Analyse d'image

Cours d'introduction

Xavier Descombes (d'après Mathias Ortner)

ESINSA, Option TNS, 2004

Plan du cours

1. **Courte Introduction**
   Les enjeux du traitement de l'image.
2. **l'image numérique**
   Les notions de bases : résolution, intensité, stockage.
3. **Histogrammes et contraste**
   Dynamique de l'image, rehaussement de contraste.
4. **Détection des contours**
5. **Bruit et filtrage**
   Courte présentation de notions de traitement du signal.
6. **Morphologie Mathématique**
   Une classe d'algorithmes et de méthodes couramment utilisées.
7. **Segmentation des images**
   Quelques notions sur les approches régions.

Définition et historique

Une image :

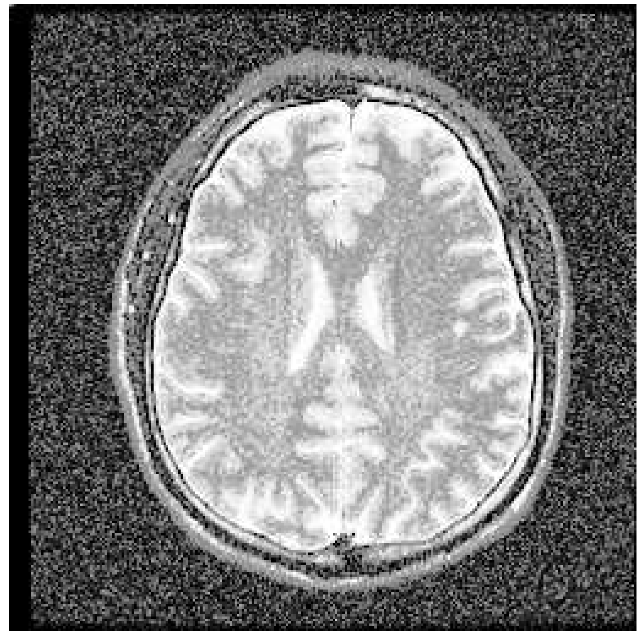
Dans le dictionnaire *Représentation de la forme ou de l'aspect d'un être ou d'une chose*

En informatique Une quantité d'information visuelle, sous une certaine représentation.

Différents capteurs Différents types d'information.

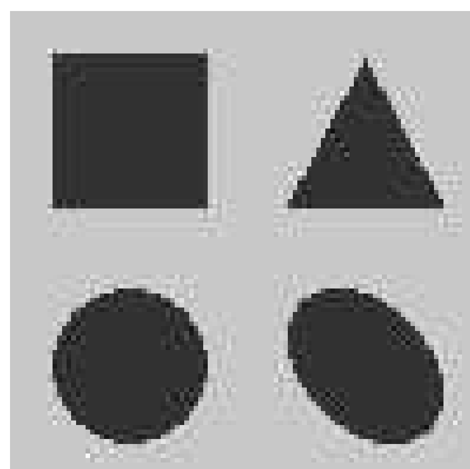
Exemples d'images

Lena



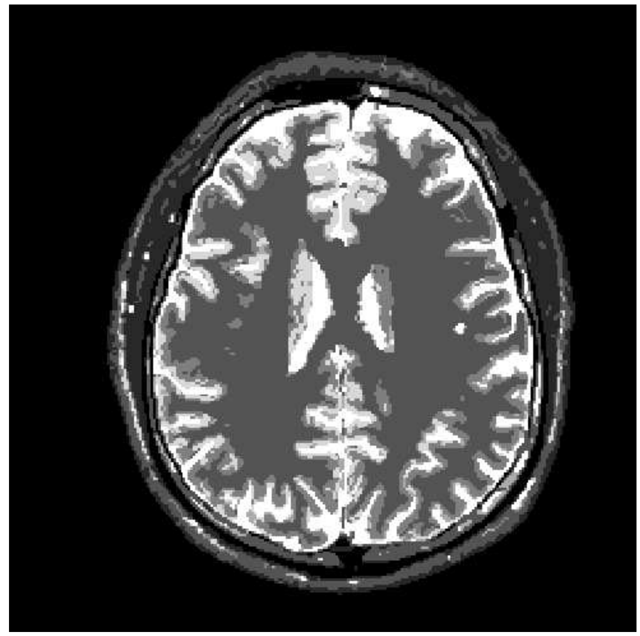
IRM

Amiens (IGN)

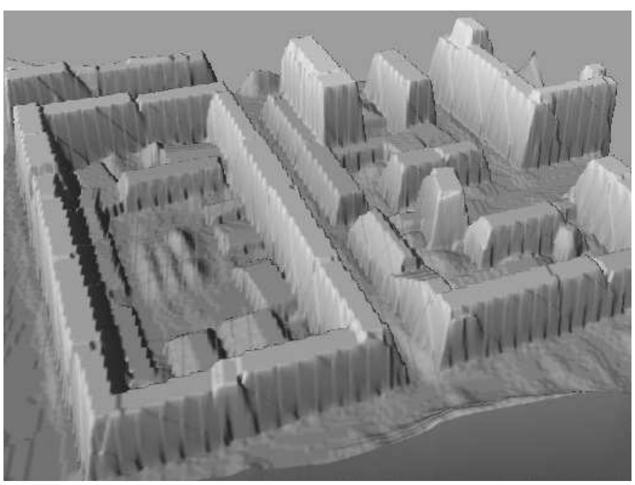


Exemples de traitements

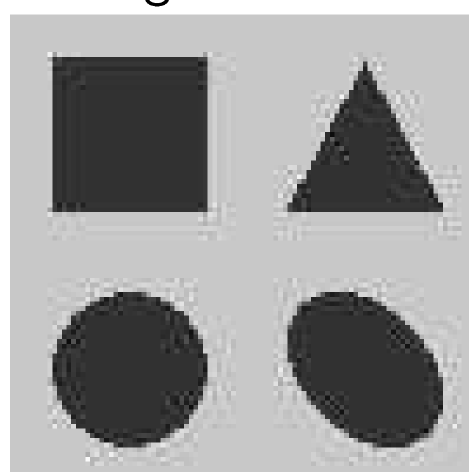
Det. Contours



Segmentation

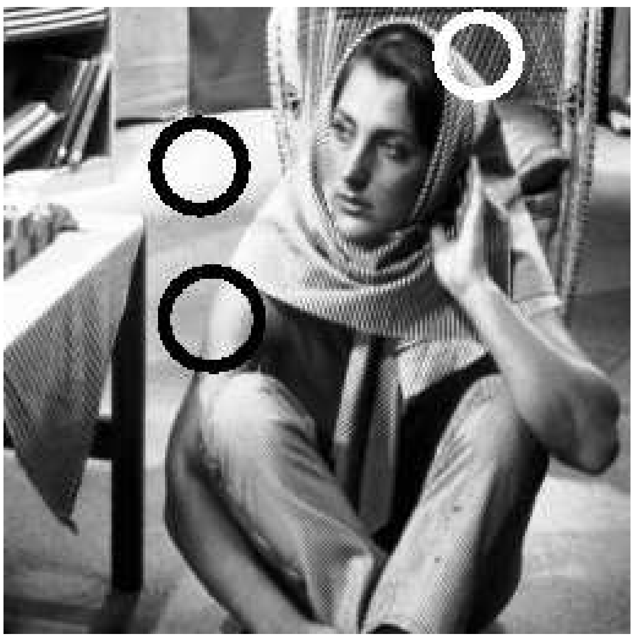


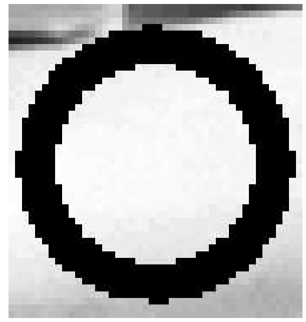
Stéréo-vision



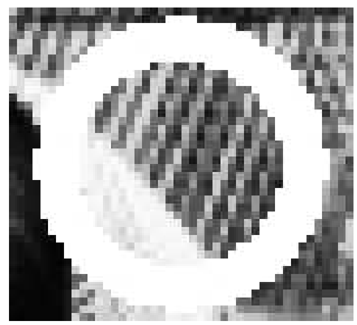
Det. Objets

Quelques éléments simples d'une image

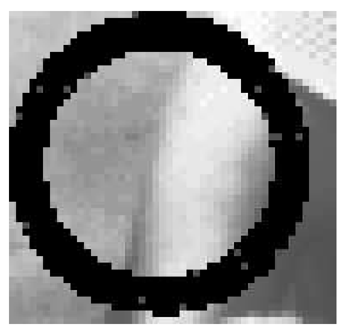




Une zone homogène



Une texture



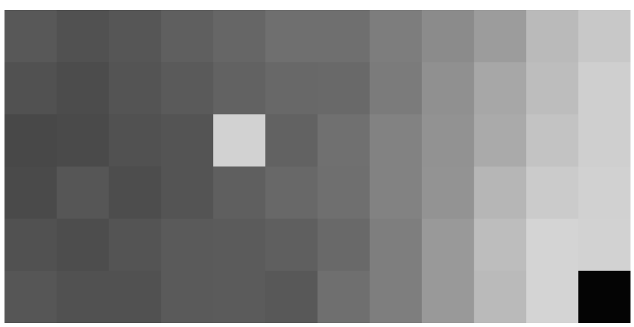
Un contour

Image Numérique

Une image numérique est un tableau de **pixels**

Notion de Pixel

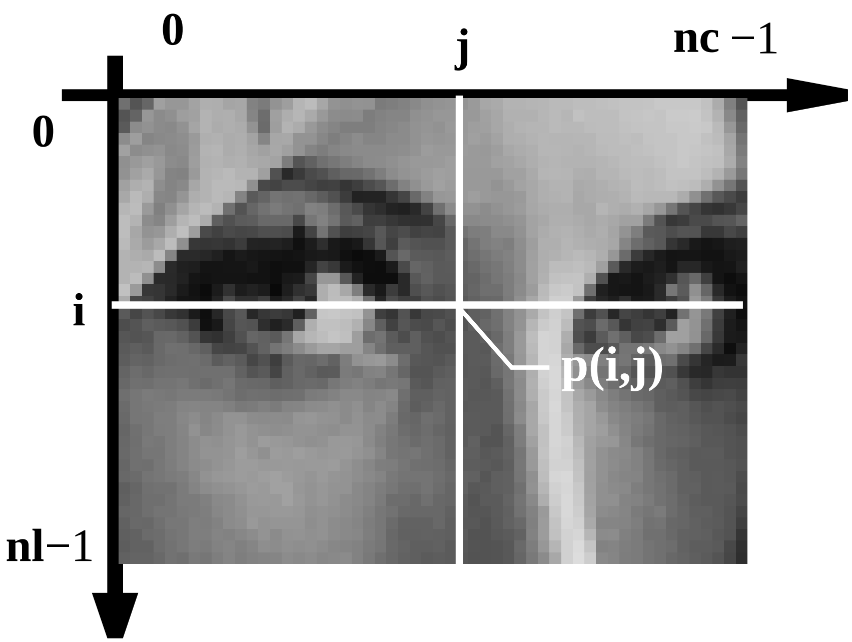
Définition : Un pixel est l'unité indivisible permettant de coder l'information relative à la luminosité en une certaine position.



Remarques :
– Les pixels sont en général carrés.
– Pixel vient de "picture element"
– Un ensemble de pixels donne une image.

L'image Numérique

Une image est un tableau de pixels : si le nombre de lignes vaut nl et le nombre de colonnes nc :



Un pixel est donc composé :
- de coordonnées (i, j) permettant de le situer,
- d'une valeur $v = p(i, j)$ représentant sa couleur .

Notion de résolution

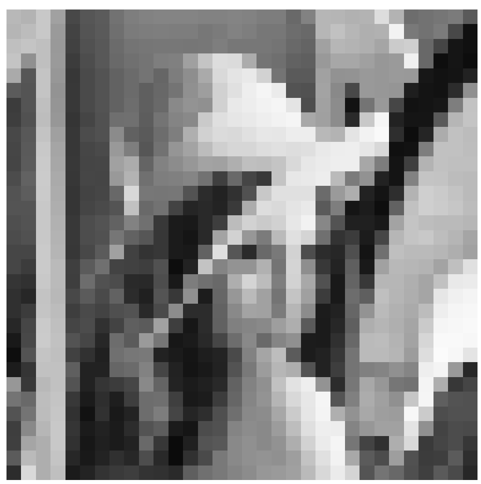
La résolution est donnée par la taille du pixel.

256x256

128x128

64x64

32x32



La résolution correspond à la finesse de la description spatiale de l'image (nombre de pixels).

Le codage des couleurs

Rappel : La valeur v d'un pixel $p =$ intensité lumineuse

En niveaux de gris

binaire : $v = 0$ (noir) ou $v = 1$ (blanc),

codage 8 bit : $v = 0, \ldots, 255$ (du plus foncé au plus clair)

codage 4 bit : $v = 0, \ldots, 15$.

En couleur

Le cas le plus courant : trois intensités lumineuses, vR, vG, vB (rouge,vert bleu).

codage 24 bit $vR = 0, \ldots, 255$, $vG = 0, \ldots, 255$ et $vB = 0, \ldots, 255$.

Codage des couleurs, illustration

Nb bits	8	1	4
Nb couleurs	255	2	16
			

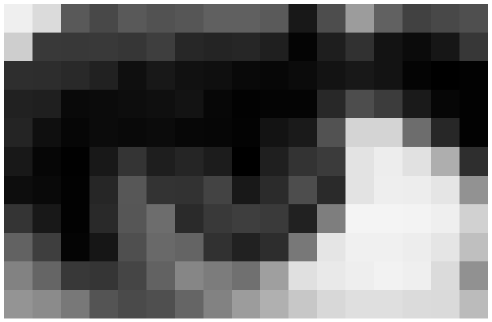
Conclusion : La qualité et la taille mémoire d'une image dépendent directement de la **résolution** et de la **quantification** (nb de bits par pixel).
Exemple 1 : SPOT3 : rés. 10 m, 1 oc/pix, scène 60 km x 60 km : 3,6 Mo
Exemple 2 : Pléiades : rés. 70 cm, 1 oc/pix, scène 10,5 km x 10,5 km : 225 Mo

Les formats de stokages

Les formats ont différentes propriétés. Les critères de comparaison élémentaires :

- Type de Format (matriciel ou vectoriel ?)
- Nature du codage (binaire ou ASCII ?)
- Possesseur du brevet (format libre ou non)
- Nombre de couleurs supportées
- Compression :
  - type de compression
  - avec ou sans perte

Exemple d'un format de codage : PGM



– C'est un format ASCII
– C'est un format matriciel.
– Deux parties : une entête et le corps de l'imag

```
P2
# CREATOR: XV Version 3.10a  Rev: 12/29/94
17 11
255
239 219  89  72  89  82  86  96  96  90  24  77 156  97  65  72  79
206  57  56  57  54  62  39  36  38  31   4  30  49  19  11  22  55
 46  45  41  33  15  24  17  15   9   7  11  19  24  19   4   0   2
 33  31   9  11  13  15  19   7   2   4   4  40  77  60  26   6   2
 36  15   6  11   9  11   7   6   3  19  26  82 212 212 109  39   2
 24   6   1  24  52  30  37  28   0  31  51  59 227 237 225 174  46 ...
```

Principaux formats

On ne détaille que quelques formats usuels, tous matriciels.

Format	Nature	Brevet ?	Nb couleurs	Comp	Usage
PGM	ASCII	Non	Variable	NON !	Format très simpl d'utilisation
PGMraw	binaire	Non	256	Non	idem
BMP	Binaire	-	-	Non	simple d'utilisation
Gif	Binaire	Unisys	256	sans perte	Logos, image avec peu de cou leurs et des zone homogénes
JPEG2000	Binaire	Non	16 M	avec ou sans perte	Images naturelles

L'image comme une fonction

On peut voir l'image comme une fonction u et donc une surface.

$$\begin{array}{rccc} u: & I \times J & \to & V \\ & (i,j) & \to & p(i,j) \end{array}$$

"A un point on associe une valeur d'intensité"

En discret

$$\begin{array}{ll} I = \{0, \ldots, nl-1\} & \text{et, par exemple :} \\ J = \{0, \ldots, nc-1\} & V = \{0, \ldots, 255\} \end{array}$$

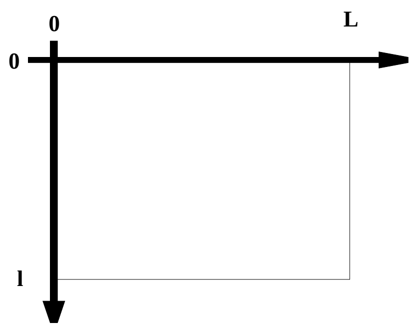
En continu

$$\begin{array}{ll} I = [0, l] & \text{et :} \\ J = [0, L] & V = [0, 1] \end{array}$$

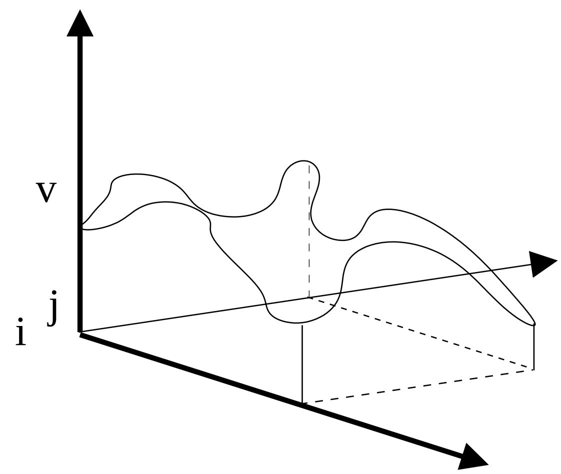
L'avantage d'une telle représentation continue vient de la possibilité de dériver...

L'image comme une surface

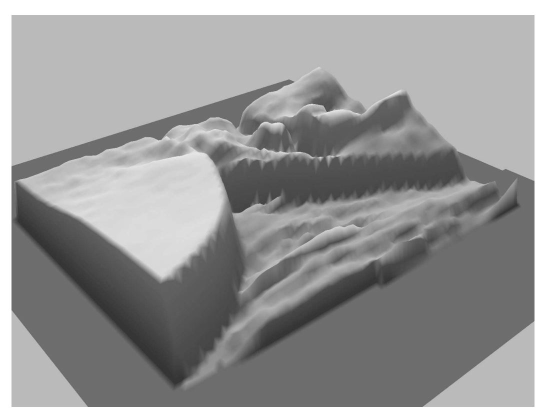
En utilisant la représentation précédente, on peut voir l'image comme une surface.



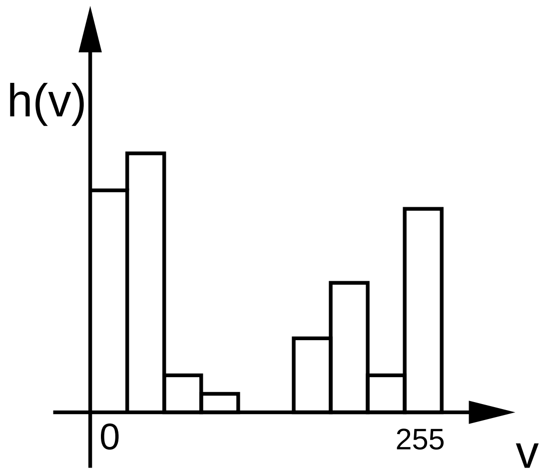
devient



donne

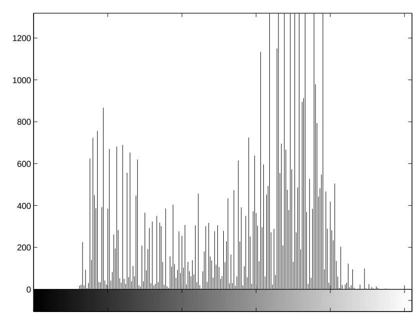


Histogramme d'une image



On se donne une image 8 bits u. On peut calculer la présence d'un niveau de gris v dans l'image :

$$\forall v \in \{0, \ldots, 255\} \qquad \boxed{h_u(v) = \text{Nb de pixels d'intensité } v}$$



Le tracé de la fonction h donne donc l'histogramme de l'image.

Densité de probabilité

On peut aussi calculer le **taux de présence** de chaque niveau de gris :

$$\boxed{P_u(v) = \frac{h_u(v)}{nl.nc}}$$

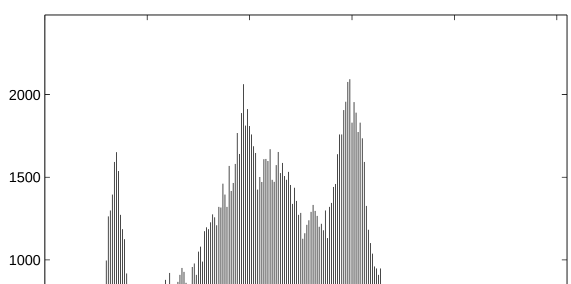
On peut vérifier que :

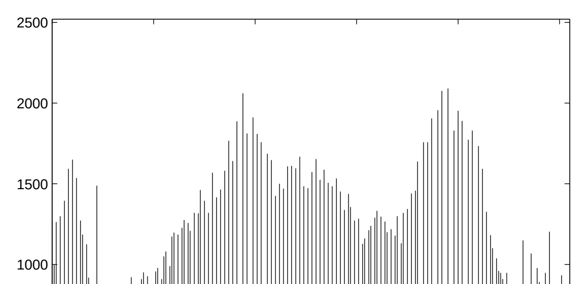
$$\sum_{v=0}^{v=255} P_u(v) = 1$$

donc $P_u(.)$ est une densité de probabilité : $P_u(v)$ est la probabilité de trouver un niveau de gris v en tirant un pixel au hasard dans l'image u.

Utilité de l'histogramme

Comparons les deux images suivantes et leurs histogrammes :





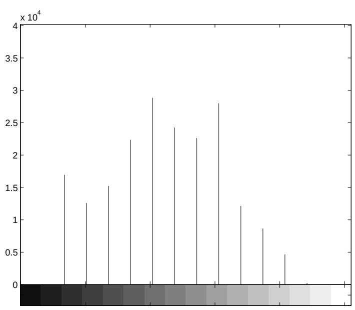
Conclusion : en jouant sur l'histogramme, on peut jouer sur le contraste.

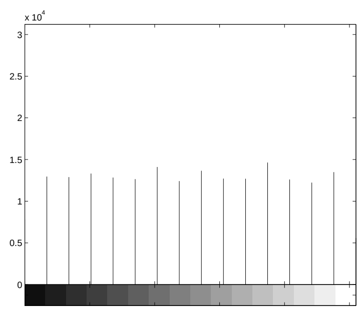
Histogrammes à n niveaux

On peut construire des histogrammes en utilisant moins d'intervalles. Par exemple si on considère 16 niveaux :

$$V_0 = \{0, \ldots, 15\} \qquad V_i = \{16 * i, \ldots, 16(i+1) - 1\}$$

$$V_{15} = \{240, \ldots, 255\}$$



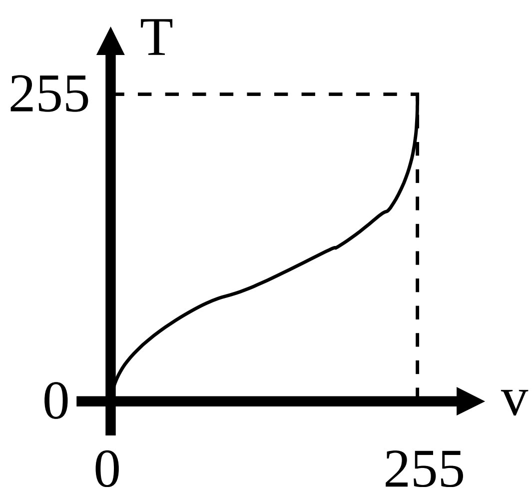
On constate que pour rehausser le contraste on a opéré une **égalisation d'histogramme à 16 niveaux**.

Transformation d'histogrammes

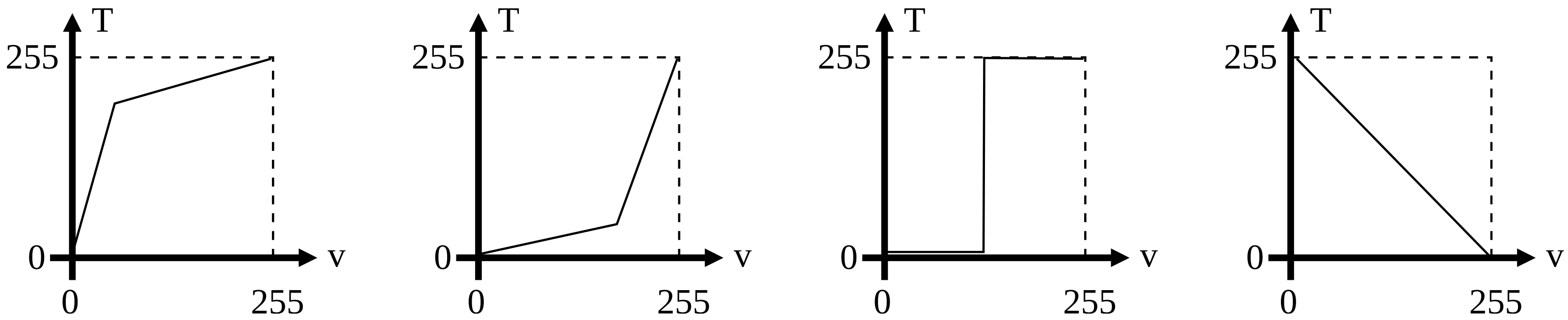
On imagine l'algorithme suivant de transformation des intensités :

- Parcourir les pixels de l'image
  - Pour le pixel courant p :
    1. lire la valeur de l'intensité v
    2. remplacer v par $T(v)$.

La fonction $T(.)$ remplace un niveau de gris par un autre. Cette fonction est appelée **transformation d'histogramme**. On la représente par son tracé :

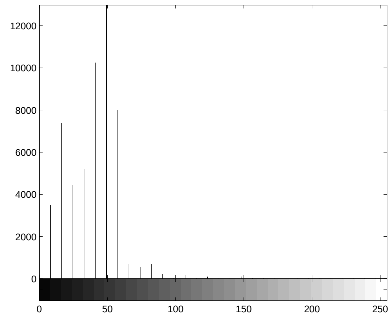


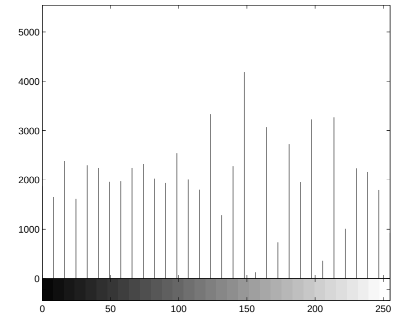
Exemples



Dynamique de l'image

On appelle **dynamique** d'une image la qualité de l'utilisation des niveaux de gris possibles.





Le rehaussement de contraste revient souvent à une amélioration de la dynamique de l'image. Il s'agit de trouver une transformation d'histogramme qui permette de passer de l'image de gauche à l'image de droite.

Égalisation d'histogrammes

Soit une image u. On rappelle que : $P_u(v) = \frac{h_u(v)}{nl.nc}$ (histogramme normalisé.)

On considère la **fonction de cumul :**

$$\forall k \in \{1, \ldots, 255\} \quad \phi_u(k) = \sum_{v=0}^{v=k} P_u(v)$$

$\phi_u(k)$ représente le taux de pixels dont le niveau de gris v est inférieur à k dans l'image u.

Si on applique la transformation d'histogramme $T(v) = 255 * \phi_u(v)$ à u on trouve **l'image normalisée** u' :

$$\forall v \in \{0, \ldots, 255\} \quad P_{u'}(v) \simeq 1/256$$

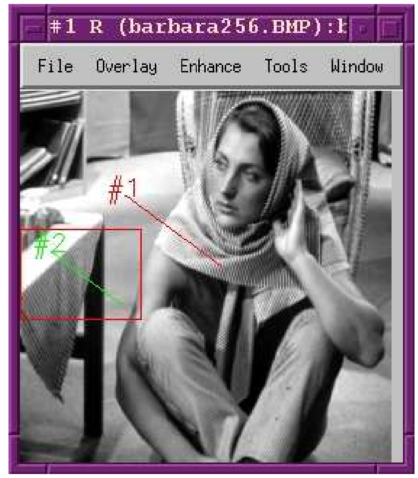
C'est à dire que dans l'image obtenue le nombre d'occurrences de chaque niveau de gris est à peu près égal aux autres.

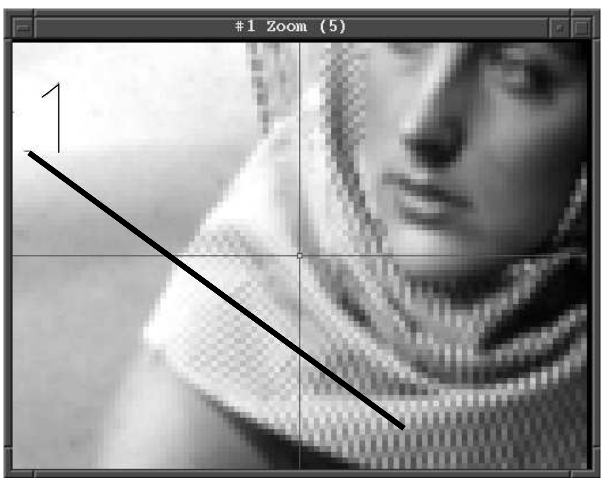
Introduction

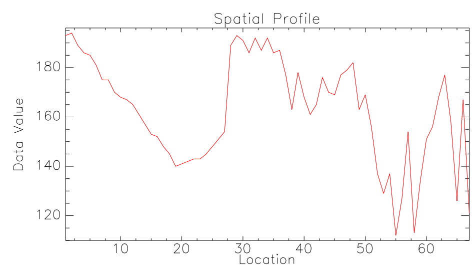
Dans ce chapitre, l'image est considérée comme une fonction :

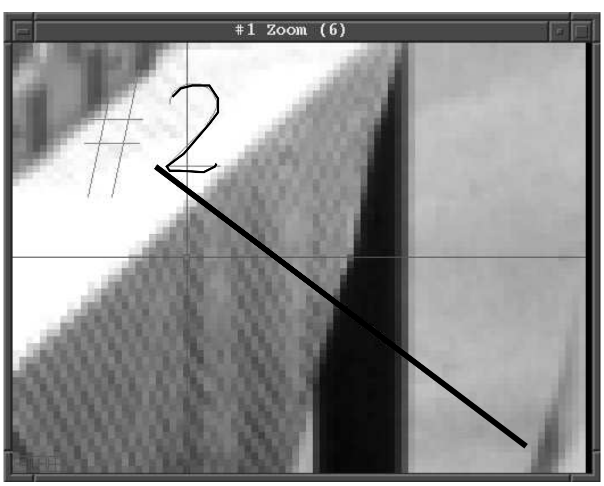
$$\begin{array}{cccc} u: & I \times J & \rightarrow & V \\ & (x,y) & \rightarrow & u(x,y) \end{array}$$

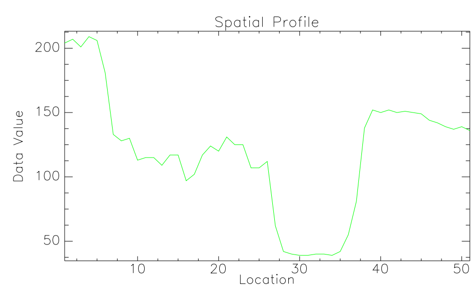
A un point de l'image, cette fonction associe une valeur.





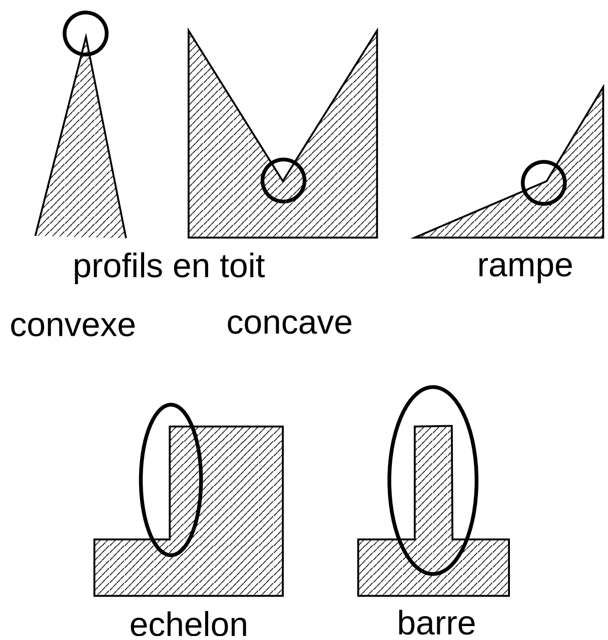






Discontinuités

Les différents types de discontinuités intéressantes dans une image :



Ces différentes discontinuités correspondent à des transitions entre deux zones différentes ou à la coupure d'une zone par une ligne.

Pour détecter ces discontinuités, il va falloir utiliser la dérivée d'un profil.

Les discontinuités sont présentes quand la dérivée change rapidement autour du point d'intérêt.

Différents types de contours

- **l'échelon** correspond au contour idéal et à une discontinuité franche.

- **la rampe** correspond au dégradé.

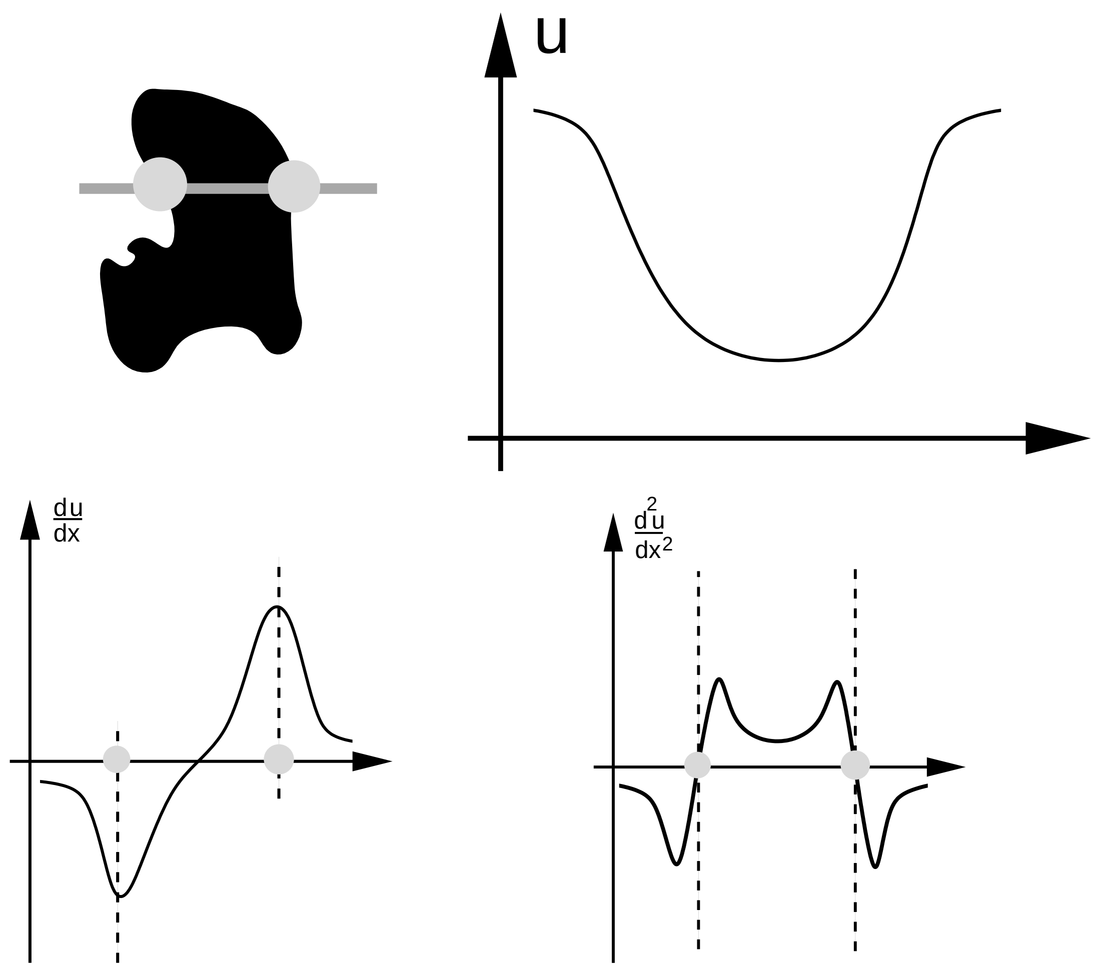
- **les profils en toit** correspondent à des lignes blanches (convexe) ou noires (concave).

- **la barre** correspond à une fine bande séparant deux zones.

Détection d'une discontinuité

Repose sur l'utilisation de la notion de dérivée : il y a deux possibilités :

– Recherche des maxima de la dérivée
– Recherche des passages par 0 de la dérivée seconde.



Gradient

Le gradient en un point est un vecteur (de $\mathbb{R}^2$) dont les composantes (δ_x, δ_y) mesurent à quelle vitesse la valeur des pixels évolue dans chacune des directions x et y, autour du point d'intérêt.

$$\begin{aligned} \delta_x &= \frac{\partial u}{\partial x} \simeq \frac{u(x+d_x, y) - u(x,y)}{d_x} \\ \delta_y &= \frac{\partial u}{\partial y} \simeq \frac{u(x, y+d_y) - u(x,y)}{d_y} \end{aligned}$$

avec d_x et d_y qui mesurent des distances dans les directions x et y respectivement.

Le gradient permet d'obtenir la dérivée dans n'importe quelle direction.

Dans le cas discret

d_x et d_y mesurent des distances en pixels. On peut prendre $d_x = d_y = 1$.

$$\begin{aligned}\delta_x &= u(i+1,j) - u(i,j)\\ \delta_y &= u(i,j+1) - u(i,j)\end{aligned}$$

Discontinuités

Pour détecter la présence ou non d'une discontinuité de gradient, on peut calculer :

- **La norme du gradient** $G = \sqrt{\delta_x^2 + \delta_y^2}$
- **Sa direction** $\theta = \arctan(\delta_y/\delta_x)$. La direction du gradient est la direction dans laquelle la dérivée est la plus grande.

Commentaires

– La détection des contours dépend du type d'image considéré. **Il n'y a pas de détecteur de contours universel.**

– En fonction de l'objectif, on optera pour telle ou telle méthode de détection des contours.

– Lors de la conception d'un détecteur de contours, il convient de toujours percevoir l'image comme un paysage dont la topographie constitue l'information importante.

Masques de convolutions

A droite, un masque m , a gauche un morceau de l'image u :

a	b	c
d	e	f
g	h	l

	(i-1,j-1)	(i-1,j)	(i-1,j+1)	
	(i,j-1)	**(i,j)**	(i,j+1)	
	(i+1,j-1)	(i+1,j)	(i+1,j+1)	

Convoluer l'image u par le masque m veut dire que l'on calcule l'image u' en appliquant la procédure suivante :

Pour tout pixel de l'image :

$$\begin{aligned} u'(i,j) &= l.u(i-1,j-1) + h.u(i-1,j) + g.u(i-1,j+1) \\ &+ f.u(i,j-1) + e.u(i,j) + d.u(i,j+1) \\ &+ c.u(i+1,j-1) + b.u(i+1,j) + a.u(i+1,j+1) \end{aligned}$$

Masques de gradients

Par exemple pour calculer les gradients on peut prendre les masques suivants (connus sous le nom d'opérateurs de Roberts)

δ_x

0	0	0
0	1	−1
0	0	0

δ_y

0	0	0
0	1	0
0	−1	0

Il existe une variante :

δ_1

0	0	0
0	1	0
0	0	−1

δ_2

0	0	0
0	0	1
0	−1	0

Ces opérateurs calculent les gradients dans les directions +45 degrés et -45 degrés.

Cependant : ces opérateurs sont très simples. Il existe d'autres opérateurs qui utilisent totalement le masque 3*3.

Les opérateurs de Sobel

Les avantages provenant de l'utilisation d'un masque 3*3 :

- Les erreurs sont moins grandes parce que les comportements aberrants sont moyennés sur une plus grande fenêtre,
- l'opérateur est centré, il n'y a pas de problème de parité.

δ_x

−1	0	1
−2	0	2
−1	0	1

δ_y

1	2	1
0	0	0
−1	−2	−1

Maxima du gradient

Nous savons calculer :
- les composantes du gradients
- la norme du gradient
- la direction du gradient

en un point de l'image.

Nous devons trouver les points tels que **le gradient soit maximal localement** : pour cela, deux méthodes :
- seuillage du gradient
- recherche du maximum local

Seuillage du gradient

Une fois la norme du gradient calculée en tous les points de l'image, on selectionne les pixels tels que :

$$G = \sqrt{\delta_x^2 + \delta_y^2} > G_{\min}$$

Par exemple :

Gradient

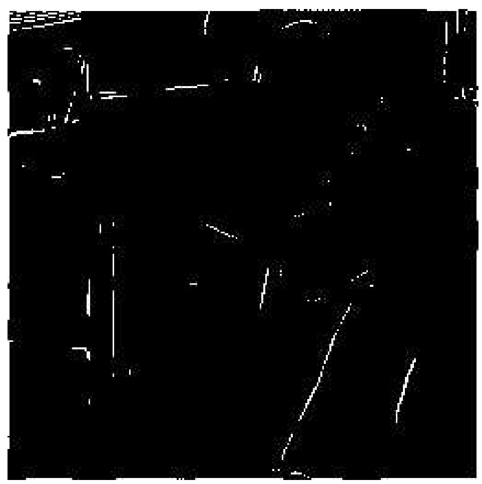
Gradient seuillé ($|G| > G_{min}$)

Cette méthode repose sur l'hypothèse que les maxima du gradients correspondent a peu prés au gradients plus grands que le seuil.

Problème du seuillage

Le problème des méthodes par seuillage vient du seuil a fixer. Par exemple :

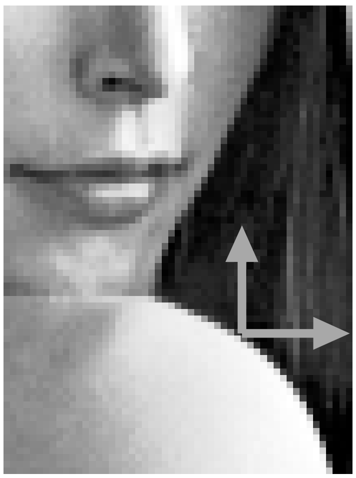
Seuil faible



Seuil grand

Recherche d'un maximum local

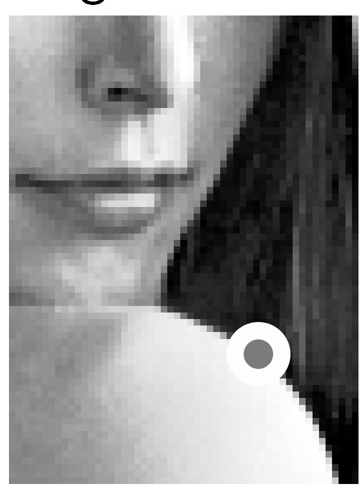
Pour éviter ce problème, il existe une autre méthode qui repose sur la détection d'un maximum local dans la direction du gradient.



Calcul de gradient δ_x, δ_y

Recherche de la direction du gradient : $\arctan(\delta_y/\delta_x)$

Dans la direction : recherche du maximum du gradient



Si le maximum coincide avec le point considéré : on a une discontinuité.

Contours et Laplacien

Une autre méthode de détection des discontinuités repose sur l'utilisation du Laplacien :

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

On cherche ensuite les endroits ou le Laplacien passe par 0.

Calcul du Laplacien

On peut utiliser l'un des deux masques suivants :

$$M_1 = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \qquad M_2 = \frac{1}{3}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

M_1 est obtenu par composition des masques de derivation, M_2 est une estimation directe d'un masque de Laplacien.

Détection du passage par 0

On commence par calculer la valeur du Laplacien sur toute l'image. On applique ensuite la règle suivante :

	Δ_1	
Δ_2	Δ	Δ_3
	Δ_4	

Si $\Delta < 0$ et l'un des autres $\Delta_i \geq 0$ **ou** $\Delta > 0$ et l'un des autres $\Delta_i \leq 0$ on considère qu'il y a un changement de signe.
Souvent on ne considère que les points où le Laplacien Δ est suffisamment grand.

Problème du bruit

Si l'image est trop bruitée, la détection par le Laplacien n'est pas très bonne. Il faut appliquer un filtre moyen auparavant (cf chapitre sur le filtrage)

Gradients et contours

Nous avons vu comment détecter la présence ou non d'une discontinuité en un pixel. Si l'on veut numéroter les pixels :
On peut utiliser :

- **le détecteur** (par exemple seuil sur la norme du gradient, maximum local ou passage par 0 du Laplacien)
- et sa **direction** si on cherche à **numéroter les contours** . Deux pixels voisins où l'on a détecté la présence de forts gradients appartiennent au même contour si les gradients ont même direction.

Par exemple :


Gradient

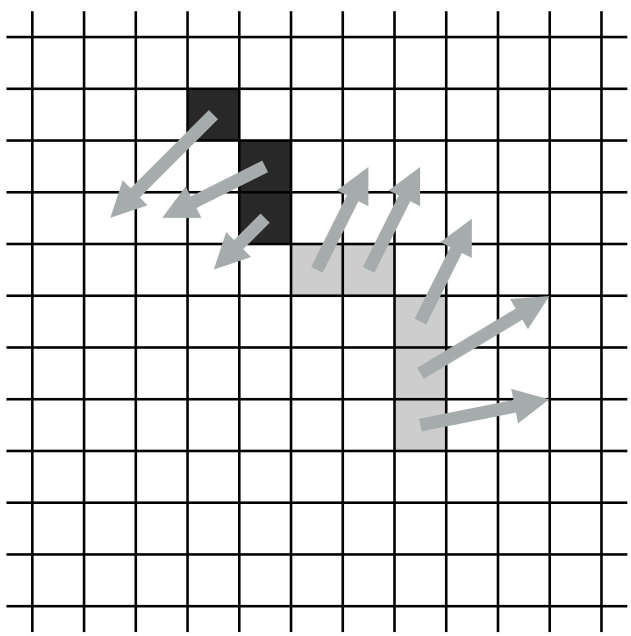

Gradient seuillé ($|G| > G_{min}$)

Un algorithme de numérotation des contours (1)

Test de présence : en un pixel $G > G_{min}$

Contrainte de similarité entre deux pixels p et p' : Les gradients sont orientés de la même manière :

$$|\theta(p) - \theta(p')| \leq \Delta\theta$$



Par exemple sur l'image : 2 contours

Un algorithme de numérotation des contours (2)

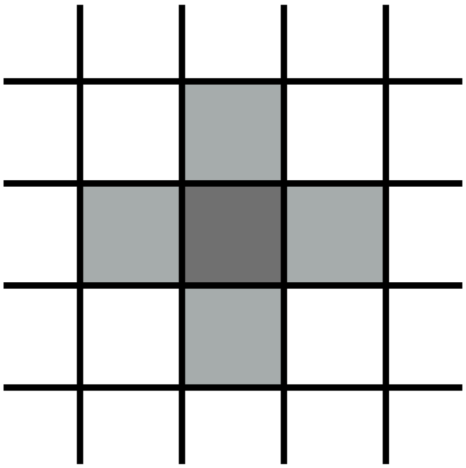
1. $C = 0$
2. Choisir un pixel p n'ayant pas encore été attribué à un contour tel que $(G(p) > G_{min})$
3. Attribuer au pixel p l'étiquette C.
4. Regarder les pixels voisins de p. Si un voisin p' de p vérifie :
   – le gradient $G(p')$ est plus grand que G_{min},
   – ainsi que la contrainte de similarité entre p et p'

   alors p' appartient au même contour :
   (a) on lui attribue l'étiquette C
   (b) on explore ses voisins (*voisinage + diffusion*).
5. Lorsque tout le voisinage pertinent a été exploré : $C = C + 1$, retour en 2 sauf si tous les pixels tels que $G > G_{min}$ ont été attribués à un contour.

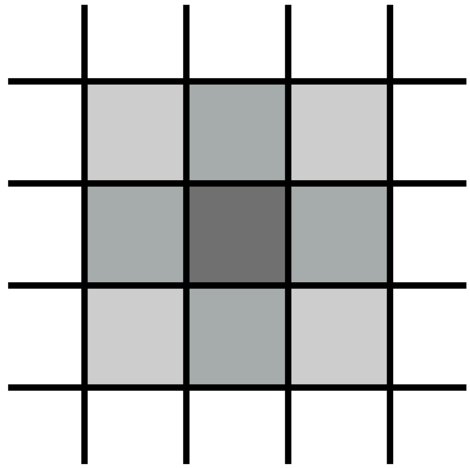
Notion de voisinage d'un pixel

Il y a deux types de voisinage simple :

4-connexe



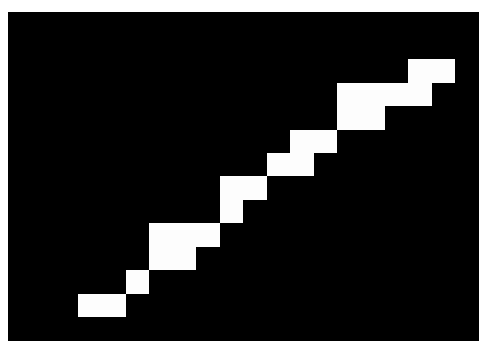
8-connexe



Deux exemples : un seul contour en 8-connexité mais en 4-connexité :

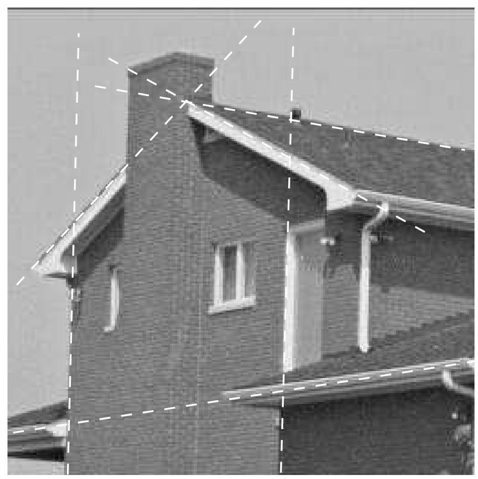
2 contours

6 contours



Détection de droites

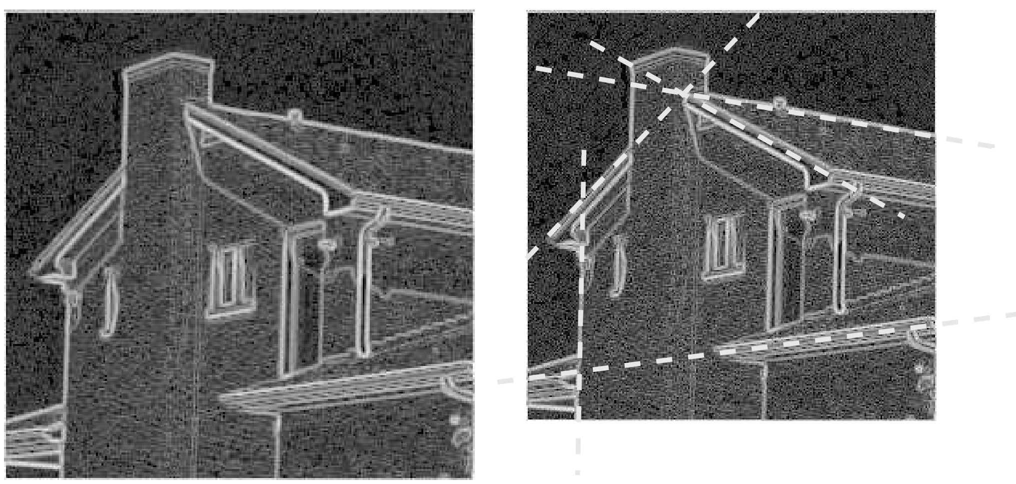
Le but est de trouver des droites dans une image. Par exemple :

Cette partie s'inscrit donc dans un contexte de détection d'objet.

Méthode :

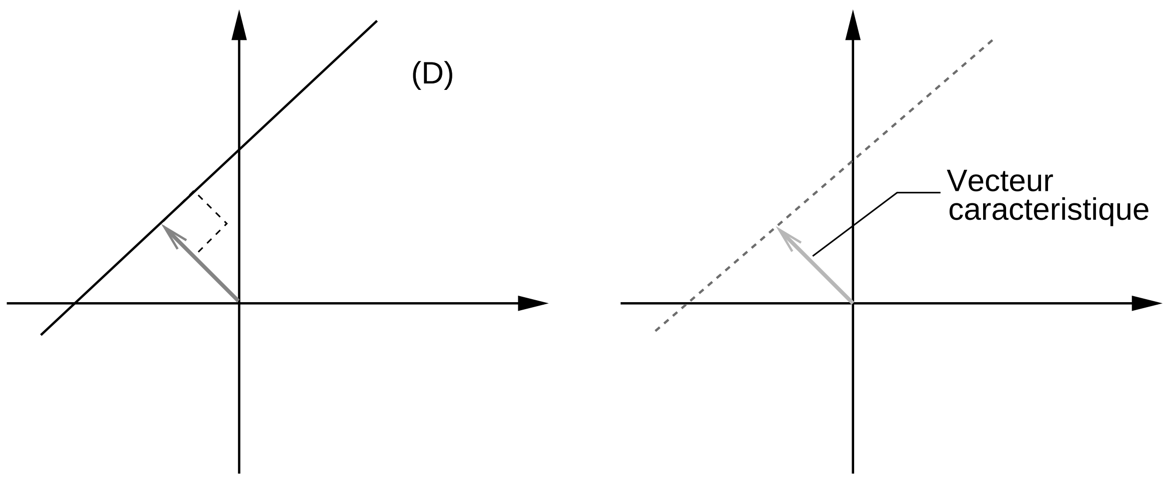
On va utiliser une détection de gradient :



Le problème : comment identifier les droites dans l'image du gradient ?

Comment caractériser une droite ?

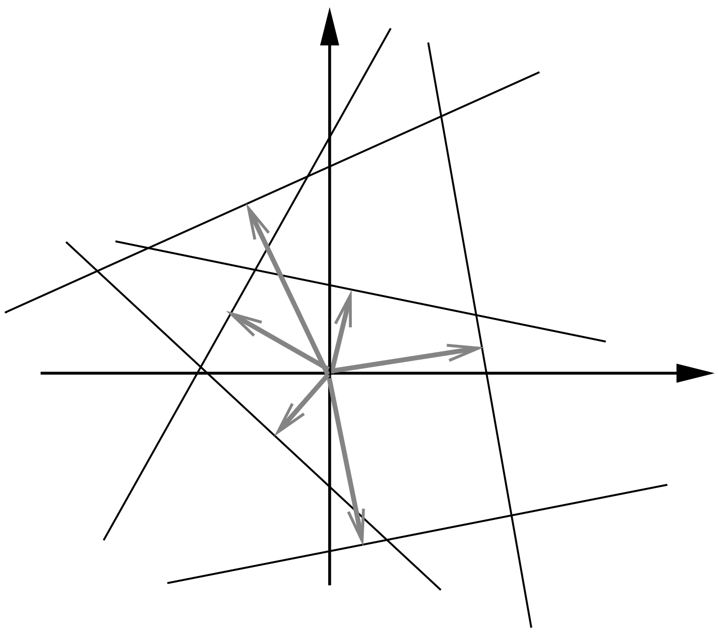
Le plus simple : $y = ax + b$. Ce n'est pas le plus utile pour nous :

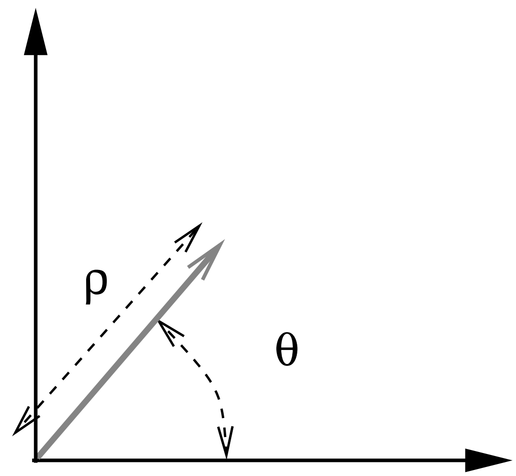


On va utiliser le vecteur caractéristique ci-dessus : c'est le vecteur normal à la droite mesurant la distance entre la droite et l'origine.

Paramétrisation de la droite

On considère la paramétrisation suivante, qui permet de considérer tous les cas de droites du plan.





On peut faire les remarques suivantes :

Une droite est caractérisée par :

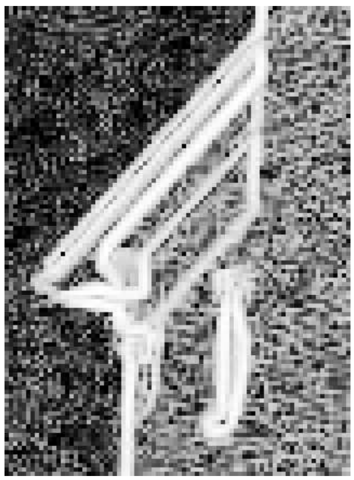
- la norme ρ.
- l'orientation θ

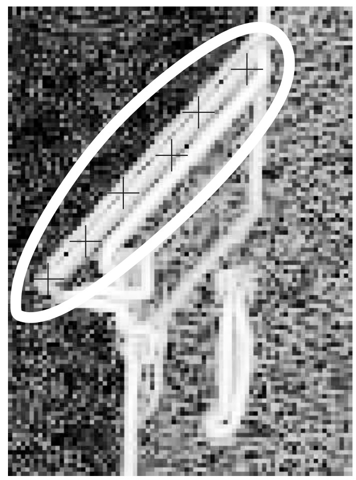
du vecteur caractéristique.

toutes les droites sont caractérisables par un couple.

Norme du gradient et droite

Question : qu'est ce qui donne l'impression d'une droite sur l'image des contours ?





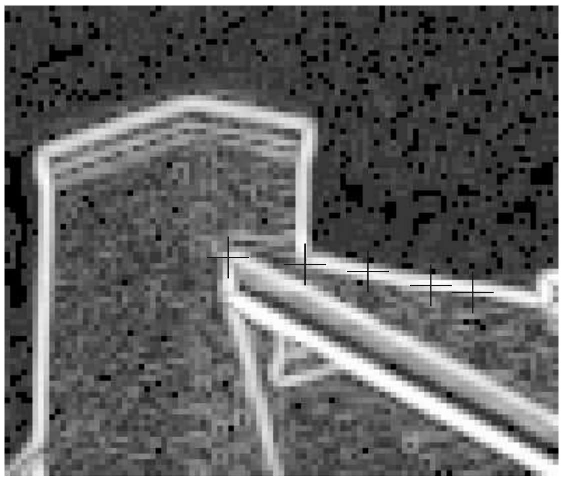
Une droite de l'image semble être caractérisée par l'alignement des discontinuités.

Conclusion : Une solution pour détecter les droites est de détecter les alignements de points de discontinuités.

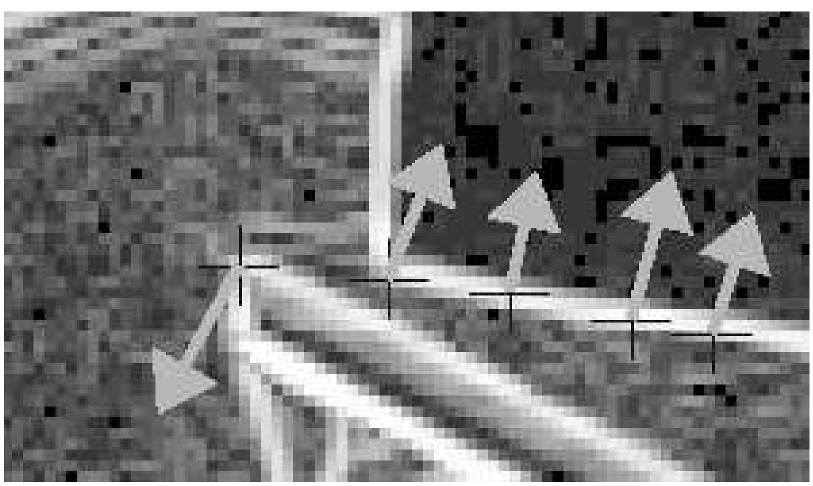
Mais ce n'est pas suffisant...

Direction du gradient et droite

Sur l'exemple suivant on remarque qu'il n'est pas suffisant de détecter les alignements de discontinuités fortes :



Il faut considérer les directions des contours :



Rappel Contours et gradient

Le gradient d'une image $\nabla u = (\delta_x, \delta_y)^T$ en un point donne la variation de l'image dans toutes les directions au travers d'un produit scalaire :

$$\Delta_\theta u \simeq \nabla u . v_\theta$$

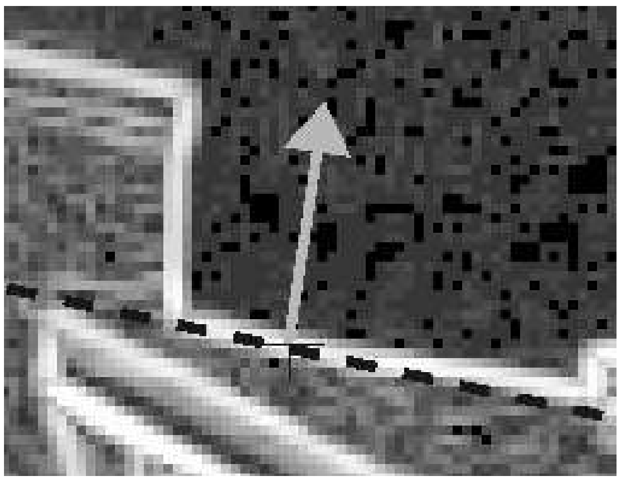
où $v = (\cos(\theta), \sin(\theta))^T$ est le vecteur unitaire de la direction θ.

La $\mathbf{direction\ du\ gradient}$ est la direction de variation maximale de l'image :

$$\theta = \arctan \frac{\delta_y}{\delta_x} \tag{6.1}$$

Gradient et droites

Lorsque l'on calcule la direction du gradient, cela donne la normale à la droite plausible.



Conclusion : chaque point tel que le norme du gradient $G = \sqrt{\delta_x^2 + \delta_y^2}$ soit plus grande qu'un certain seuil $G_{\min}$ peut être vu comme une hypothèse de droite de direction normale θ, la direction du gradient.

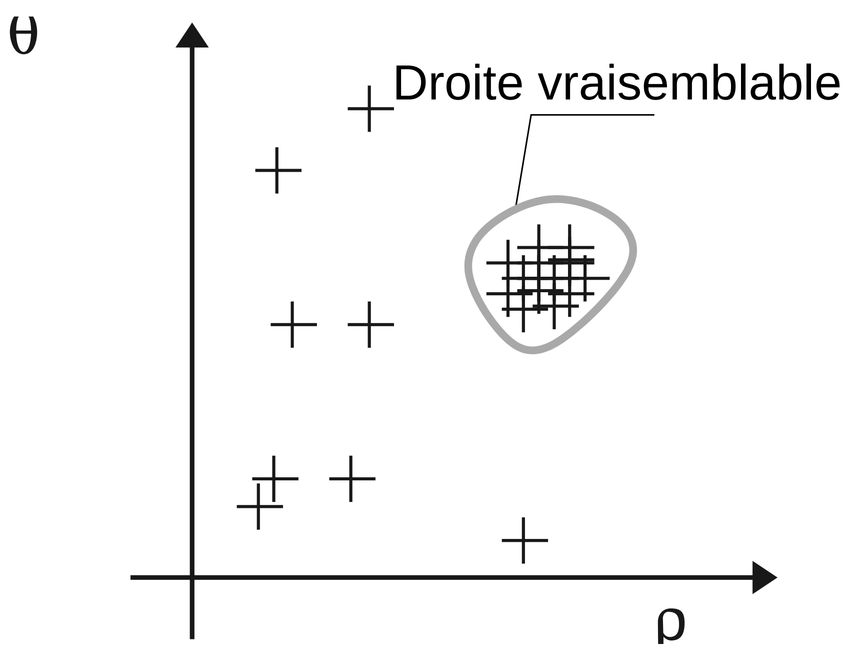
Et ρ ? On vérifie que :

$$\rho = x\cos(\theta) + y\sin(\theta)$$

Mécanisme de vote

La transformée de Hough repose sur le mécanisme de vote suivant :

1. On parcourt les pixels de l'image,
2. On calcule en chaque point le gradient, sa norme et sa direction,
3. On en déduit l'hypothèse de droite (ρ, θ),
4. Si le gradient est suffisamment grand ($G \geq G_{\min}$), on vote, c'est à dire que l'on rajoute un point dans l'espace des droites :



Trouver les droites

Il faut ensuite trouver les amas de points dans l'espace (ρ, θ).

Cela commence par un lissage. Ensuite : extraction des maxima locaux ou seuillage.

Identification des droites

Une fois que l'on a identifié les maximaux locaux, on en déduit les droites présentes dans l'image.

En pratique

On applique les étapes suivantes :

1. Discrétiser l'espace des paramètres ρ, θ en $\rho_1, \ldots, \rho_N$ et $\theta_1, \ldots, \theta_M$.

2. Parcourir les pixels, pour chacun calculer la norme et la direction du gradient. Si le gradient est suffisamment grand, calculer le ρ et le θ correspondant. Déduire le couple ρ_i et θ_j correspondant dans l'espace des paramètres discrétisés,

3. Lisser et optimiser : c'est à dire trouver les couples $(\rho_{i_k}, \theta_{j_k})$ pour lesquels le nombre de votes a été suffisant.

Conclusion

La transformée de Hough est un outil efficace pour trouver les droites dans une image.

Il existe d'autres transformées de Hough, dites transformées de Hough généralisées pour extraire d' autres formes comme les cercles ou les ellipses.

L'extraction des droites est souvent une première étape de la détection d'objets.

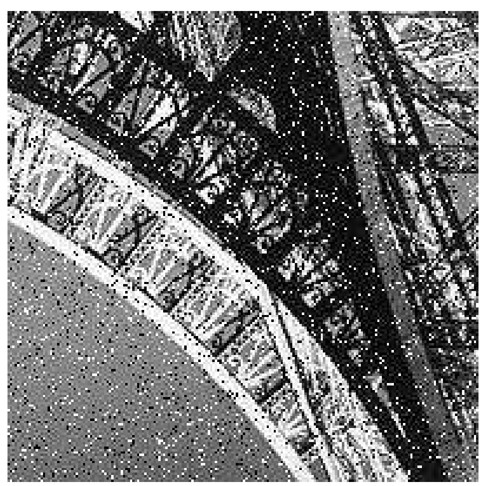
Le bruit

Le bruit est une altération de l'image : toute l'information pertinente dans l'image n'est pas simplement accessible.

Origine :

- Acquisition
  - illumination
  - perturbations
- Transmission
- Stockage
  - échantillonnage,
  - quantification

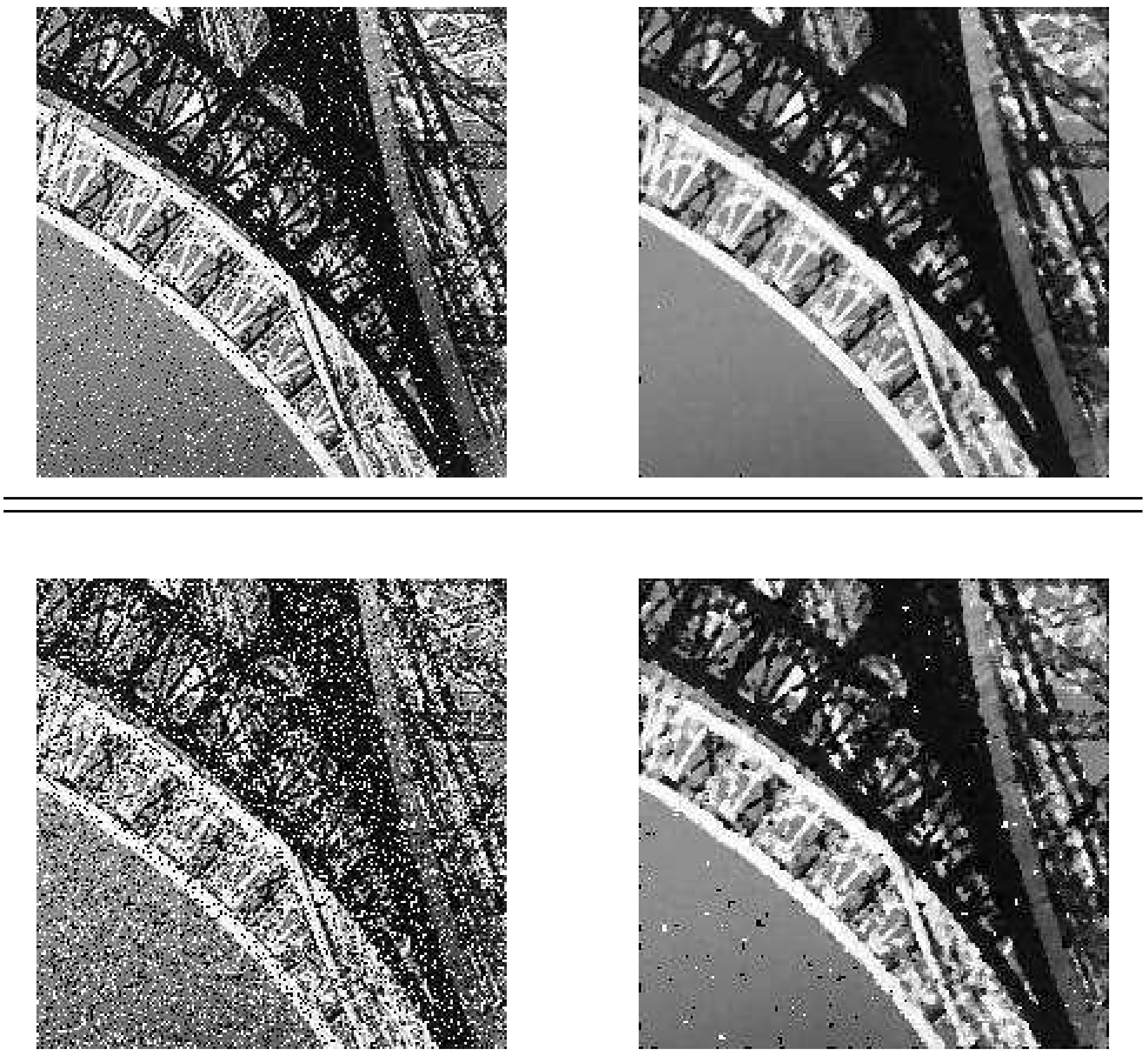
Exemple :



Le filtrage

Consiste à retrouver le maximum d'information sous l'image bruitée.

Exemple :

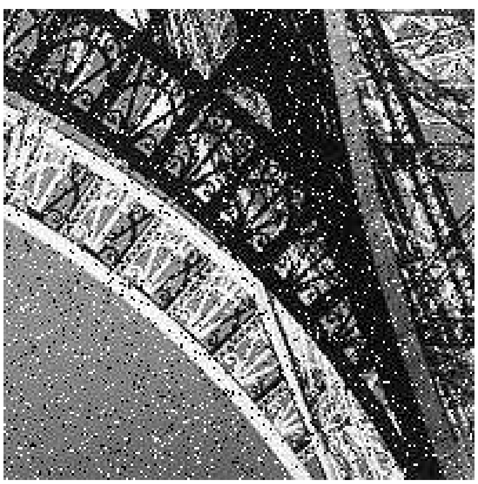


Bruit poivre et sel

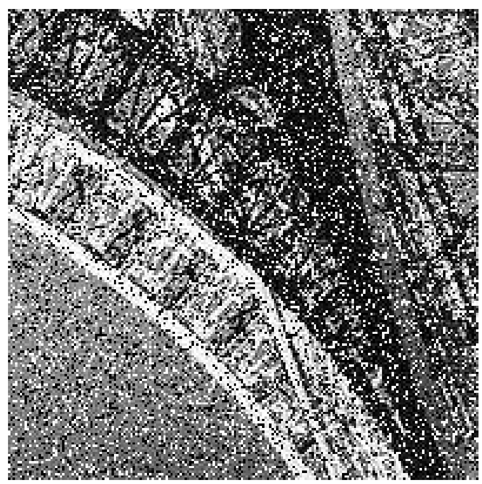
Définition : Le bruit “poivre et sel à x%” est obtenu en remplaçant

- $x/2$ % des pixels de l’image par des pixels blancs
- et $x/2$ % des pixels de l’image par des pixels noirs.

10 %

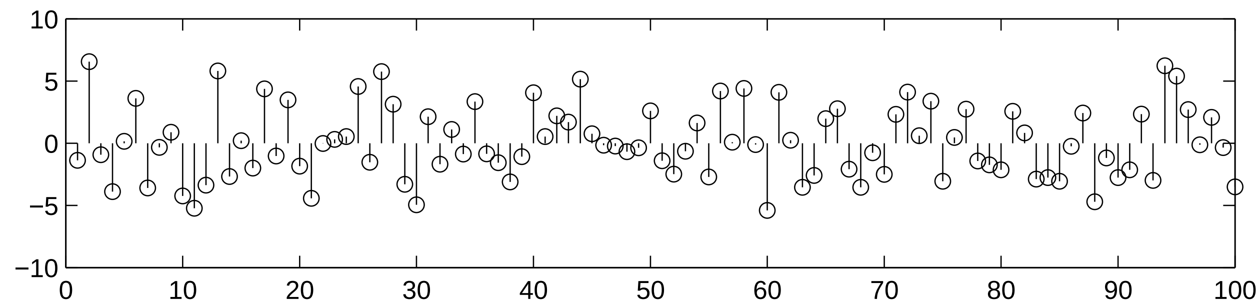


30 %



Bruit Gaussien (1)

On génère au hasard une suite de valeurs réelles $v_0, \ldots, v_N$:



On ajoute successivement à chaque pixel de l'image une valeur v_i :

$$u(0,0) = u(0,0) + v_0 \tag{7.1}$$

$$u(0,1) = u(0,1) + v_1 \tag{7.2}$$

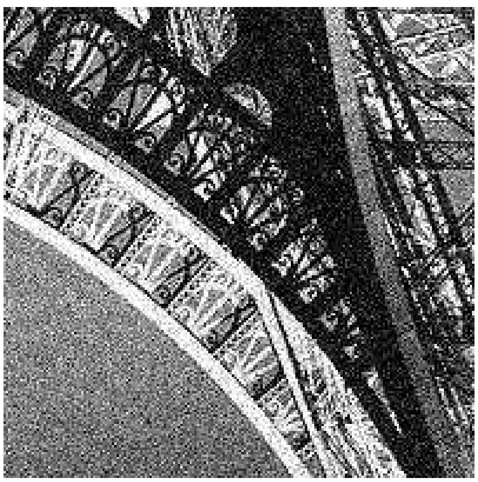
$$\vdots \tag{7.3}$$

$$u(nl,nc) = u(nl,nc) + v_N \tag{7.4}$$

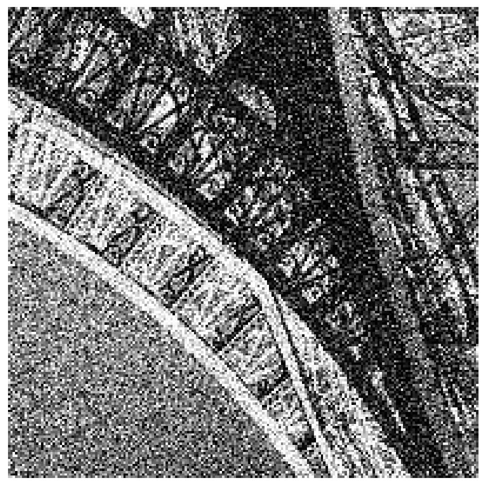
Le bruit est dit Gaussien parce que la génération aléatoire est faite en suivant une **Loi gaussienne** (de moyenne $m = 0$).

Bruit Gaussien (2)

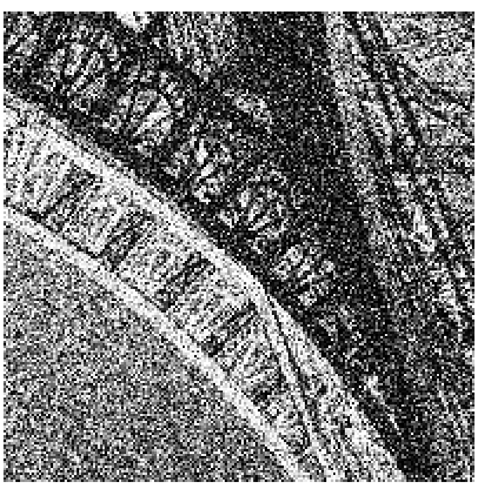
Le bruit Gaussien est caractérisé par sa variance (ou de manière équivalente par l'écart type).



$\sigma = 0.01 * 256$



$\sigma = 0.05 * 256$



$\sigma = 0.1 * 256$

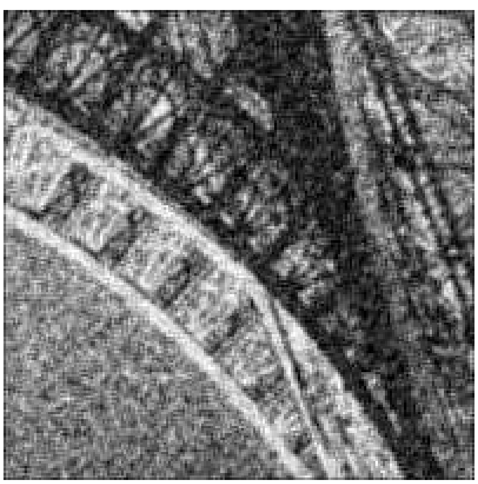
$\sigma = 0.20 * 256$

Filtrage Linéaire

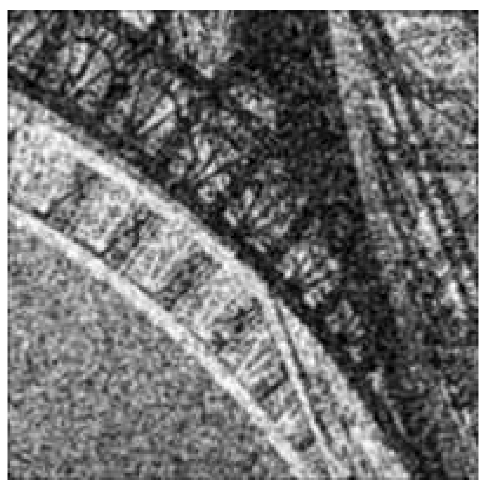
Le filtrage linéaire repose sur l'utilisation de masques de convolution (cf cours sur les contours), comme par exemple :

$$M_1 = \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array} \qquad M_2 = \frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

En appliquant M_1 et M_2 sur l'image $\sigma = 0.1 * 256$ de la page précédente :



Avec M_1



Avec M_2

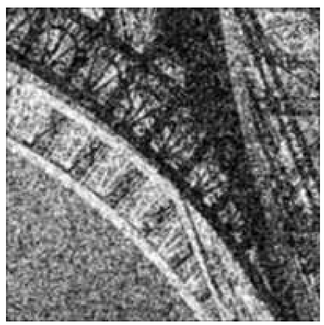
Filtrage linéaire itéré

Pour améliorer les filtres linéaires on peut :
- chercher les meilleures valeurs du masque,
- augmenter la taille du masque,
- Appliquer plusieurs fois de suite le même masque.

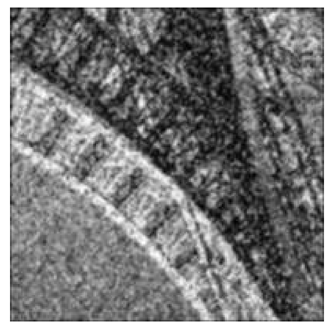
Remarque : On peut trouver un lien entre ces trois points.

Exemple : Si on applique M_2 :

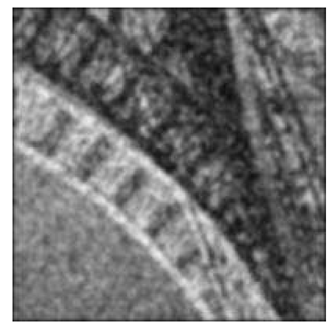
0 fois



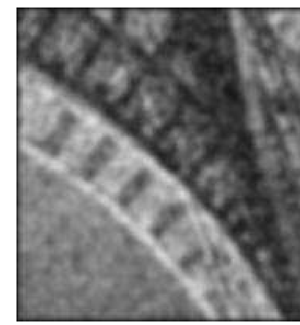
1 fois



2 fois



5 fois



10 fois

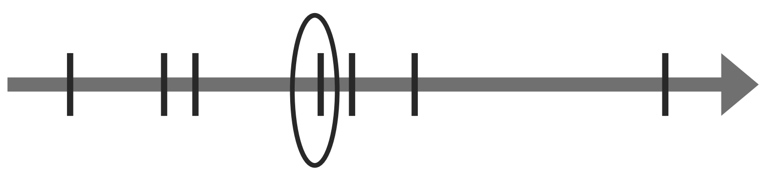
Filtrage Non linéaire

Le filtre non linéaire le plus utilisé est **le filtre médian**. Ce filtre remplace la valeur d'un pixel par la médiane des valeurs de ses voisins.

Rappel : La médiane d'un ensemble de valeur, est la valeur " du milieu".

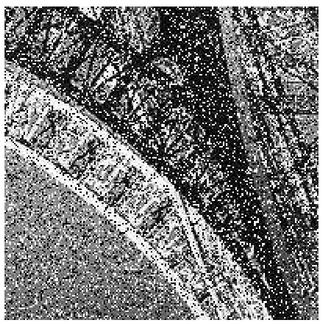
Algo :

1. Chercher les valeurs des voisins du pixel courant,
2. Ordonner ces valeurs (**tri**),
3. Sélectionner celle du milieu.

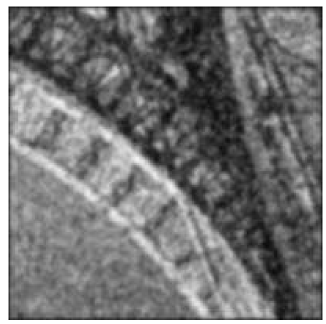


Filtrage Non linéaire (2)

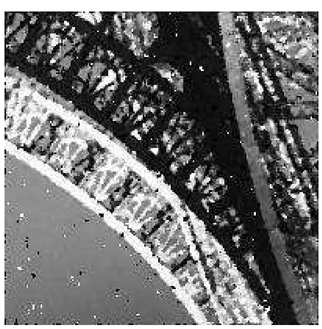
Les avantages du filtre médian :
- il préserve les contours,
- il est plus robuste à certain bruits (par exemple, le bruit de poivre et sel.)



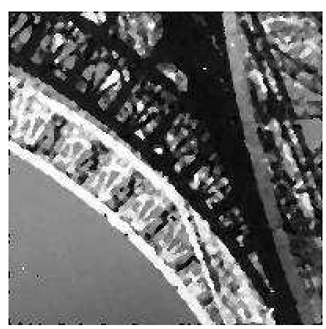
poivre et sel 30 %



M_2 5 fois



Médian



Médian 2 fois