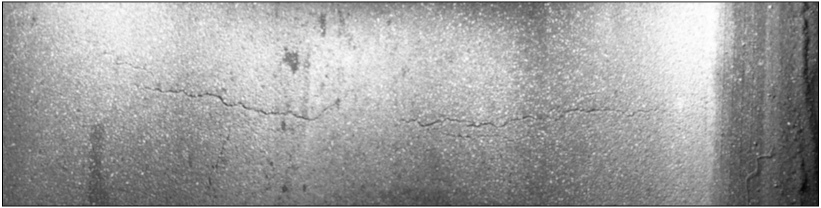


Image originale



Filtre 5×5

Transformations locales

Filtres linéaires : convolution, filtre gaussien : comportement et comparaison

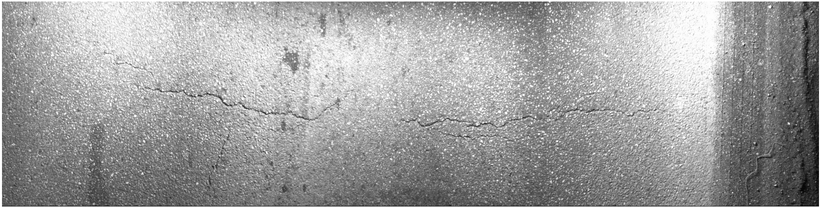
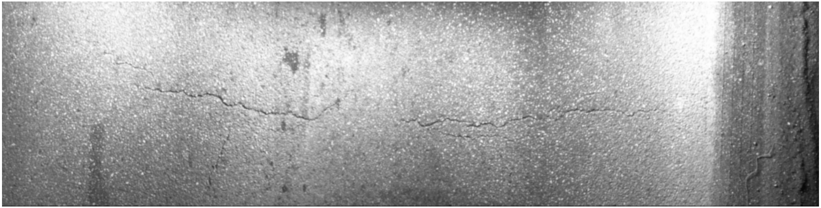


Image originale



Filtre gaussien 3×3

Transformations locales

Filtres linéaires : convolution, filtre gaussien : comportement et comparaison

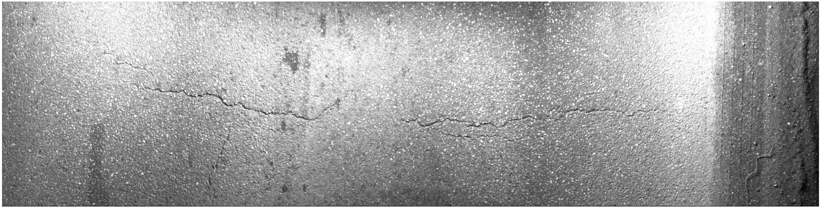
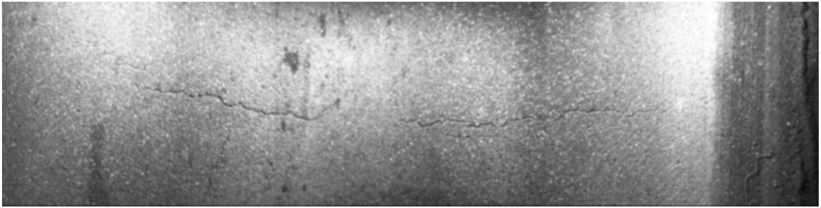


Image originale



Filtre gaussien 9×9

Transformations locales

Filtres linéaires : convolution, filtre gaussien : comportement et comparaison

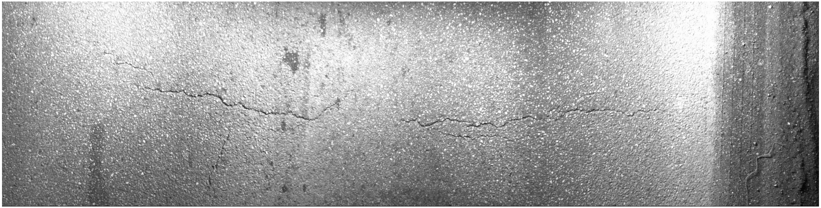
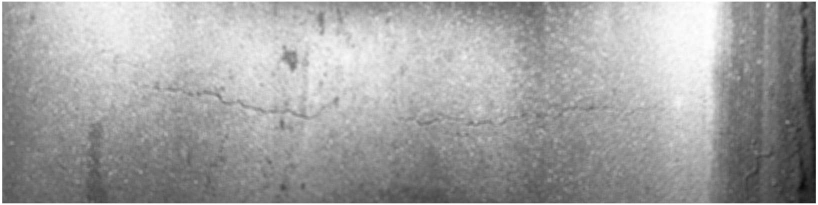


Image originale



Filtre moyenneur 9×9

Transformations locales

Filtres non linéaires : filtre médian et conservateur

- Les filtres linéaires présentés ne permettent pas d'éliminer facilement des bruits de type "poivre et sel".

Transformations locales

Filtres non linéaires : filtre médian et conservateur

- Les filtres linéaires présentés ne permettent pas d'éliminer facilement des bruits de type "poivre et sel".
- **Médian**

$$I^{'}(x, y) = \text{med}\{\mathcal{V}(x, y)\}$$

Transformations locales

Filtres non linéaires : filtre médian et conservateur

- Les filtres linéaires présentés ne permettent pas d'éliminer facilement des bruits de type "poivre et sel".
- **Médian**

$$I^{'}(x,y) = \text{med}\{\mathcal{V}(x,y)\}$$

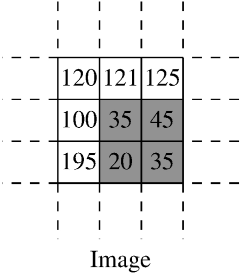
- **Conservateur** : Version "plus douce " du filtre médian

$$I^{'}(x,y) = \begin{cases} I_{\min}(x,y) & \text{Si } I(x,y) < I_{\min} \\ I_{\max}(x,y) & \text{Si } I(x,y) > I_{\max} \\ I(x,y) & \text{Sinon.} \end{cases}$$

où $I_{\min/\max}(x,y) = \min / \max(\mathcal{V}(x,y))$ et $I(x,y) \notin \mathcal{V}(x,y)$

Transformations locales

Filtres non linéaires : filtre médian et conservateur



Transformations locales

Filtres non linéaires : filtre médian et conservateur

120	121	125
100	35	45
195	20	35

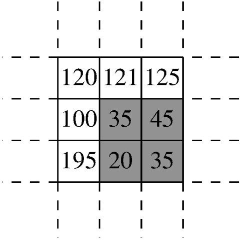
Image

120	121	125
100	100	45
195	20	35

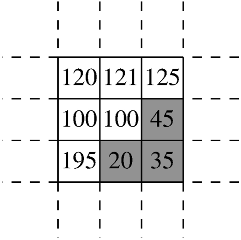
Filtre médian

Transformations locales

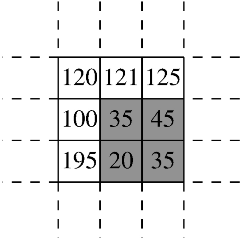
Filtres non linéaires : filtre médian et conservateur



Image



Filtre médian



Filtre conservateur

Transformations locales

Filtres non linéaires : comportement et comparaison

Image bruitée (10%)

Transformations locales

Filtres non linéaires : comportement et comparaison

Image bruitée (10%)

Filtre moyenneur 3×3

Transformations locales

Filtres non linéaires : comportement et comparaison

Image bruitée (10%)

Filtre moyenneur 5×5

Transformations locales

Filtres non linéaires : comportement et comparaison

Image bruitée (10%)

Filtre gaussien 3×3

Transformations locales

Filtres non linéaires : comportement et comparaison

Image bruitée (10%)

Filtre gaussien 5×5

Transformations locales

Filtres non linéaires : comportement et comparaison

Image bruitée (10%)

Filtre médian 3×3

Transformations locales

Filtres non linéaires : comportement et comparaison

Image bruitée (10%)

Filtre médian 5×5

Transformations locales

Filtres non linéaires : comparaison

Image bruitée (1%)

Transformations locales

Filtres non linéaires : comparaison

Image bruitée (1%)

Filtre médian 3×3

Transformations locales

Filtres non linéaires : comparaison

Image bruitée (1%)

Filtre conservateur 3×3

Transformations locales

Filtres linéaires et non linéaires : résumé

Filtre	**Moyenne**	**Gaussien**
Principe	Moyenne des niveaux de gris	Moyenne pondérée des niveaux de gris
Intérêts	Atténue bruits, texture (lissage)	Plus doux que la moyenne
Inconvénients	Flou, atténue petits détails	Moins importants que la moyenne
Filtre	**Médian**	**Conservateur**
Principe	Médiane des niveaux de gris	Médiane tronquée
Intérêts	Atténue bruit impulsionnel	Plus doux que la médiane
Inconvénients	Supprime les coins	Non robuste aux images très bruitées

Ponctuel versus local

- Quel est l'intérêt d'utiliser des transformations ponctuelles plutôt que locales ?
- Ne permettent-elles pas de faire la même chose ?

Ponctuelle	**Locale**
Amélioration surtout du **contraste**	Amélioration surtout en supprimant les **bruits**
Permer d'étudier la **dynamique de l'image et de la modifier**	Permet d'étudier des primitives de l'image, comme les **contours**, cf. chapitre suivant

Comment palier les défauts ?

- Exemple : coins enlevés par filtre médian $\rightarrow$ filtre conservateur
- Proposition : combiner des filtres
- **Ce qu'il faut retenir**

  Indications fournies pour faire le meilleur choix en fonction de l'application visée

Plan de la présentation

Définitions et notion de dérivée

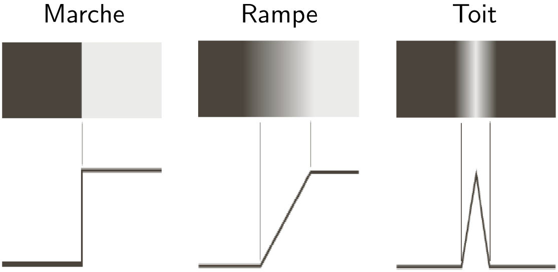
Définitions

Un contour = frontière séparant deux objets/entités
En **pratique** *= variation d'intensité*

Définitions et notion de dérivée

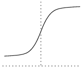
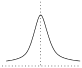
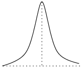
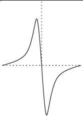
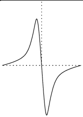
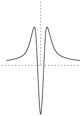
Notion de dérivées

Profil théorique : Pourquoi choisir marche ou toit alors qu'on observe rampe ?



Définitions et notion de dérivée

Notion de dérivée

Profil	 Marche	 Toit
Dérivée première	 Maximum	 Passage par zéro
Dérivée seconde	 Passage par zéro	 Minimum

Définitions et notion de dérivée

Notion de dérivée

- La plupart du temps : contours de type marche ou toit (mais surtout marche).
- Ils sont favorisés car les profils des dérivées sont plus faciles à analyser qu'avec une rampe.
- **Si nous choisissons un des deux modèles, nous détecterons moins bien les contours qui suit le deuxième modèle.**
- Tout le raisonnement proposé pour un contour de type marche est strictement généralisable au cas des contours de type toit.

Plan de la présentation

Algorithme de détection basique

Données

Image à traiter : I

Image binaire représentant les contours : I_c

Algorithme de calcul de contours

1. Pour chaque $(x, y) \in I$ faire

   Calcul du gradient en évaluant sa norme et son gradient

   - Norme : $\|\nabla I(x, y)\| = \sqrt{(I_x^2 + I_y^2)}$
   - Direction : $\theta = \arctan\left(\frac{I_y}{I_x}\right)$

   ou calcul du laplacien
2. Seuillage de l'image ou des images

   ou détection des passages par zéro
3. Fermeture des contours

Analyse de cet algorithme

- Variations induites par
  1. Manière d'effectuer le seuillage l'image
  2. Utilisation ou non d'une étape de fermeture
  3. Filtre utilisé pour le calcul de la dérivée

Analyse de cet algorithme

- Variations induites par
  1. Manière d'effectuer le seuillage l'image
  2. Utilisation ou non d'une étape de fermeture
  3. Filtre utilisé pour le calcul de la dérivée
- Dérivées premières : seuillage effectué avec l'image des normes

Seuillage/binarisation

- Choix du seuillage délicat : Otsu, distribution gaussienne
- **Seuillage par hystérésis**

Seuillage/binarisation

- Choix du seuillage délicat : Otsu, distribution gaussienne
- **Seuillage par hystérésis**
  - Deux niveaux de seuils : S_h, un seuil haut, et S_b un seuil bas : $S_b < S_h$

Seuillage/binarisation

- Choix du seuillage délicat : Otsu, distribution gaussienne
- **Seuillage par hystérésis**
  - Deux niveaux de seuils : S_h, un seuil haut, et S_b un seuil bas : $S_b < S_h$
  - Étapes
    1. Sélection de $I_h = \{I(x,y) | \|\nabla I(x,y)\| \geq S_h\}$ (points fiables)
    2. Sélection de $I_b = \{I(x,y) | \|\nabla I(x,y)\| \geq S_b \text{ et } \exists\ I(k,l) \ \in \mathcal{V}(x,y) \mid I(k,l) \in I_h\}$
    3. $I_h = I_h \bigcup I_b$

Seuillage/binarisation

- Choix du seuillage délicat : Otsu, distribution gaussienne
- **Seuillage par hystérésis**
  - Deux niveaux de seuils : S_h, un seuil haut, et S_b un seuil bas : $S_b < S_h$
  - Étapes
    1. Sélection de $I_h = \{I(x,y) | \|\nabla I(x,y)\| \geq S_h\}$ (points fiables)
    2. Sélection de $I_b = \{I(x,y) | \|\nabla I(x,y)\| \geq S_b \text{ et } \exists\ I(k,l)\ \in \mathcal{V}(x,y) \mid I(k,l) \in I_h\}$
    3. $I_h = I_h \bigcup I_b$
  - (2) et (3) sont répétés de manière itérative jusqu'à ce que $I_b = \emptyset$

Seuillage/binarisation

- Choix du seuillage délicat : Otsu, distribution gaussienne
- **Seuillage par hystérésis**
  - Deux niveaux de seuils : S_h, un seuil haut, et S_b un seuil bas : $S_b < S_h$
  - Étapes
    1. Sélection de $I_h = \{I(x,y) | \|\nabla I(x,y)\| \geq S_h\}$ (points fiables)
    2. Sélection de $I_b = \{I(x,y) | \|\nabla I(x,y)\| \geq S_b \text{ et } \exists\ I(k,l)\ \in \mathcal{V}(x,y) \mid I(k,l) \in I_h\}$
    3. $I_h = I_h \bigcup I_b$
  - (2) et (3) sont répétés de manière itérative jusqu'à ce que $I_b = \emptyset$
  - **Avantages :** contours mieux fermés et choix des seuils moins sensible

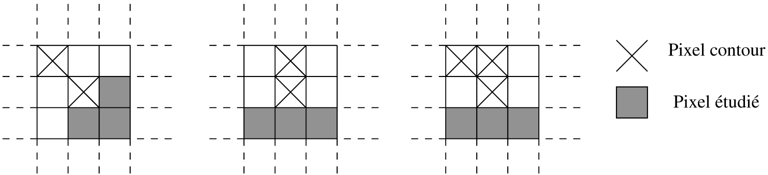
Seuillage/binarisation

- Choix du seuillage délicat : Otsu, distribution gaussienne
- **Seuillage par hystérésis**
  - Deux niveaux de seuils : S_h, un seuil haut, et S_b un seuil bas : $S_b < S_h$
  - Étapes
    1. Sélection de $I_h = \{I(x,y) | \|\nabla I(x,y)\| \geq S_h\}$ (points fiables)
    2. Sélection de $I_b = \{I(x,y) | \|\nabla I(x,y)\| \geq S_b \text{ et } \exists\ I(k,l)\ \in \mathcal{V}(x,y) \mid I(k,l) \in I_h\}$
    3. $I_h = I_h \bigcup I_b$
  - (2) et (3) sont répétés de manière itérative jusqu'à ce que $I_b = \emptyset$
  - **Avantages :** contours mieux fermés et choix des seuils moins sensible
  - **Contraintes à ajouter :** Lien entre les deux seuils

Fermeture de contours

- **Motivations**
  1. Manques dus à la présence de bruit ou à des occultations
  2. Erreurs dues à des leurres dans les images
- **Principes utilisés**
  1. **Configuration des points**
  2. Direction du gradient
  3. Suppression des contours non-fermés

Fermeture de contours par configuration de points



Représentation des contours par codage de Freeman

Un exemple ensemble

- **Principe :** Représentation compact d'un contour en numérotant les 8 directions possibles pour un point de contour, et en considérant qu'un élément de contour relie 2 pixels connexes
- **Étapes :**
  1. Choix du nombre de directions (4 ou 8)
  2. Choix d'un pixel initial
  3. Codage de la direction qui permet de passer au pixel contour suivant.

  L'étape (3) est réalisée jusqu'à revenir au point initial.

Plan de la présentation

Calcul de dérivées premières

Illustration du gradient

Approximation par différences finies

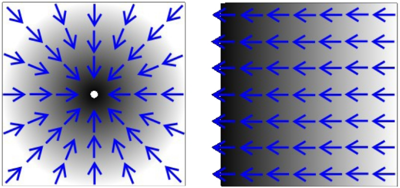
- Le principe est de calculer des différences finies pour approximer les dérivées, exactement comme cela peut être fait en mathématiques pour estimer les dérivées de fonctions.
- Plusieurs directions sont possibles, d'où deux filtres : un pour calculer la dérivée suivant les lignes et un suivant les colonnes.

Calcul de dérivées premières

Illustration du gradient

Approximation par différences finies

- Le principe est de calculer des différences finies pour approximer les dérivées, exactement comme cela peut être fait en mathématiques pour estimer les dérivées de fonctions.
- Plusieurs directions sont possibles, d'où deux filtres : un pour calculer la dérivée suivant les lignes et un suivant les colonnes.



Calcul de dérivées premières

Convolution sans lissage

Nom	F_x	F_y
Différences de pixels 1	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix}$
Différences de pixels 2	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$
Roberts	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Calcul de dérivées premières

Convolution avec pondération

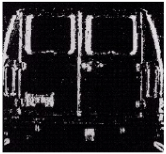
Prewitt	$\frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$	$\frac{1}{3}\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$
Sobel	$\frac{1}{4}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$	$\frac{1}{4}\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$
Frei-Chen	$\frac{1}{4}\begin{pmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{pmatrix}$	$\frac{1}{4}\begin{pmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{pmatrix}$

Calcul de dérivées premières

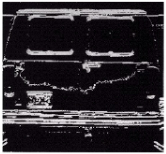
Exemples de détection : effet des filtres directionnels

Image

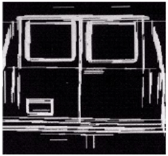
Verticaux



Horizontaux



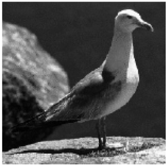
Norme



Calcul de dérivées premières

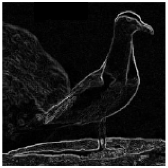
Exemples de détection : effet des différents filtres

Image

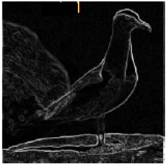


Roberts

Prewitt



Sobel



Calcul de dérivées premières

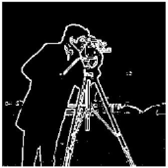
Exemples de détection : effet du seuillage

Image

Sans seuil

Seuil $= 25$

Seuil $= 60$



Calcul de dérivées premières

Analyse et conclusion sur ces opérateurs basiques

- Sensible aux bruits
- Nécessité d'un lissage
- Filtrage, par exemple, gaussien

Calcul de dérivées premières

Convolution avec lissage

- On souhaite calculer

$$I^{'} = (I \times f)^{'}$$

Calcul de dérivées premières

Convolution avec lissage

- On souhaite calculer
$$I^{'} = (I \times f)^{'}$$

- En utilisant la transformée de Fourier, sachant que f est la réponse impulsionnelle d'un filtre
$$(I \times f)^{'} = I \times f^{'}$$

- De même : $(I \times f)^{''} = I \times f^{''}$

Calcul de dérivées premières

Convolution avec lissage

- On souhaite calculer

$$I^{'} = (I \times f)^{'}$$

- En utilisant la transformée de Fourier, sachant que f est la réponse impulsionnelle d'un filtre

$$(I \times f)^{'} = I \times f^{'}$$

- De même : $(I \times f)^{''} = I \times f^{''}$

On peut calculer la dérivée tout en effectuant un lissage si on filtre avec la dérivée d'un filtre de lissage.

Calcul de dérivées premières

Exemples de masques obtenus avec la dérivée gaussienne

- **1D**

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

Calcul de dérivées premières

Exemples de masques obtenus avec la dérivée gaussienne

- **1D**

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

- **Dérivée en 1D**

$$f'(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}}$$

Calcul de dérivées premières

Exemples de masques obtenus avec la dérivée gaussienne

- **1D**

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

- **Dérivée en 1D**

$$f^{'}(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}}$$

- **2D**

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

Calcul de dérivées premières

Exemples de masques obtenus avec la dérivée gaussienne

- **1D**

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

- **Dérivée en 1D**

$$f'(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}}$$

- **2D**

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

- **Dérivée en 2D, en x, et en y**

$$f_{x/y}(x) = \frac{-x/y}{2\pi\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Calcul de dérivées premières

Exemples de masques obtenus avec la dérivée gaussienne

- Masque en 1D avec une taille de 5

$$F_5^{'} = \begin{pmatrix} f^{'}(-2) & f^{'}(-1) & f^{'}(0) & f^{'}(1) & f^{'}(2) \end{pmatrix}$$

- Masque en x en 2D avec une taille de 3

$$F_{3,3}^{'} = \begin{pmatrix} f^{'}(-1,-1) & f^{'}(-1,0) & f(-1,1) \\ f^{'}(0,-1) & f^{'}(0,0) & f^{'}(0,1) \\ f^{'}(1,-1) & f^{'}(1,0) & f^{'}(1,1) \end{pmatrix}$$

Calcul des dérivées

Exemples de masques obtenus avec la dérivée gaussienne

Simplification car $\boxed{f^{'}(x) = -f^{'}(x)}$

$$F^{'}_{5} = \begin{pmatrix} f^{'}(2) & f^{'}(1) & 0 & f^{'}(1) & f^{'}(2) \end{pmatrix}$$

$$F^{'}_{3,3} = \begin{pmatrix} f(-1,1) & f(-1,0) & f(-1,1) \\ 0 & 0 & 0 \\ -f(-1,1) & -f(-1,0) & -f(-1,1) \end{pmatrix}$$

Calcul des dérivées

Exemples de masques obtenus avec la dérivée gaussienne

Normalisation par la somme des éléments positifs

$$\boxed{F_5^N = \frac{1}{f'(1) + f'(2)} F_5 \text{ et } F_{3,3}^N = \frac{1}{f(-1,0) + 2f(-1,1)} F_{3,3}.}$$

Plan de la présentation

Tranformations d'images

Détection de contours
- Définitions et notion de dérivée
- Algorithme de détection basique
- Calcul de dérivées premières
- Calcul des dérivées secondes

Calcul des dérivées secondes

Par convolution

$$\nabla^2 I(x,y) = I_{xx}(x,y) + I_{yy}(x,y).$$

- Approximation du calcul du laplacien en intégrant un passage par zéro au niveau du pixel étudié
- Somme des poids $= 0$
- Coefficients opposés entre le centre et le reste des pixels du masque

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Le Laplacien est sensible au bruit d'où le laplacien de gaussien.

Calcul des dérivées secondes

Par laplacien de gaussien ou *Laplacian of Gaussian, LoG*

$$LoG(x,y) = g_{xx}(x,y) + g_{yy}(x,y),$$

où g_{xx}, respectivement g_{yy}, sont les dérivées secondes de la gaussienne, en x, respectivement en y. Après développement de cette formule, nous obtenons :

- Après développement

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

- Approximation du laplacien, plus robuste au bruit, moins coûteux en temps de calculs :

  Différences de gaussiennes ou DoG, Difference Of Gaussian