Traitement et analyse d'images numériques

Partie 1: Introduction, Histogrammes, Premières transformations

Pierre Maurel

Visages, IRISA/INRIA

pierre.maurel@irisa.fr

http://www.normalesup.org/~pmaurel/IMA/

Organisation du cours

Traitement et analyse des images numériques : Pierre Maurel

- 12h de cours
- 12h de TP

Synthèse d'images : Fabrice Lamarche

- 12h de cours
- 12h de TP

Introduction

Traitement des images : cas particulier du traitement du signal

- notion de signal : observation de phénomène.
- quantités dépendantes du temps, de l'espace ou de la fréquence.
- modélisation sous forme de fonction d'une ou plusieurs variables.

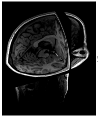
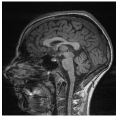
Traitement des images : cas particulier du traitement du signal

- notion de signal : observation de phénomène.
- quantités dépendantes du temps, de l'espace ou de la fréquence.
- modélisation sous forme de fonction d'une ou plusieurs variables.
  - 1D : variable $\rightarrow$ **le temps** t (exemple : le son) 

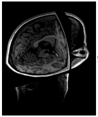
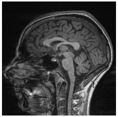
Traitement des images : cas particulier du traitement du signal

- notion de signal : observation de phénomène.
- quantités dépendantes du temps, de l'espace ou de la fréquence.
- modélisation sous forme de fonction d'une ou plusieurs variables.
  - 1D : variable $\rightarrow$ **le temps** t (exemple : le son) 
  - 2D : les images, variables $\rightarrow$ **l'espace** (x, y) 

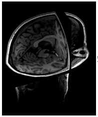
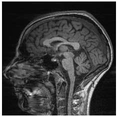
Traitement des images : cas particulier du traitement du signal

- notion de signal : observation de phénomène.
- quantités dépendantes du temps, de l'espace ou de la fréquence.
- modélisation sous forme de fonction d'une ou plusieurs variables.
  - 1D : variable $\rightarrow$ **le temps** t (exemple : le son) 
  - 2D : les images, variables $\rightarrow$ **l'espace** (x, y) 
  - 3D : images 3D (x, y, z), , 

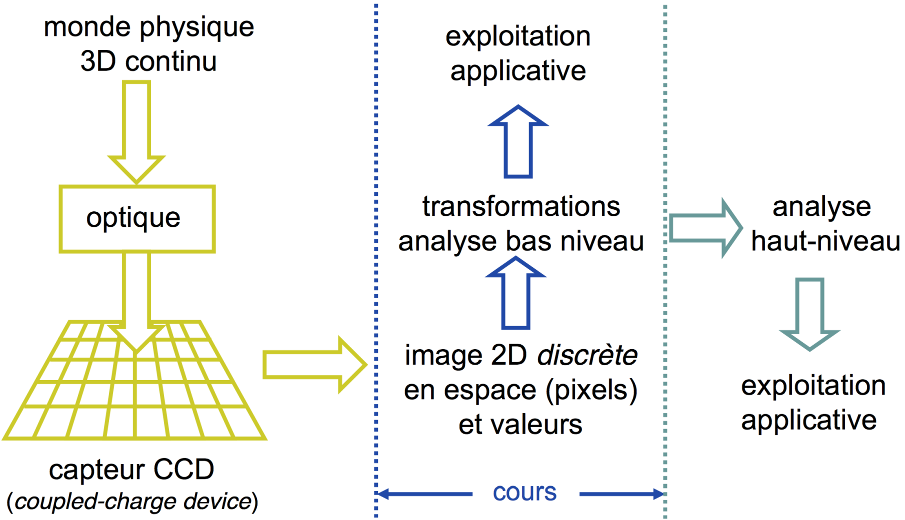
Traitement des images : cas particulier du traitement du signal

- notion de signal : observation de phénomène.
- quantités dépendantes du temps, de l'espace ou de la fréquence.
- modélisation sous forme de fonction d'une ou plusieurs variables.
  - 1D : variable $\rightarrow$ **le temps** t (exemple : le son) 
  - 2D : les images, variables $\rightarrow$ **l'espace** (x, y) 
  - 3D : images 3D (x, y, z), ,  vidéos $((x, y), t)$ 

Traitement des images : cas particulier du traitement du signal

- notion de signal : observation de phénomène.
- quantités dépendantes du temps, de l'espace ou de la fréquence.
- modélisation sous forme de fonction d'une ou plusieurs variables.
  - 1D : variable $\rightarrow$ **le temps** t (exemple : le son) 
  - 2D : les images, variables $\rightarrow$ **l'espace** (x, y) 
  - 3D : images 3D (x, y, z), ,  vidéos $((x, y), t)$ 
  - 4D : volumes 3D évoluant dans le temps

Contexte


monde physique
3D continu
optique
capteur CCD
(*coupled-charge device*)
exploitation
applicative
transformations
analyse bas niveau
image 2D *discrète*
en espace (pixels)
et valeurs
cours
analyse
haut-niveau
exploitation
applicative

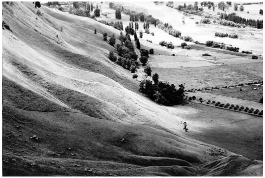
Domaines d'application ...

- Depuis les années 60
  - Astronomie (images multispectrales, télescopes terrestres et spatiaux)
  - Imagerie médicale (radios, échographie, angiographie, IRM, etc.)
  - Imagerie biologique (microscopie, puce ADN)
  - Image satellite (météo, climato, océano, surveillance, ressources)
  - défense, sécurité, surveillance (bio-métrie : iris, veines, empreintes, visages, plaques, etc.)
  - Robotique
  - Contrôle non-destructif (qualité de production, intégrité des structures)
  - . . .
- Plus récemment
  - Capture et analyse du mouvement humain (prothèses, geste sportif, interfaces gestuelles)
  - Réalité augmentée, réalité virtuelle
  - Tatouage numérique (Watermarking)
  - Assistance à la conduite
  - . . .

Domaines d'application ...

- Transformations
  - Compression
  - Transformations géométriques
  - Changement de contraste, de couleurs, etc.
  - Débruitage, restauration et rehaussement
  - Édition de contenu
- Analyse "bas niveau"
  - Calcul de mesures globales (statistiques, distributions, contenu fréquentiel)
  - Calcul de mesures locales (dérivées, statistiques locales)
  - Partitionnement en régions "homogènes" (texture, couleur)
  - Extraction de "primitives" (contours, coins, blobs, lignes)

Exemples 1/6

 $\rightarrow$ 

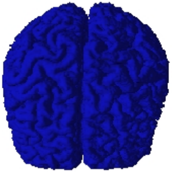
 $\rightarrow$ 

 $\rightarrow$ 

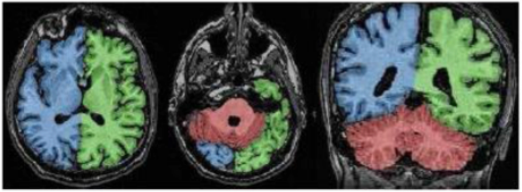
 $\rightarrow$ 

Exemples 2/6

$\longrightarrow$



Hémisphère gauche,
hémisphère droit, cervelet



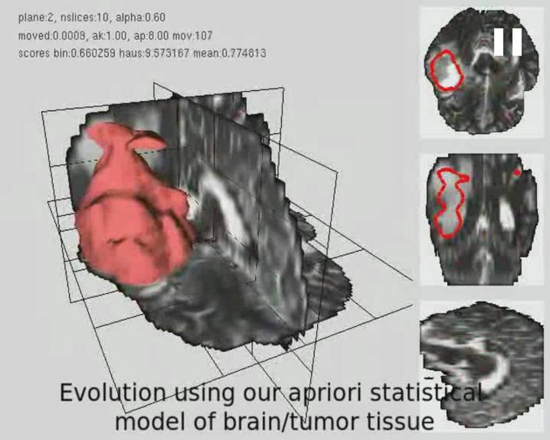
Exemples 3/6

Geo-Referenced 3D Reconstruction : Fusing Public Geographic Data and Aerial Imagery. M.Maurer, M.Rumpler, A.Wendel, C.Hoppe, A.Irschara, and H.Bischof at the Institute for Computer Graphics and Vision, Graz University of Technology
`http://www.youtube.com/watch?v=vYXhVG7VMy0`

Exemples 4/6

www.mobileye.com

Exemples 5/6



2008 : Cobzas, Birkbeck, Schmidt, Jagersand, Murtha A.
3D Variational Brain Tumor Segmentation using a High Dimensional Feature Set

Exemples 6/6

Demo Tractographie

http://med.inria.fr/

Les images numériques

- image = signal bidimensionnel

Les images numériques

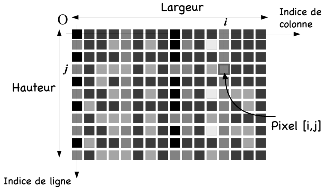
- image = signal bidimensionnel
- image **analogique** : image formée sur la rétine de l'oeil, image obtenue par la photo argentique classique

Les images numériques

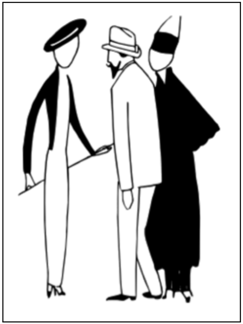
- image = signal bidimensionnel
- image **analogique** : image formée sur la rétine de l'oeil, image obtenue par la photo argentique classique
- image **numérique** : signal numérique composé d'unités élémentaires (pixels) qui représentent chacun une portion de l'image.

Les images numériques

- image = signal bidimensionnel
- image **analogique** : image formée sur la rétine de l'oeil, image obtenue par la photo argentique classique
- image **numérique** : signal numérique composé d'unités élémentaires (pixels) qui représentent chacun une portion de l'image. Caractérisée par :
  - le nombre de **pixels** (largeur, hauteur)
  - étendue des teintes de gris ou des couleurs que peut prendre chaque pixel $\rightarrow$ **dynamique de l'image**.



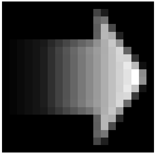
Images binaires (noir ou blanc)



```
%--------------------------------------------------------------
\begin{frame}
\frametitle{Traitement des images : cas particulier du traitement du
        \begin{itemize}
                \item notion de signal : observation de phénomène.
                \item quantités dépendantes du temps, de l'espace ou
                \item modélisation sous forme de fonction d'une ou p
                        \begin{itemize}
                                \pause
                        \item 1D : variable $\rightarrow$ \t
                                \pause
                        \item 2D : les images, variables $\r
                                \pause
                        \item 3D : images 3D ($x,y,z$), \rai
                                \pause
                        \item 4D : volumes 3D évoluant dans
                        \end{itemize}
        \end{itemize}
```

Images en niveau de gris

- généralement : à valeurs dans $\{0, \dots, 255\}$
- par convention : 0=noir, 255=blanc

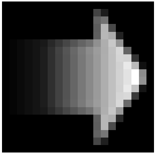


$\longleftrightarrow$

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	0	0	0	0	0	0	0	67	28	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	99	114	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	93	182	72	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	95	183	172	34	0	0	0	0
0	5	10	15	20	25	40	55	70	85	100	113	144	174	200	147	0	0	0	0
0	5	10	15	20	29	47	67	85	104	122	140	156	173	193	215	99	0	0	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	213	211	52	0	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	209	233	186	16	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	210	226	255	139	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	210	226	255	146	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	209	233	194	21	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	213	217	57	0	0
0	5	10	15	20	29	47	67	85	104	122	140	156	173	193	216	106	0	0	0
0	5	10	15	20	25	40	55	70	85	100	113	144	175	199	153	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	95	183	176	40	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	93	183	76	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	99	119	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	70	31	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	0

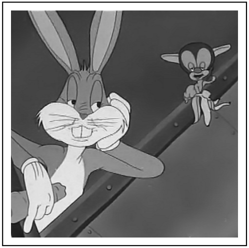
Images en niveau de gris

- généralement : à valeurs dans $\{0, \ldots, 255\}$
- par convention : 0=noir, 255=blanc



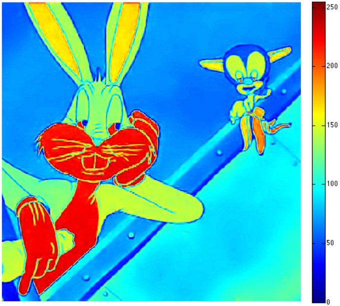
$\longleftrightarrow$

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	0	0	0	0	0	0	0	67	28	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	99	114	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	93	182	72	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	95	183	172	34	0	0	0	0
0	5	10	15	20	25	40	55	70	85	100	113	144	174	200	147	0	0	0	0
0	5	10	15	20	29	47	67	85	104	122	140	156	173	193	215	99	0	0	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	213	211	52	0	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	209	233	186	16	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	210	226	255	139	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	210	226	255	146	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	209	233	194	21	0
0	5	10	15	20	29	47	66	83	102	120	137	154	174	192	213	217	57	0	0
0	5	10	15	20	29	47	67	85	104	122	140	156	173	193	216	106	0	0	0
0	5	10	15	20	25	40	55	70	85	100	113	144	175	199	153	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	95	183	176	40	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	93	183	76	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	99	119	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	70	31	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	0



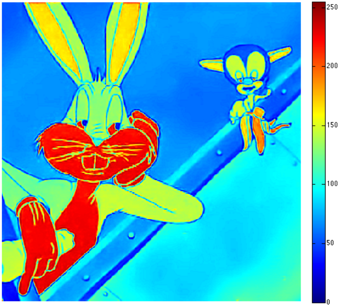
Images couleurs

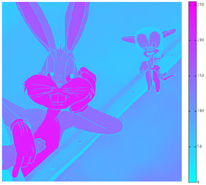
- "fausses couleurs"



Images couleurs

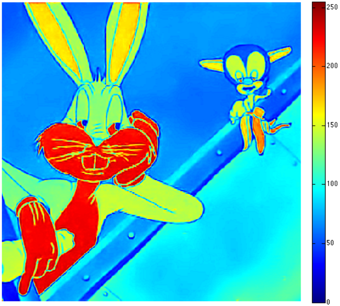
- "fausses couleurs"

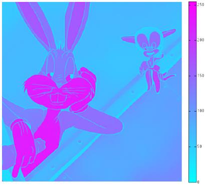


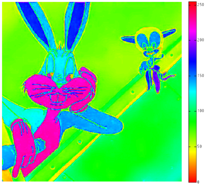


Images couleurs

- "fausses couleurs"







Images couleurs

- "fausses couleurs" : intérêt $\rightarrow$ images indexées
  - Si on a N de couleurs distinctes dans l'image et que $N \ll 256^3 = 16777216$
  - Gain de place (en mémoire)

GIMP : exple 1   exple 2

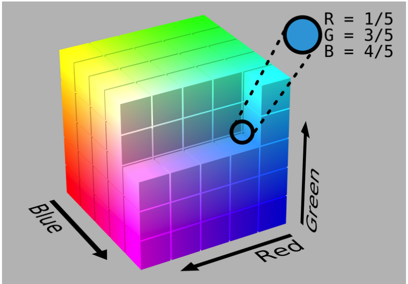
- "fausses couleurs" : peut aussi permettre de mettre en évidence des variations mineures de niveau de gris (images biomédicales, satellitaires)

Images couleurs

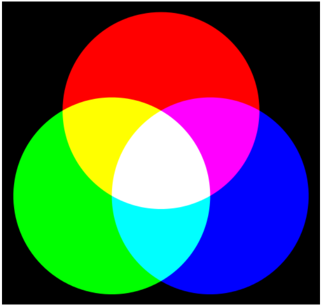
- Bcp plus souvent : espace couleur est basé sur la synthèse additive des couleurs $\rightarrow$ mélange de 3 composantes
- exemple classique : Red-Green-Blue (RGB) ou Rouge-Vert-Bleu (RVB)

Images couleurs

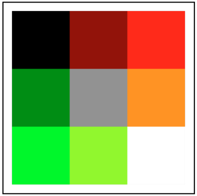
- Bcp plus souvent : espace couleur est basé sur la synthèse additive des couleurs $\rightarrow$ mélange de 3 composantes
- exemple classique : Red-Green-Blue (RGB) ou Rouge-Vert-Bleu (RVB)
- Images à valeurs dans $\{0, \ldots, 255\}^3$



Wikipedia



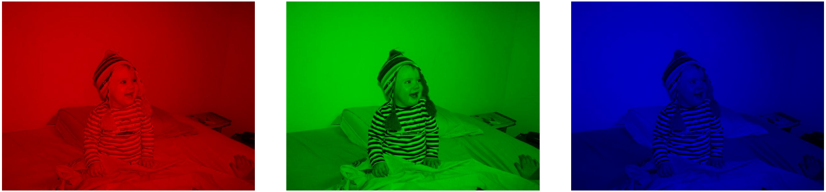
Images couleurs



$$\longleftrightarrow \begin{bmatrix} (0,0,0) & (128,0,0) & (255,0,0) \\ (0,128,0) & (128,128,128) & (255,128,0) \\ (0,255,0) & (128,255,0) & (255,255,255) \end{bmatrix}$$

ou encore $\longrightarrow \left(\begin{bmatrix} 0 & 128 & 255 \\ 0 & 128 & 255 \\ 0 & 128 & 255 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 128 & 128 & 128 \\ 255 & 255 & 255 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 128 & 0 \\ 0 & 0 & 255 \end{bmatrix} \right)$

Images couleurs



R G B

Images couleurs

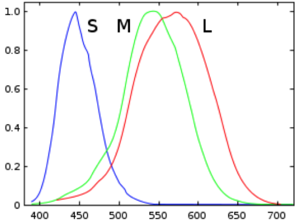
R

G

B

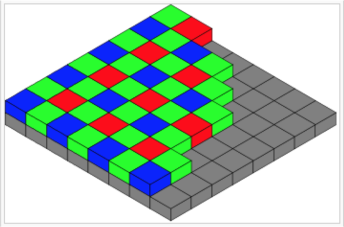
Images couleurs

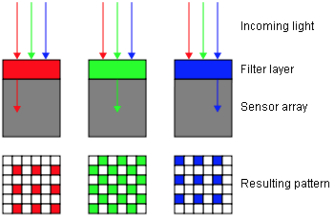
- **Onde lumineuse** : superposition d'une infinité d'ondes pures
- **Système visuel**
  - tout signal lumineux est bien approché, *visuellement*, par une combinaison linéaire de trois primaires
  - Constat empirique (mélange en peinture et imprimerie)
  - Explication anatomique : trois types de cônes sur la rétine sensibles sur trois plages différentes de longueurs d'onde grossièrement associées au rouge, vert et bleu.



Capteur CCD (coupled-charge device)

- grille de photo-senseurs : chacun convertit les photons en courant électrique sur un petit intervalle de temps fixe ; la réponse peut dépendre de la longueur d'onde des photons
- réponses "récoltées" et numérisées (quantifiées sur un nombre fini de valeurs)
- images couleurs : trois types de capteurs, spécialisés sur le vert, le bleu et le rouge (mosaïque de Bayer [Kodak, 1976]) ; interpolation pour avoir autant de pixels que de cellules





Espaces colorimétriques

- Plusieurs espaces linéaires
  - Red-Green-Blue : RGB
  - Cyan-Yellow-Magenta : CYM (synthèse soustractive)
  - XYZ

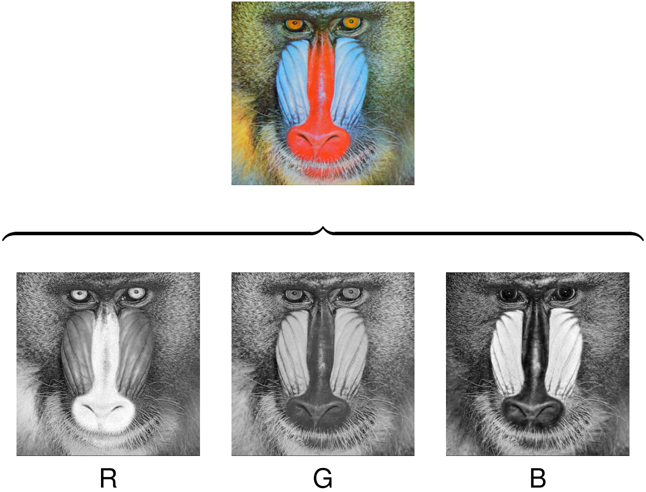
Espaces colorimétriques

- Plusieurs espaces linéaires
  - Red-Green-Blue : RGB
  - Cyan-Yellow-Magenta : CYM (synthèse soustractive)
  - XYZ
- Mais : peu conformes à la perception psycho-visuelle
  - la teinte est une dimension " fermée " (roue des couleurs)
  - distances dans un espace linéaire et dissimilarités perçues ne sont pas en bon accord

Espaces colorimétriques

- Plusieurs espaces linéaires
  - Red-Green-Blue : RGB
  - Cyan-Yellow-Magenta : CYM (synthèse soustractive)
  - XYZ
- Mais : peu conformes à la perception psycho-visuelle
  - la teinte est une dimension " fermée " (roue des couleurs)
  - distances dans un espace linéaire et dissimilarités perçues ne sont pas en bon accord
- Plusieurs espaces non-linéaires
  - xyY
  - Hue(teinte)-Saturation-Value : HSV
  - Lab

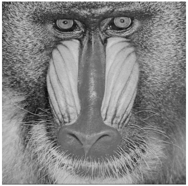
Espaces colorimétriques : RGB



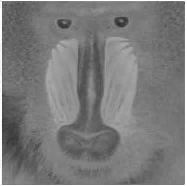
Espaces colorimétriques : Yuv

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,14713 & -0,28886 & 0,436 \\ 0,615 & -0,51498 & -0,10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

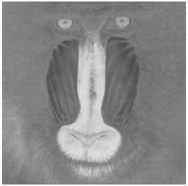
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1,13983 \\ 1 & -0,39465 & -0,58060 \\ 1 & 2,03211 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$



Y



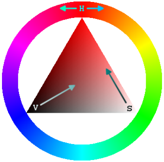
u



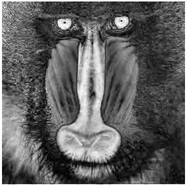
v

Espaces colorimétriques : HSV (Hue Saturation Value)

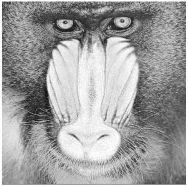
(Transformation non linéaire)



H



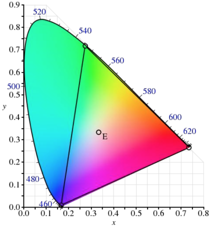
S

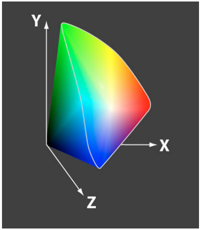


V

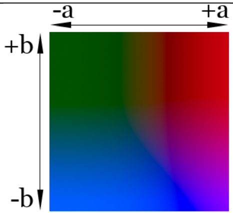
Espaces colorimétriques

CIE-XYZ

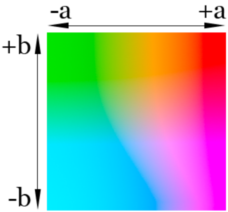




CIE-Lab

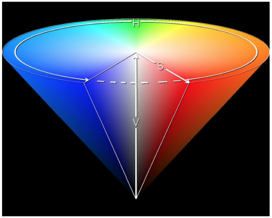


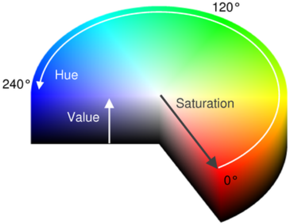
Luminosité à 25%



Luminosité à 75%

CIE-HSV





Histogramme d'une image (en niveau de gris)

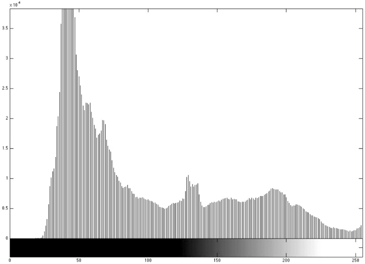
Histogramme d’une image

Définition

Histogramme d’une image I : fonction discrète qui associe à chaque valeur d’intensité le nombre de pixels prenant cette valeur.

$$\begin{array}{rccl} h_I : & \{0,\ldots,255\} & \rightarrow & \mathbb{N} \\ & n & \mapsto & \text{Card}\{(x,y) | I(x,y) = n\} \end{array}$$

I



h_I

Histogramme d'une image

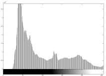
- Si I est de taille $p \times q$ à valeurs dans $\{0, \dots, 255\}$, $\sum_n h_I(n) =$

Histogramme d'une image

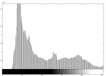
- Si I est de taille $p \times q$ à valeurs dans $\{0, \ldots, 255\}$, $\sum_n h_I(n) = p * q$

Histogramme d'une image

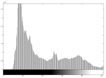
- Si I est de taille $p \times q$ à valeurs dans $\{0, \ldots, 255\}$, $\sum_n h_I(n) = p * q$

- Pour $I =$ , on a $h_I =$ . Pour $I =$ , $h_I =$?

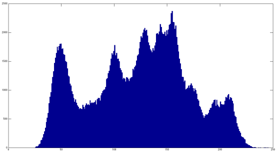
Histogramme d'une image

- Si I est de taille $p \times q$ à valeurs dans $\{0, \ldots, 255\}$, $\sum_n h_I(n) = p * q$
- Pour $I =$ , on a $h_I =$ . Pour $I =$ , $h_I =$? idem

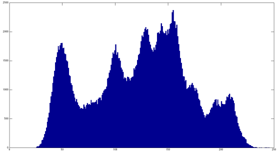
Histogramme d'une image

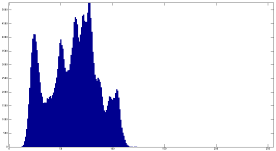
- Si I est de taille $p \times q$ à valeurs dans $\{0, \ldots, 255\}$, $\sum_n h_I(n) = p * q$
- Pour $I =$ , on a $h_I =$ . Pour $I =$ , $h_I =$? idem
- Deux images différentes peuvent avoir le même histogramme, **ce n'est pas** une caractéristique de l'image

Histogramme d'une image

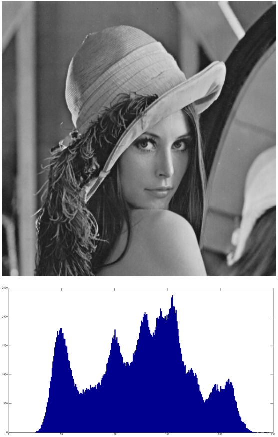


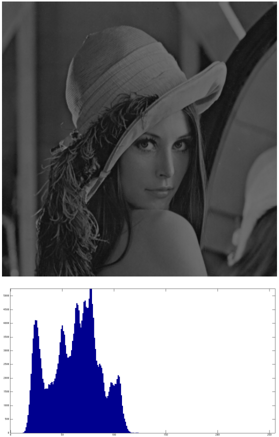
Histogramme d'une image

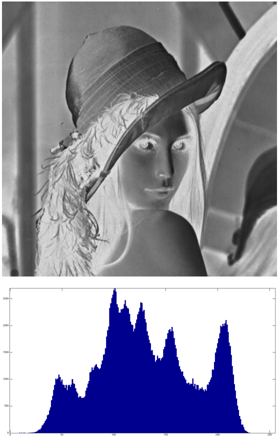




Histogramme d'une image

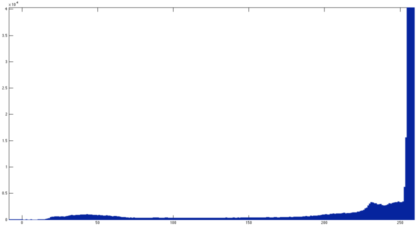






Histogramme d'une image

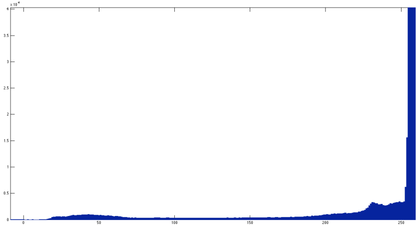
- Que peut-on dire d'une photo ayant pour histogramme



?

Histogramme d'une image

- Que peut-on dire d'une photo ayant pour histogramme

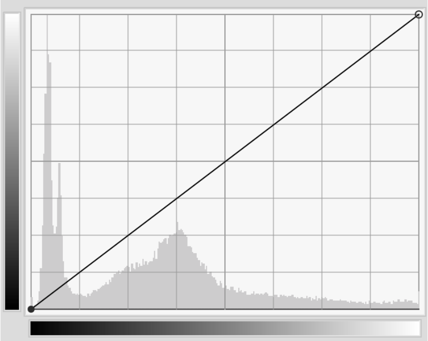


?

- Il s'agit d'une image **sur-exposée**

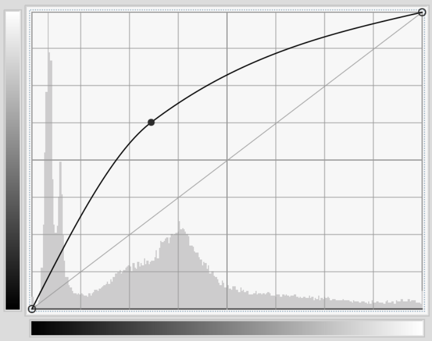
Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : $\boxed{\text{avec GIMP}}$



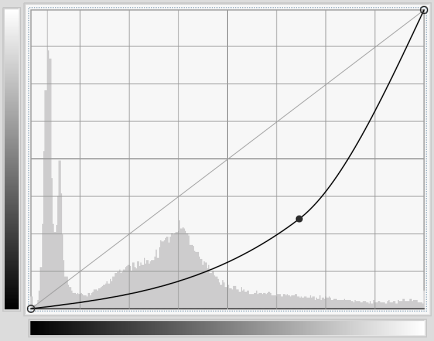
Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : avec GIMP



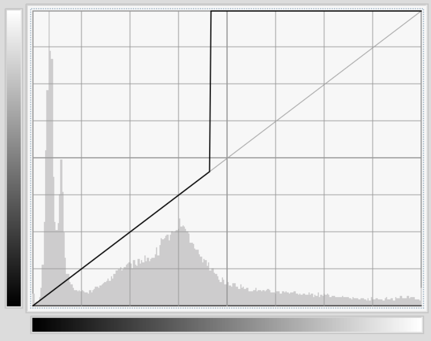
Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : avec GIMP



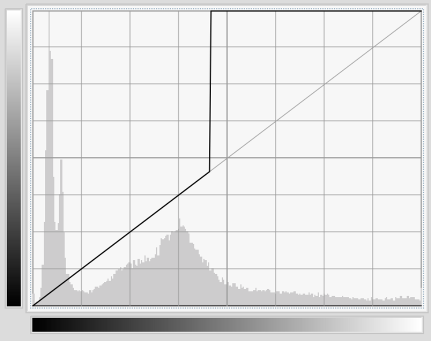
Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : avec GIMP



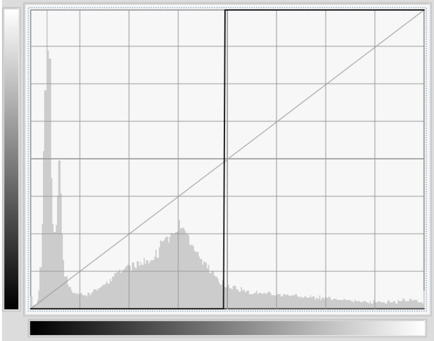
Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : $\boxed{\text{avec GIMP}}$



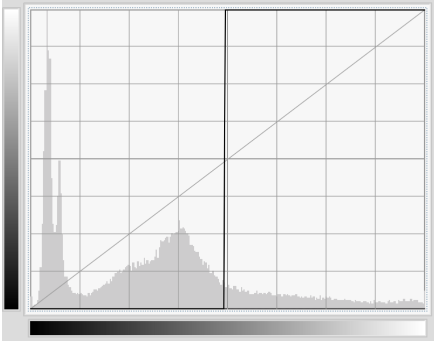
Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x,y) = \phi(I(x,y))$, avec ϕ fonction croissante
- Édition manuelle, exple : $\boxed{\text{avec GIMP}}$



Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : avec GIMP

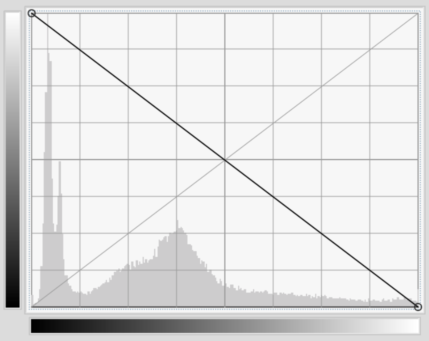


Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : avec GIMP

Histogramme d'une image

- Utile pour changer le contraste d'une image.
- On construit une image J ainsi : $J(x, y) = \phi(I(x, y))$, avec ϕ fonction croissante
- Édition manuelle, exple : $\boxed{\text{avec GIMP}}$

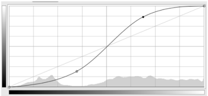


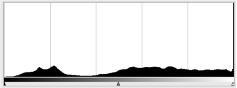
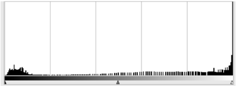
Histogramme d'une image

- Renforcement du contraste

Idée :

- pixel $<$ s $\rightarrow$plus sombres
- pixel $>$ s $\rightarrow$plus clairs

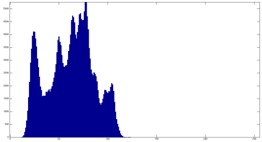



Normalisation/étirement d'histogrammes

- Si l'image I n'utilise pas l'ensemble de la dynamique possible :

$\longleftrightarrow$



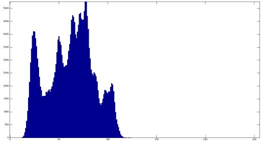
- On peut "étirer" l'histogramme. On cherche ϕ linéaire et croissante telle que

$$\min\left(\phi\left(I\left(x,y\right)\right)\right) \to 0 \quad \text{et} \quad \max\left(\phi\left(I\left(x,y\right)\right)\right) \to 255 \quad \text{(par exemple)}$$

- $\phi(a) =$

Normalisation/étirement d'histogrammes

- Si l'image I n'utilise pas l'ensemble de la dynamique possible :

 $\longleftrightarrow$ 

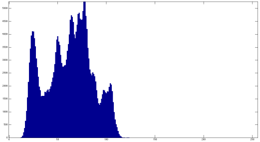
- On peut "étirer" l'histogramme. On cherche ϕ linéaire et croissante telle que

$$\min\left(\phi\left(I\left(x,y\right)\right)\right) \to 0 \quad \text{et} \quad \max\left(\phi\left(I\left(x,y\right)\right)\right) \to 255 \ \text{(par exemple)}$$

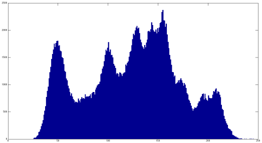
- $\phi(a) = 255\,\dfrac{a - \min(I)}{\max(I) - \min(I)}$

Normalisation/étirement d'histogrammes

- Image originale

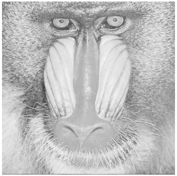
 $\longleftrightarrow$ 

- Image "normalisée"

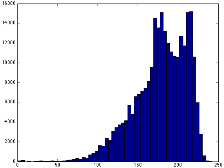
 $\longleftrightarrow$ 

Normalisation/étirement d'histogrammes

- Image originale



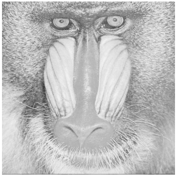
$\longleftrightarrow$



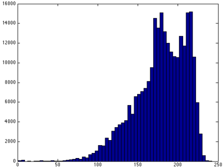
- Image "normalisée"

Normalisation/étirement d'histogrammes

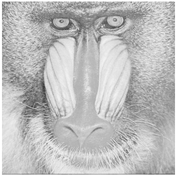
- Image originale



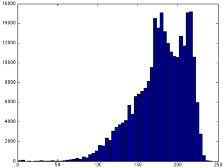
$\longleftrightarrow$



- Image "normalisée"



$\longleftrightarrow$



- Dans le cas où l'histogramme initial occupe déjà toute la plage de dynamique, **aucun changement n'est visible**.

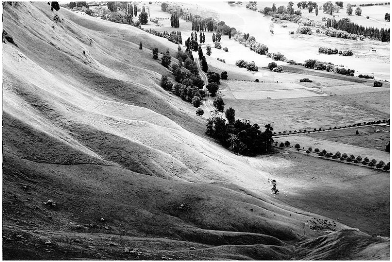
Égalisation d'histogramme

- ajustement automatique du contraste
- but : "étaler" les valeurs de l'image pour mieux répartir sur l'ensemble des valeurs possibles

Égalisation d'histogramme

- ajustement automatique du contraste
- but : "étaler" les valeurs de l'image pour mieux répartir sur l'ensemble des valeurs possibles
- Exemple (wikipédia)

$\longrightarrow$



Égalisation d'histogramme

Principes de l'algorithme

- x : image à valeurs dans $\{0, \ldots, L-1\}$
- intensités des pixels (x_k) : vues comme des variables aléatoires dans $[0, L-1]$ (continues)
- $p_x(X = x_k)$: densité de probabilité

Égalisation d'histogramme

Principes de l'algorithme

- x : image à valeurs dans $\{0, \dots, L-1\}$
- intensités des pixels (x_k) : vues comme des variables aléatoires dans $[0, L-1]$ (continues)
- $p_x(X = x_k)$: densité de probabilité
- Quelle transformation T doit on appliquer à x pour que p_s, la densité de probabilité associée à l'image $s = T(x)$, soit uniforme ?

Égalisation d'histogramme

Principes de l'algorithme

- x : image à valeurs dans $\{0, \dots, L-1\}$
- intensités des pixels (x_k) : vues comme des variables aléatoires dans $[0, L-1]$ (continues)
- $p_x(X = x_k)$: densité de probabilité
- Quelle transformation T doit on appliquer à x pour que p_s, la densité de probabilité associée à l'image $s = T(x)$, soit uniforme ?
- On peut montrer qu'il faut prendre

$$T(x_k) = (L-1) \int_0^{x_k} p_x(w) \, \mathrm{d}w \qquad \rightarrow \textit{Fonction de répartition} \text{ de } X$$

Égalisation d'histogramme

Principes de l'algorithme

- x : image à valeurs dans $\{0, \ldots, L-1\}$
- intensités des pixels (x_k) : vues comme des variables aléatoires dans $[0, L-1]$ (continues)
- $p_x(X = x_k)$: densité de probabilité
- Quelle transformation T doit on appliquer à x pour que p_s, la densité de probabilité associée à l'image $s = T(x)$, soit uniforme ?
- On peut montrer qu'il faut prendre

$$T(x_k) = (L-1) \int_0^{x_k} p_x(w)\,\mathrm{d}w \qquad \rightarrow \textit{Fonction de répartition} \text{ de } X$$

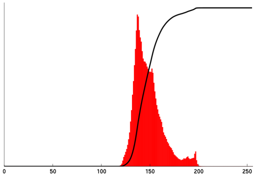
- En pratique : histogrammes discrets. Impossible d'obtenir un histogramme plat mais on suit la même idée $T(x_k) = \dfrac{(L-1)}{n} \displaystyle\sum_{j=0}^{k} n_j$

Égalisation d'histogramme

Résultats :

avant

après



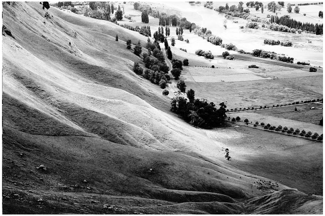
histogramme (rouge) et histogramme cumulé (noir)

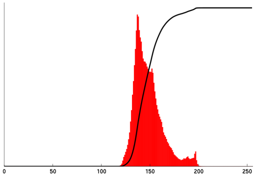
Égalisation d'histogramme

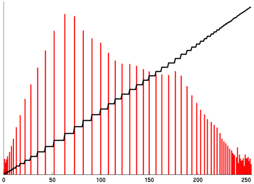
Résultats :

avant

après





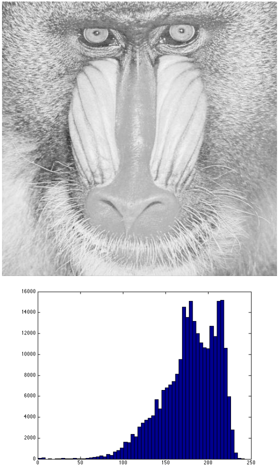


histogramme (rouge) et histogramme cumulé (noir)

Égalisation d'histogramme

Résultats :

avant

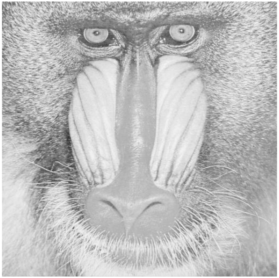


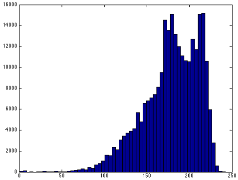
après

Égalisation d'histogramme

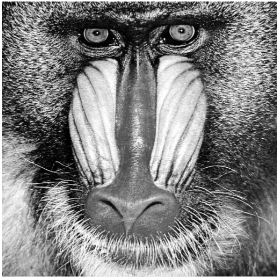
Résultats :

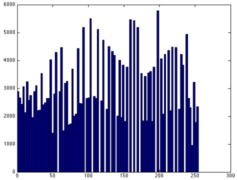
avant





après





Premières transformations

Réduction du nombre de couleurs (quantification)

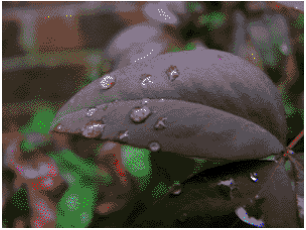
But

Compression ou adaptation au système d'affichage

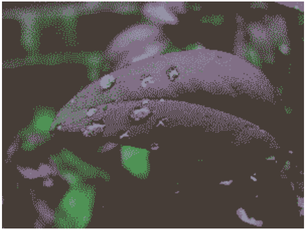
24 bits par pixel (bpp)
16 millions de couleurs

Réduction du nombre de couleurs (quantification)

8 bpp (256 couleurs)



4 bpp (16 couleurs)



2 bpp (4 couleurs)

1 bpp (2 couleurs)

Réduction du nombre de couleurs (quantification)

Réduction n.d.g. $\rightarrow$ images binaires

- Utilité de la binarisation : imprimantes, fax

Réduction du nombre de couleurs (quantification)

Réduction n.d.g. → images binaires

- Utilité de la binarisation : imprimantes, fax
- Binarisation par seuillage

Figure 1: Original image

Figure 2: Average dithering

Réduction du nombre de couleurs (quantification)

Réduction n.d.g. → images binaires

- Binarisation par tramage

Principe : utiliser un ensemble (cluster) de pixels pour représenter l'intensité
→exploite l'intégration spatiale de l'oeil
→utilisé par les imprimantes

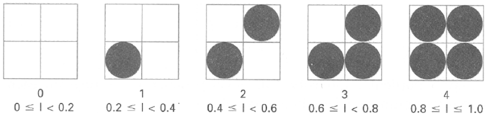


Figure 3: Original image

Figure 4: Ordered dithering

Figure 5: Random dithering

Réduction du nombre de couleurs (quantification)

Réduction n.d.g. $\rightarrow$ images binaires

- Binarisation par propagation de l'erreur (Floyd-Steinberg)

Principe : diffusion de l'erreur de quantification d'un pixel à ses voisins

$$I_{ndg} = I_b + \text{erreur} \Rightarrow \text{erreur} = I_{ndg}(i,j) - I_b(i,j)$$

	(i,j) erreur	$\frac{7}{16}$ erreur
$\frac{3}{16}$ erreur	$\frac{5}{16}$ erreur	$\frac{1}{16}$ erreur

Figure 6: Original image

Figure 7: Floyd-Steinberg dithering

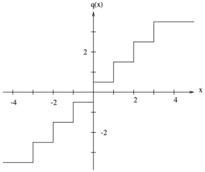
plus

DEMO MATLAB

Réduction du nombre de couleurs (quantification)

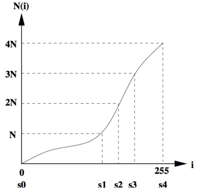
Réduction n.d.g. → n.d.g.

- Quantification scalaire uniforme :



Wikipedia

- Quantification scalaire non uniforme : on peut souhaiter que chaque classe soit utilisée par le même nombre de pixels

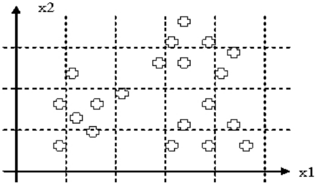


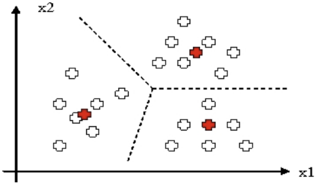
en utilisant l'histogramme cumulé

Réduction du nombre de couleurs (quantification)

Réduction RGB $\rightarrow$ RGB

- On peut appliquer une quantification scalaire sur chaque composante (R, G et B) séparément $\longrightarrow$ pas optimal
- Optimal au sens de l'erreur quadratique moyenne : appliquer les k-means dans $\mathbb{R}^3$. On cherche des représentants de chaque classe directement dans RGB.





demo

Transformations géométriques

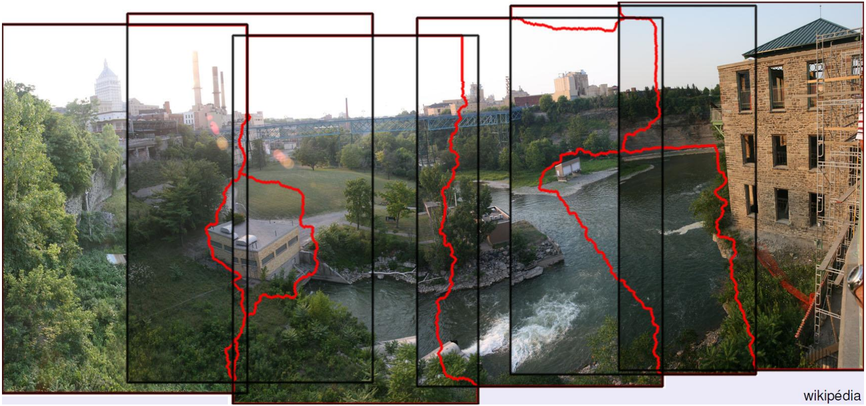
But

Déformer une image (sans modifier ses valeurs)

- Retouche d'images
- Création de panoramas
- Recalage d'images (déformer une image I pour qu'elle se "superpose" sur une image J)

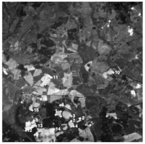
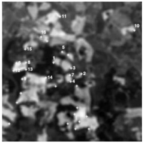
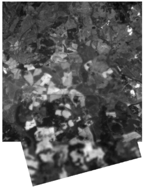
Exemples d'applications 1/3

- Création de panoramas

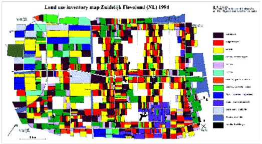


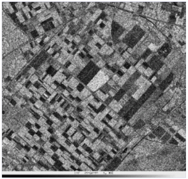
Exemples d'applications 2/3

- Images satellitaires

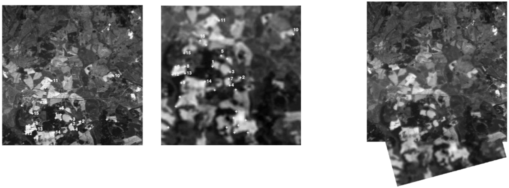
- Télédétection : Carte de végétation et image satellitaire



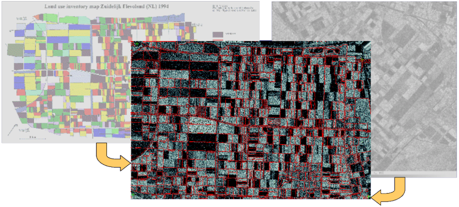


Exemples d’applications 2/3

- Images satellitaires



- Télédétection : Carte de végétation et image satellitaire

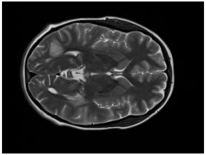
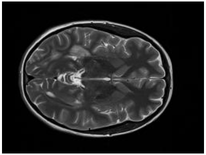
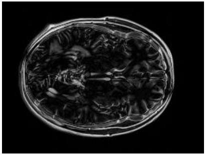


Exemples d'applications 3/3

- Recalage d'images (indispensable en traitement des images médicales)

Exemples d'applications 3/3

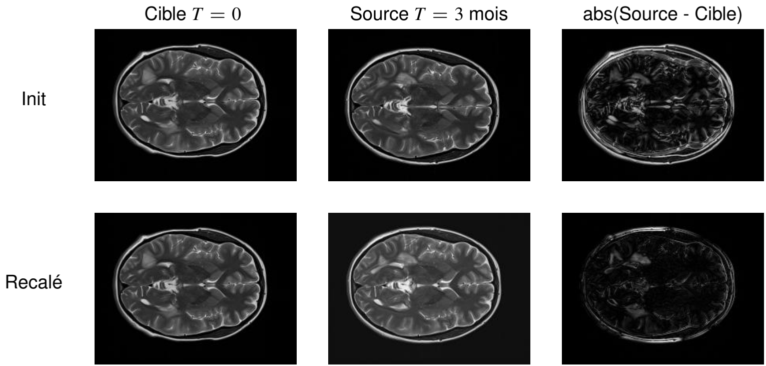
- Recalage d'images (indispensable en traitement des images médicales)

	Cible $T = 0$	Source $T = 3$ mois	abs(Source - Cible)
Init			

évolution de lésions (images IRM-T2 d'un patient atteint de sclérose en plaques à quelques mois d'intervalle)

Exemples d'applications 3/3

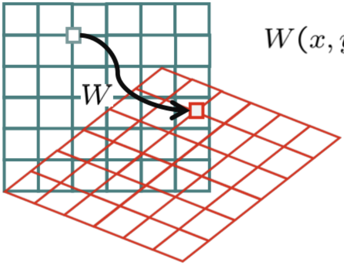
- Recalage d'images (indispensable en traitement des images médicales)



évolution de lésions (images IRM-T2 d'un patient atteint de sclérose en plaques à quelques mois d'intervalle)

Transformations géométriques

- Transformation affine (paramétrique de faible degré)



$$W(x,y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformations géométriques

- Transformation rigide (rotation, translation)

Transformations géométriques

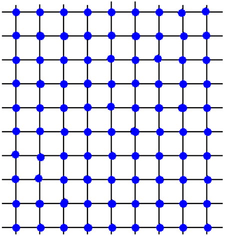
- Transformation affine (paramétrique de faible degré)

Transformations géométriques

- Transformation non-linéaire (paramétrique de degré élevé)

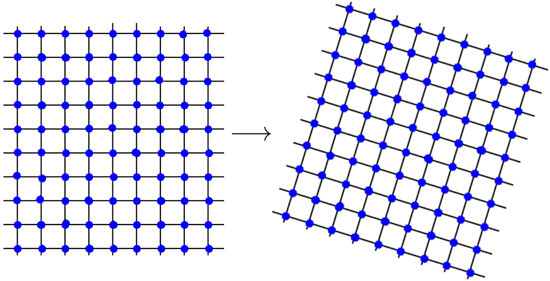
Transformations géométriques

- $W(x,y)$ n'est généralement pas à coordonnées entières



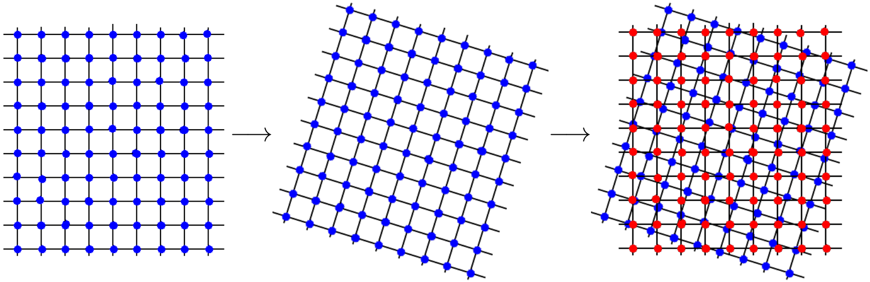
Transformations géométriques

- $W(x, y)$ n'est généralement pas à coordonnées entières



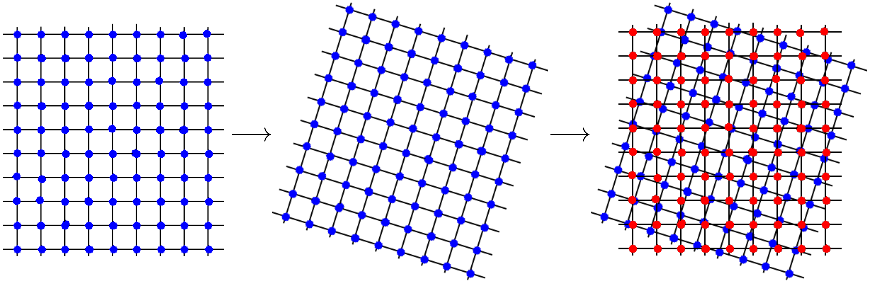
Transformations géométriques

- $W(x,y)$ n'est généralement pas à coordonnées entières



Transformations géométriques

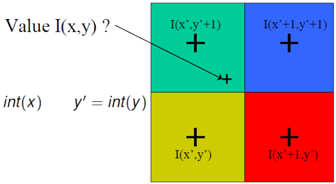
- $W(x, y)$ n'est généralement pas à coordonnées entières



Interpolation d'intensités

Attribuer une intensité *raisonnable* à une position physique tombant entre des positions où l'intensité est connue (donc une position à coordonnées non entières)

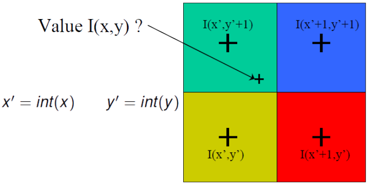
Interpolation d'intensités



$x' = int(x) \qquad y' = int(y)$

Deux approches "basiques"

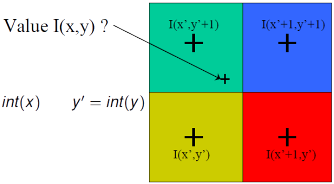
Interpolation d'intensités



Deux approches "basiques"

- interpolation au plus proche voisins : $I(x, y) = I(round(x), round(y))$

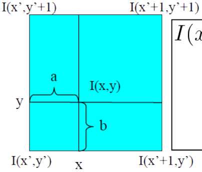
Interpolation d'intensités



$x' = int(x) \qquad y' = int(y)$

Deux approches "basiques"

- interpolation au plus proche voisins : $I(x,y) = I(round(x), round(y))$
- interpolation bi-linéaire



$$\boxed{\begin{aligned} I(x,y) &= a(1-b)I(x'+1,y') \\ &\quad +(1-a)(1-b)I(x',y') \\ &\quad +abI(x'+1,y'+1) \\ &\quad +b(1-a)I(x',y'+1) \end{aligned}}$$

Demo Gimp

Interpolation d'intensités

Approches plus sophistiquées

- On cherche une fonction "simple" qui passe par les différents points connus, et on regarde la valeur de cette fonction au point qui nous intéresse

Interpolation d'intensités

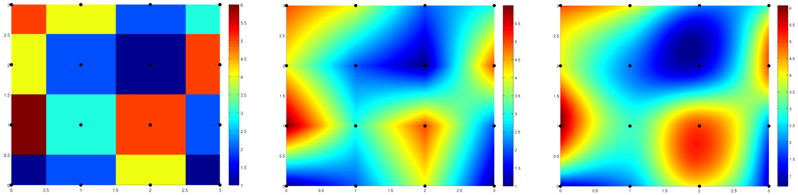
Approches plus sophistiquées

- On cherche une fonction "simple" qui passe par les différents points connus, et on regarde la valeur de cette fonction au point qui nous intéresse
- fonctions possibles :
  - Bicubique (spline) : $I(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j$

Interpolation d'intensités

Approches plus sophistiquées

- On cherche une fonction "simple" qui passe par les différents points connus, et on regarde la valeur de cette fonction au point qui nous intéresse
- fonctions possibles :
  - Bicubique (spline) : $I(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$



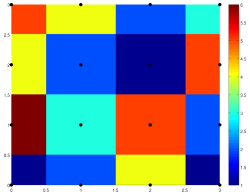
Plus proche voisin — Bilinéaire — Bicubique

http://en.wikipedia.org/wiki/Bicubic_interpolation

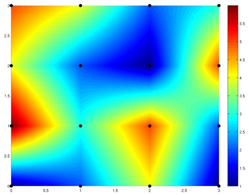
Interpolation d'intensités

Approches plus sophistiquées

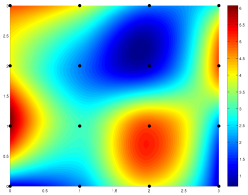
- On cherche une fonction "simple" qui passe par les différents points connus, et on regarde la valeur de cette fonction au point qui nous intéresse
- fonctions possibles :
  - Bicubique (spline) : $I(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$



Plus proche voisin



Bilinéaire



Bicubique

http://en.wikipedia.org/wiki/Bicubic_interpolation

  - basée sur $\operatorname{sinc}(x) = \dfrac{\sin(\pi x)}{\pi x}$ $\rightarrow$ Lanczos3

Transformations locales

Transformation locales

Remplacer l'intensité en un pixel par une fonction de son intensité et de celle de ses voisins

$$\forall(x,y), \quad J(x,y) = f[I(x',y'),(x',y') \in V(x,y)]$$

Transformations locales

Transformation locales

Remplacer l'intensité en un pixel par une fonction de son intensité et de celle de ses voisins

$$\forall (x,y), \quad J(x,y) = f[I(x',y'),(x',y') \in V(x,y)]$$

- **Filtrage linéaire** : la fonction est une combinaison linéaire avec des poids dépendants de la position du voisin relativement au pixel central

$$J = H * I, \quad J(x,y) = \sum_{(u,v)\in V(x,y)} H(x-u,y-v)\, I(u,v)$$

Transformations locales

Transformation locales

Remplacer l'intensité en un pixel par une fonction de son intensité et de celle de ses voisins

$$\forall(x,y),\quad J(x,y) = f[I(x',y'),(x',y') \in V(x,y)]$$

- **Filtrage linéaire** : la fonction est une combinaison linéaire avec des poids dépendants de la position du voisin relativement au pixel central

$$J = H * I,\quad J(x,y) = \sum_{(u,v)\in V(x,y)} H(x-u,y-v)\, I(u,v)$$

  - $\rightarrow$ **Convolution discrète** par le filtre de noyau H et de support fini V de taille $(2k+1)\times(2k+1)$, souvent normalisé : $\sum_{u=-k}^{k}\sum_{v=-k}^{k} |H(u,v)| = 1$

Aparté : convolution

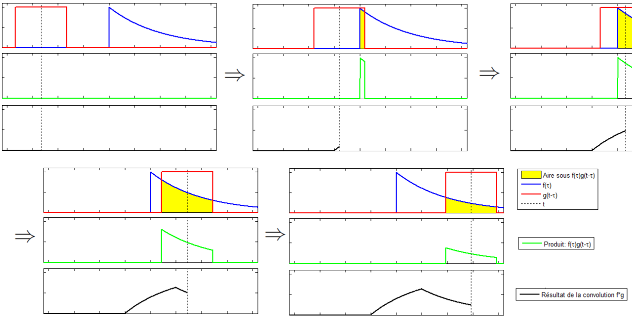
Produit de convolution de 2 fonctions

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x-t) \cdot g(t) \mathrm{d}t = \int_{-\infty}^{+\infty} f(t) \cdot g(x-t) \mathrm{d}t,$$

Aparté : convolution

Produit de convolution de 2 fonctions

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x-t) \cdot g(t) \mathrm{d}t = \int_{-\infty}^{+\infty} f(t) \cdot g(x-t) \mathrm{d}t,$$



DEMO MATLAB

Transformations locales

- **Filtre moyenneur**, on veut remplacer chaque valeur par la moyenne des valeurs des pixels autour du pixel initial

$$H(u, v) =$$

Transformations locales

- **Filtre moyenneur**, on veut remplacer chaque valeur par la moyenne des valeurs des pixels autour du pixel initial

$$H(u, v) = \frac{1}{(2\,k + 1)^2}$$

Transformations locales

- **Filtre moyenneur**, on veut remplacer chaque valeur par la moyenne des valeurs des pixels autour du pixel initial

$$H(u,v) = \frac{1}{(2\,k+1)^2} \qquad \text{exemple :} \quad \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

Transformations locales

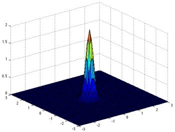
- **Filtre moyenneur**, on veut remplacer chaque valeur par la moyenne des valeurs des pixels autour du pixel initial

$$H(u,v) = \frac{1}{(2k+1)^2} \qquad \text{exemple :} \quad \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

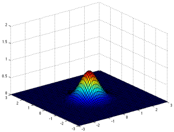
$$k = 5$$

Transformations locales

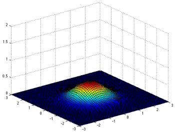
- **Filtre gaussien**
  - Gaussienne 2D : $G_\sigma(x,y) = \dfrac{1}{2\pi\sigma^2}\,\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}$ et on a $\displaystyle\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} G_\sigma(x,y)\,dx\,dy = 1$



$\sigma = 0.2$



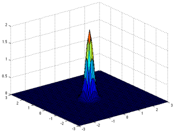
$\sigma = 0.5$



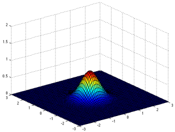
$\sigma = 0.8$

Transformations locales

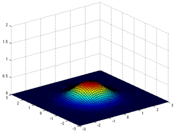
- **Filtre gaussien**
  - Gaussienne 2D : $G_\sigma(x,y) = \dfrac{1}{2\pi\sigma^2}\,\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}$ et on a $\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} G_\sigma(x,y)\,dx\,dy = 1$



$\sigma = 0.2$



$\sigma = 0.5$



$\sigma = 0.8$

  - Filtre discret à support borné (dont la taille dépend en général de l'écart type de la gaussienne)

$$u = -k,\ldots,k\ ,\quad v = -k,\ldots,k\ ,\quad H(u,v) = G_\sigma(u,v)$$

  - Un outil **primordial**

Aparté : convolution avec une gaussienne

- $\rightarrow$ généralise la notion de "moyenne glissante"

$$\boxed{\text{DEMO MATLAB}}$$

Aparté : convolution avec une gaussienne

- $\rightarrow$ généralise la notion de "moyenne glissante"

$$\boxed{\text{DEMO MATLAB}}$$

- Convolution avec une Gaussienne 2D

$$I_\sigma(x, y) = I(x, y) * G_\sigma(x, y) = \int_\Omega G_\sigma(x - a, y - b) \cdot I(a, b)\, \mathrm{d}a\, \mathrm{d}b$$

Aparté : convolution avec une gaussienne

- $\rightarrow$ généralise la notion de "moyenne glissante"

DEMO MATLAB

- Convolution avec une Gaussienne 2D

$$I_\sigma(x,y) = I(x,y) * G_\sigma(x,y) = \int_\Omega G_\sigma(x-a, y-b) \cdot I(a,b)\,\mathrm{d}a\,\mathrm{d}b$$

- Exemple

I

Aparté : convolution avec une gaussienne

- $\rightarrow$ généralise la notion de "moyenne glissante"

DEMO MATLAB

- Convolution avec une Gaussienne 2D

$$I_\sigma(x,y) = I(x,y) * G_\sigma(x,y) = \int_\Omega G_\sigma(x-a, y-b) \cdot I(a,b)\,\mathrm{d}a\,\mathrm{d}b$$

- Exemple

I

I_{10}

Aparté : convolution avec une gaussienne

- $\rightarrow$ généralise la notion de "moyenne glissante"

$\boxed{\text{DEMO MATLAB}}$

- Convolution avec une Gaussienne 2D

$$I_\sigma(x,y) = I(x,y) * G_\sigma(x,y) = \int_\Omega G_\sigma(x-a, y-b) \cdot I(a,b)\,\mathrm{d}a\,\mathrm{d}b$$

- Exemple

I

I_{10}

I_{25}

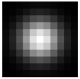
Transformations locales

- **Filtre moyenneur**

$k = 5$

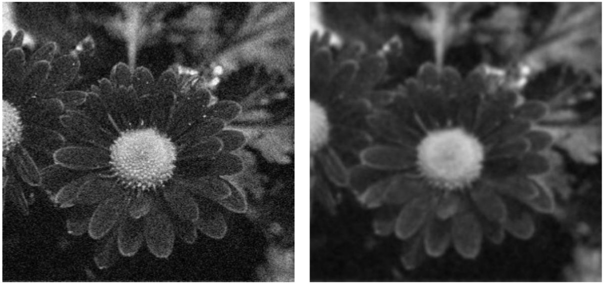
- **Filtre gaussien**

$k = 5, \sigma = 2$

Lissage, "débruitage"

- Filtres moyenneurs et gaussiens *lissent* l'image
  - peuvent enlever du bruit (ponctuel, "grains")
  - mais au prix d'une perte de détails $\rightarrow$ flou



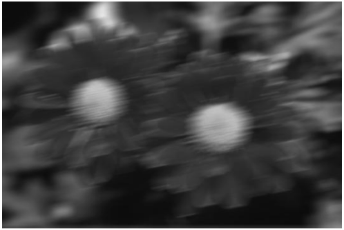
Flou de bougé

- lissage anisotrope
  - Exemple de noyau 
  - Effet de la convolution par ce noyau

Flou de bougé

- lissage anisotrope
  - Exemple de noyau 
  - Effet de la convolution par ce noyau

$\longrightarrow$



Filtrage non linéaire

Filtre médian

On remplace la valeur du pixel par la **médiane** des valeurs voisines

Filtrage non linéaire

Filtre médian

On remplace la valeur du pixel par la **médiane** des valeurs voisines

- Non linéaire : $\text{médiane}\{x+y\} \neq \text{médiane}\{x\} + \text{médiane}\{y\}$

Filtrage non linéaire

Filtre médian

On remplace la valeur du pixel par la **médiane** des valeurs voisines

- Non linéaire : $\text{médiane}\{x+y\} \neq \text{médiane}\{x\} + \text{médiane}\{y\}$
  - $x = (1,1,3),\ y = (1,2,0) \ \rightarrow \left\{ \begin{array}{r} \text{médiane}\{x\} = \\ \text{médiane}\{y\} = \\ \text{médiane}\{x+y\} = \end{array} \right.$

Filtrage non linéaire

Filtre médian

On remplace la valeur du pixel par la **médiane** des valeurs voisines

- Non linéaire : $\text{médiane}\{x+y\} \neq \text{médiane}\{x\} + \text{médiane}\{y\}$
  - $x=(1,1,3),\ y=(1,2,0) \ \rightarrow \begin{cases} \text{médiane}\{x\} = 1 \\ \text{médiane}\{y\} = 1 \\ \text{médiane}\{x+y\} = 3 \end{cases}$

Filtrage non linéaire

Filtre médian

On remplace la valeur du pixel par la **médiane** des valeurs voisines

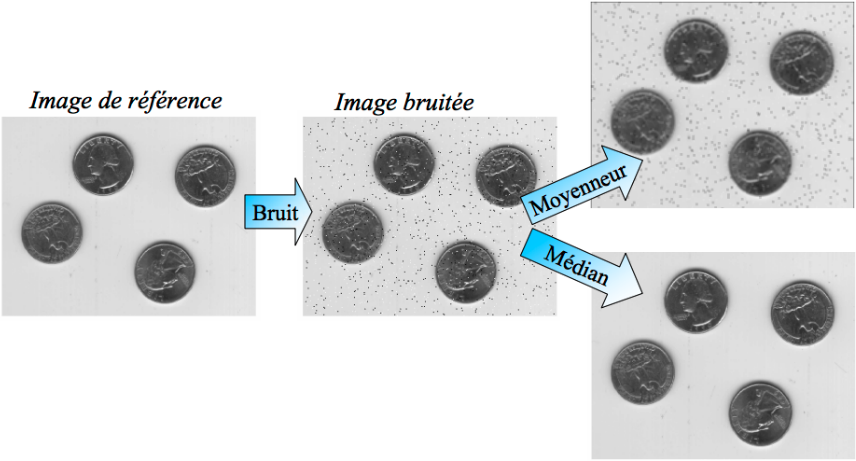
- Non linéaire : $\text{médiane}\{x+y\} \neq \text{médiane}\{x\} + \text{médiane}\{y\}$
  - $x = (1,1,3),\ y = (1,2,0) \ \rightarrow \begin{cases} \text{médiane}\{x\} = 1 \\ \text{médiane}\{y\} = 1 \\ \text{médiane}\{x+y\} = 3 \end{cases}$
- Application du filtre médian sur

91	55	90
77	68	95
115	151	210

  - On trie les valeurs voisines (centre inclus) :
    $(55, 68, 77, 90, 91, 95, 115, 151, 210) \ \rightarrow$ valeur médiane = 91

Filtrage non linéaire

- Filtre médian : approprié pour réduire le bruit impulsionnel



Filtrage non linéaire

- Filtre médian : effet de la taille k du noyau considéré

image bruitée

Filtre médian avec $k = 9$

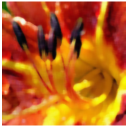
$k = 3$

$k = 5$

$k = 7$

$k = 9$

$k = 13$



$k = 17$