

SDD – In-depth : Imagery

Thomas Oberlin

ISAE-SUPAERO, Département d'ingénierie des systèmes complexes (DISC)

thomas.oberlin@isae-supraero.fr

1 / 56

Plan du cours

1. Introduction à l'imagerie

Introduction

L'image numérique

De l'analogique au numérique

Transformations de la dynamique

Exemples

Réduction du nombre de couleurs

Transformations géométriques

3 / 56

Objectifs du module

Les images numériques

- ▶ Vision humaine et couleurs
- ▶ Images numériques (échantillonnage, quantification, saturation, dynamique)
- ▶ Notions sur les traitements classiques (compression, restauration)

La vision artificielle

- ▶ Révision CNN, transfer learning
- ▶ Détection et segmentation
- ▶ Modèles génératifs

Organisation

- ▶ 5 séances de 3 heures de cours/BE
- ▶ Évaluation : Une note de BE (séances 4 et 5)

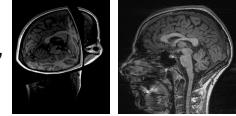
2 / 56

L'image : un signal en dimension > 1

- ▶ Notion de signal : observation de phénomène physique
- ▶ Quantités dépendantes du temps, de l'espace ou de la fréquence
- ▶ Modélisation sous forme de fonction d'une ou plusieurs variables

▶ 1D : variable → **le temps** t (exemple : le son) 

▶ 2D : les images, variables → **l'espace** (x, y) 

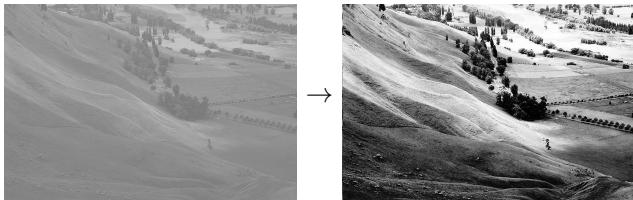
▶ 3D : images 3D (x, y, z) ,  vidéos $((x, y), t)$



▶ 4D : volumes 3D évoluant dans le temps

4 / 56

Traitement d'image – exemples



[Wikipedia]

5 / 56

Traitement d'image – exemples



[Criminisi et al., CVPR, 2003]

7 / 56

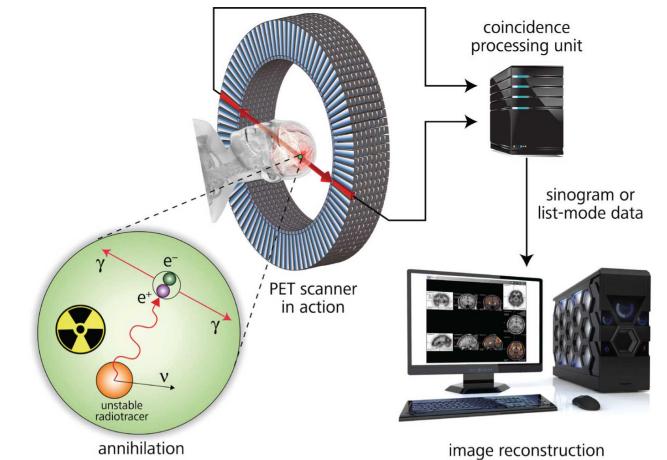
Traitement d'image – exemples



[Dorothea Lange, 1936]

6 / 56

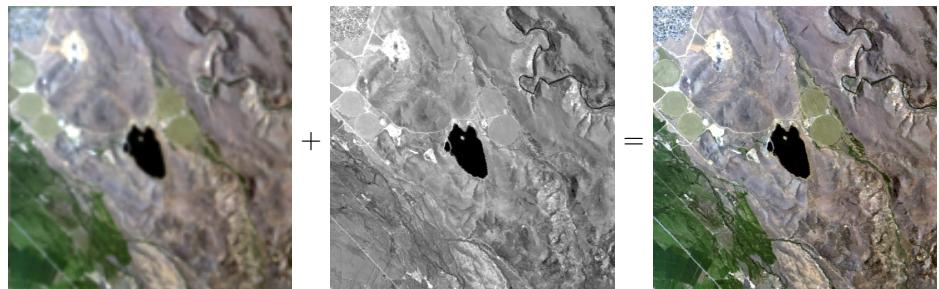
Traitement d'image – exemples



Tomographie par émission de positrons (PET)

8 / 56

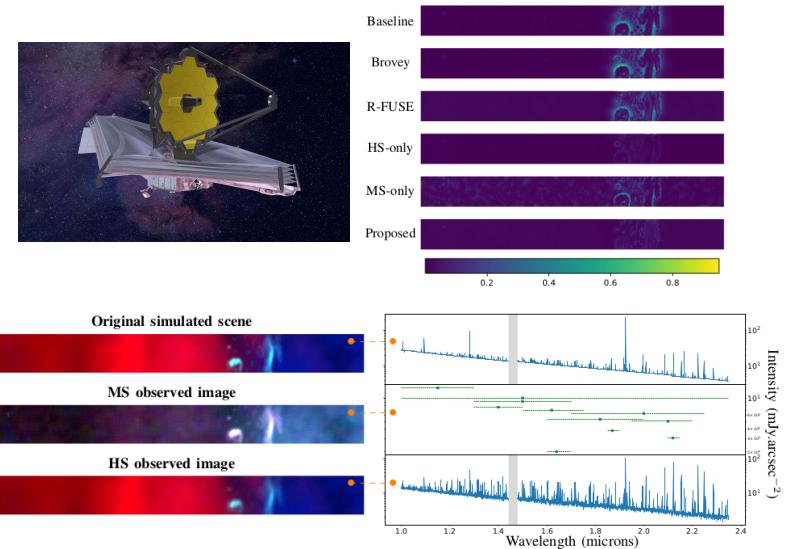
Traitement d'image – exemples



[Fusion d'images XS et PAN pour l'observation de la Terre]

9 / 56

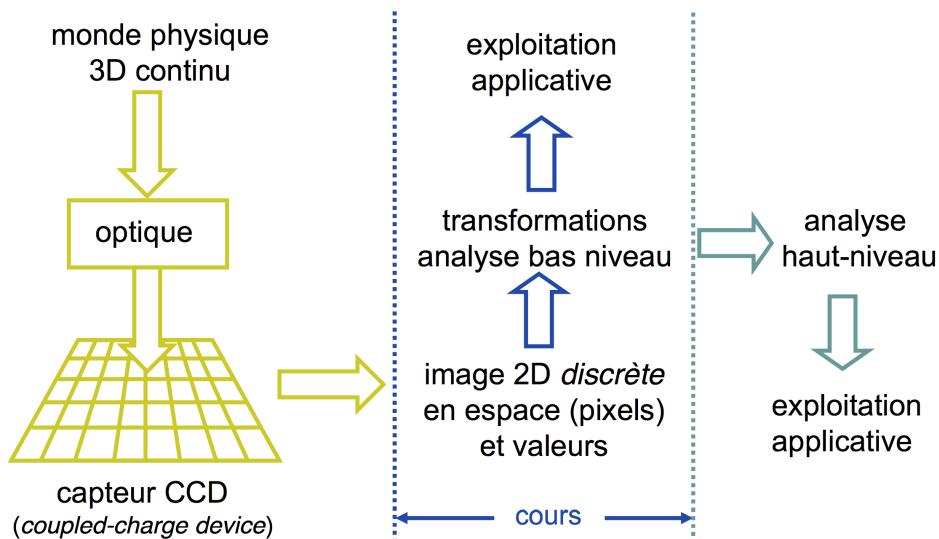
Traitement d'image – exemples



[Fusion imagerie / spectroscopie infrarouge pour le JWST]

10 / 56

Traitement d'image et vision artificielle



11 / 56

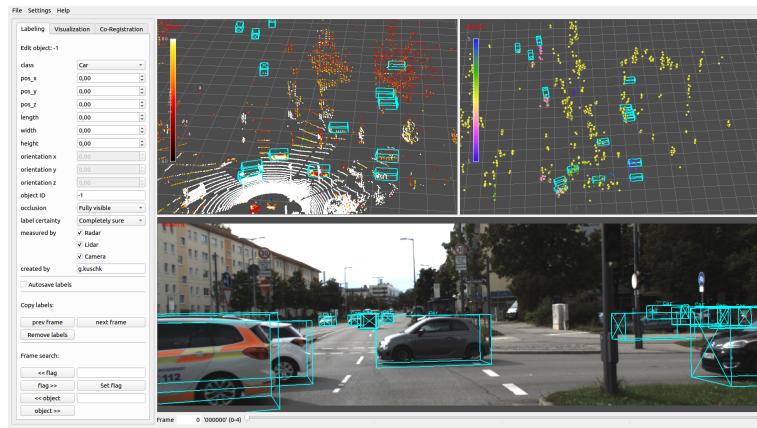
Computer vision – exemples



www.mobileye.com

12 / 56

Computer vision – exemples

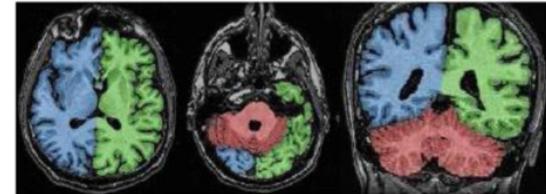


13 / 56

Computer vision – exemples

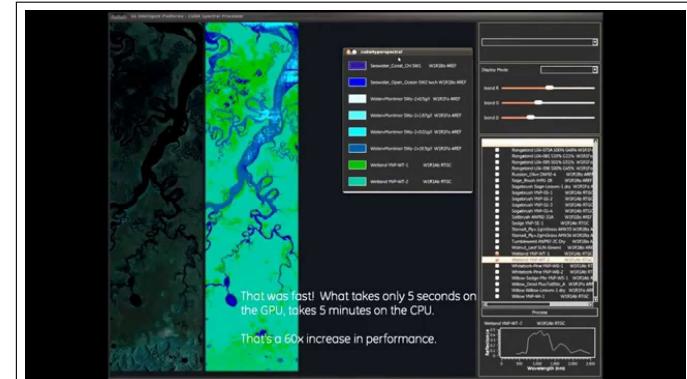


Hémisphère gauche,
hémisphère droit, cervelet



15 / 56

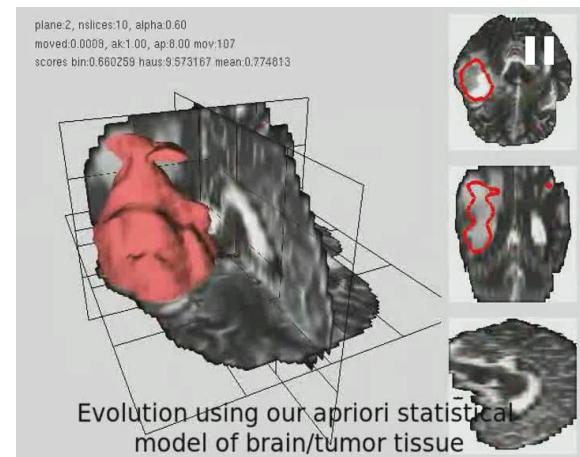
Computer vision – exemples



[GPGPU Hyperspectral Imaging, GE Intelligent Platforms]

14 / 56

Computer vision – exemples

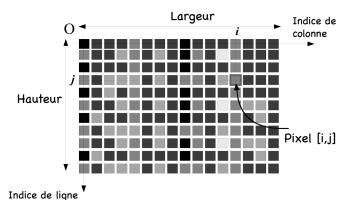


2008: Cobzas, Birkbeck, Schmidt, Jagersand, Murtha A.
3D Variational Brain Tumor Segmentation using a High Dimensional Feature Set

16 / 56

L'image numérique

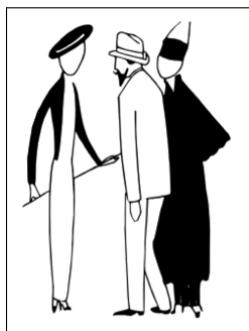
- ▶ image = signal bidimensionnel
- ▶ image **analogique** : image formée sur la rétine de l'oeil, image obtenue par la photo argentique classique
- ▶ image **numérique** : signal numérique composé d'unités élémentaires (pixels) qui représentent chacun une portion de l'image. Caractérisée par :
 - ▶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



17 / 56

Images binaires (noir ou blanc)

Cas où $I = \{0, 1\}$

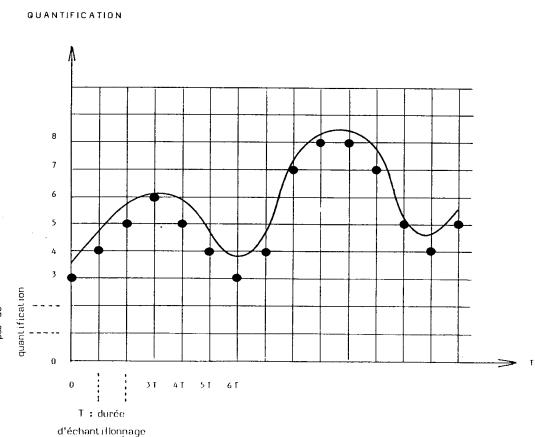


```
%-----\begin{frame}\frametitle{Traitement des images : cas particulier du traitement du\begin{itemize}\item notion de signal : observation de phénomène.\item quantités dépendantes du temps, de l'espace ou\item modélisation sous forme de fonction d'une ou p\begin{itemize}\item\begin{itemize}\item 1D : variable $→$\item 2D : les images, variables $→$\item 3D : images 3D $(x,y,z)$, $→$\item 4D : volumes 3D évoluant dans\end{itemize}\end{itemize}\end{itemize}\end{frame}%-----
```

19 / 56

Échantillonnage et quantification

- ▶ Une "image continue" = une fonction à deux variables réelles $\mathbb{R}^2 \rightarrow \mathbb{R}^{+d}$
- ▶ Image numérique : $f : \llbracket 1, M \rrbracket \times \llbracket 1, N \rrbracket \rightarrow I$, où MN est le nombre de pixels, et I est un ensemble discret de valeurs.

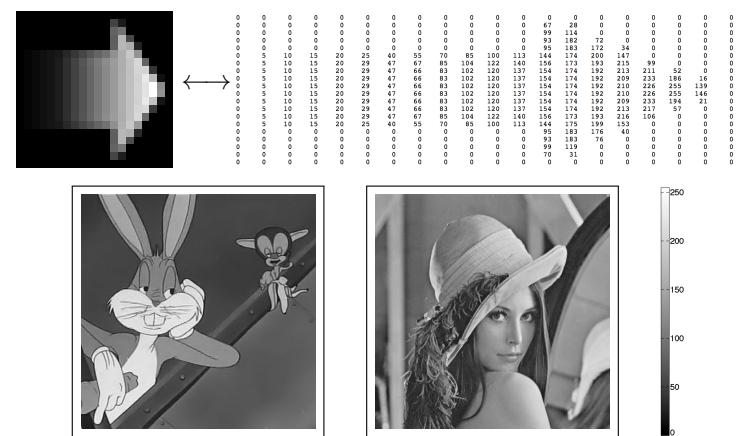


Rappel en 1D

18 / 56

Images en niveau de gris

- ▶ cas où $I = \{0, \dots, P - 1\}$ (souvent $P = 2^8 = 256$: codage sur 8 bits)
- ▶ 0=noir, 255=blanc, les autres=teintes de gris



20 / 56

Images couleurs RVB

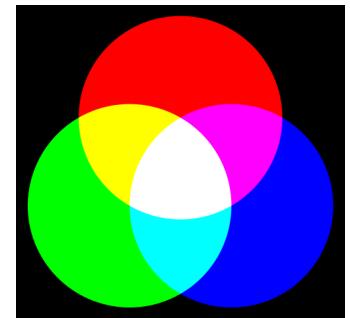
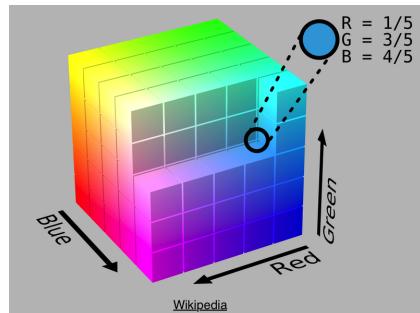
$$I = \underbrace{\{0, \dots, P-1\}}_{\text{Rouge}} \times \underbrace{\{0, \dots, P-1\}}_{\text{Vert}} \times \underbrace{\{0, \dots, P-1\}}_{\text{Bleu}}$$



21 / 56

Espace RVB

- ▶ basé sur la synthèse additive des couleurs → mélange de 3 composantes
- ▶ Souvent, chaque couleur sur 8bits donc $I = \{0, \dots, 255\}^3$
- ▶ Utilisé par les écrans d'ordinateurs



22 / 56

Images couleurs

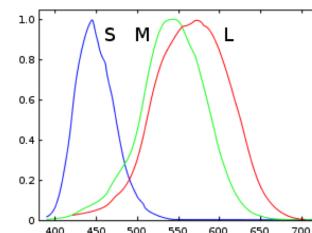
$$\begin{array}{|c|c|c|} \hline & \text{Noir} & \text{Rouge} & \text{Orange} \\ \hline & \text{Vert} & \text{Gris} & \text{Jaune} \\ \hline & \text{Bleu} & & \text{Blanc} \\ \hline \end{array} \longleftrightarrow \begin{bmatrix} (0,0,0) & (128,0,0) & (255,0,0) \\ (0,128,0) & (128,128,128) & (255,128,0) \\ (0,255,0) & (128,255,0) & (255,255,255) \end{bmatrix}$$

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

23 / 56

Images couleurs

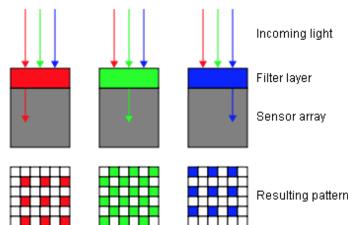
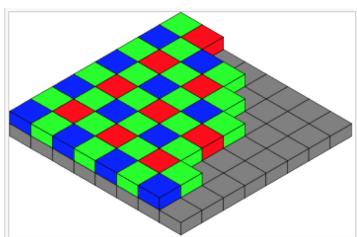
- ▶ **Onde lumineuse** : superposition d'une infinité d'ondes pures
- ▶ **Système visuel**
 - ▶ tout signal lumineux est bien approché, *visuellement*, par une combinaison linéaire de trois primaires
 - ▶ Constat empirique (mélange en peinture et imprimerie)
 - ▶ Explication anatomique : trois types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



24 / 56

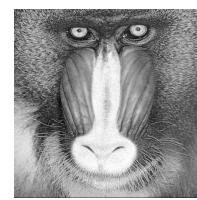
Capteur CCD (coupled-charge device)

- ▶ 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

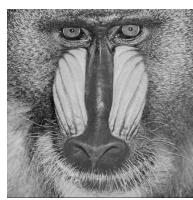


25 / 56

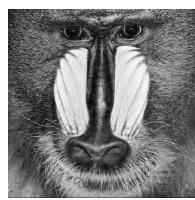
Espaces colorimétriques: RGB



R



G



B

27 / 56

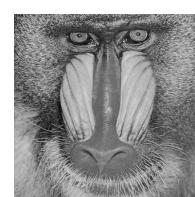
Espaces colorimétriques

- ▶ Tri-chromie
 - ▶ Red-Green-Blue : RGB
 - ▶ Cyan-Yellow-Magenta : CYM (synthèse soustractive)
- ▶ Espaces luminance-chrominance : une composante de luminance plus 2 de chrominance. Ex: xyY, YUV (linéaires), CIE Lab (non linéaire)
- ▶ 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
- ▶ Hue(teinte)-Saturation-Value : HSV (TSV en français). Non-linéaire, représentation circulaire de la teinte → représentation conique de l'espace des couleurs.

26 / 56

Espaces colorimétriques: Yuv

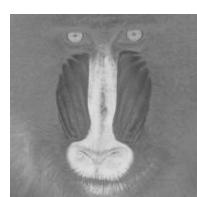
$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0, 299 & 0, 587 & 0, 114 \\ -0, 14713 & -0, 28886 & 0, 436 \\ 0, 615 & -0, 51498 & -0, 10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1, 13983 \\ 1 & -0, 39465 & -0, 58060 \\ 1 & 2, 03211 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$



Y



28 / 56



V

29 / 56

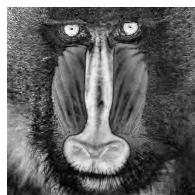
Espaces colorimétriques: HSV



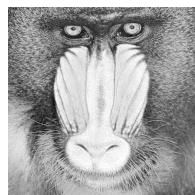
(Transformation non linéaire)



H



S



V

29 / 56

Images spectrales

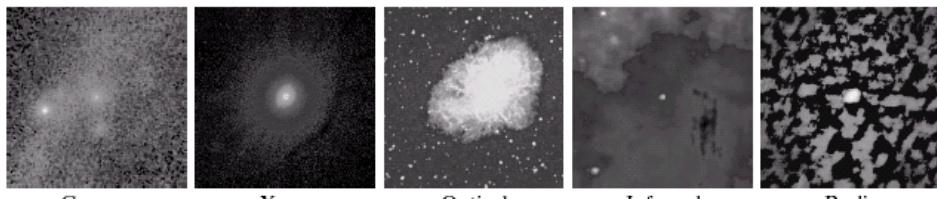


FIGURE 1.18 Images of the Crab Pulsar (in the center of images) covering the electromagnetic spectrum.
(Courtesy of NASA.)

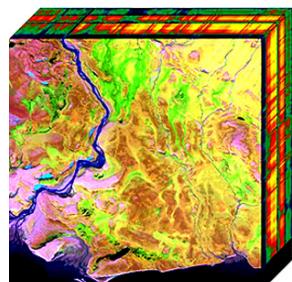
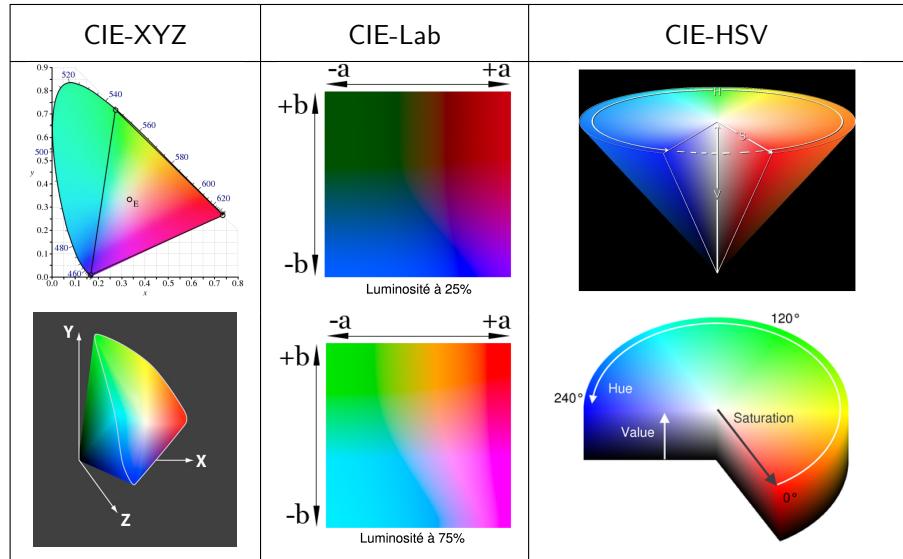


Image hyperspectrale : jusqu'à quelques centaines de longueurs d'ondes, couvrant le visible, l'infrarouge et parfois d'autres plages de longueurs d'onde.

31 / 56

Espaces colorimétriques



30 / 56

Histogramme d'une image en niveaux de gris

Définition

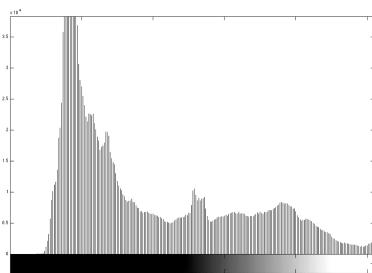
Histogramme d'une image f : fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur.

$$h_f : \{0, \dots, 255\} \rightarrow \mathbb{N}$$

$$v \mapsto \text{Card}\{(x, y) | f(x, y) = v\}$$



f



h_f

32 / 56

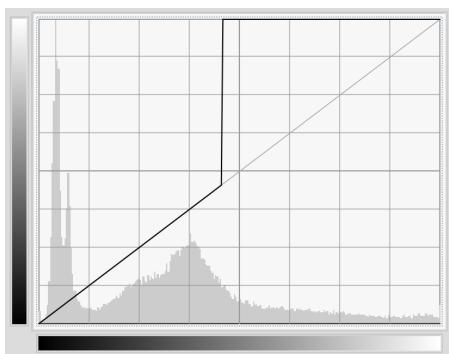
Histogramme d'une image



33 / 56

Histogramme d'une image

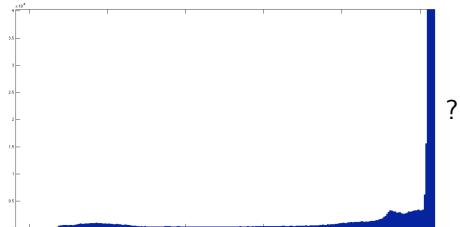
- ▶ Utile pour changer le contraste d'une image.
- ▶ On construit une image g ainsi : $g(x, y) = \phi(f(x, y))$, avec ϕ fonction croissante
- ▶ Édition manuelle, exple : [avec GIMP](#)



35 / 56

Histogramme d'une image

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



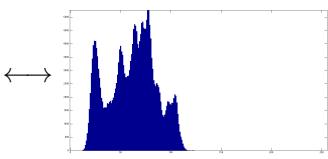
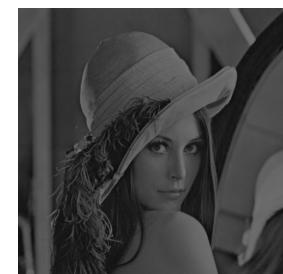
?



34 / 56

Normalisation/étirement d'histogrammes

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



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

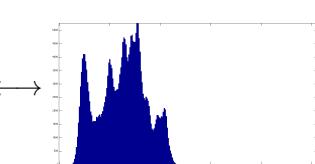
$$\min(\phi(f(x, y))) \rightarrow 0 \quad \text{et} \quad \max(\phi(f(x, y))) \rightarrow 255 \quad (\text{par exemple})$$

$$\phi(v) = 255 \frac{v - \min(I)}{\max(I) - \min(I)}$$

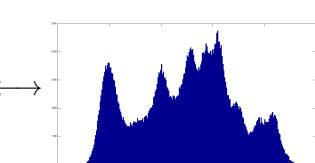
36 / 56

Normalisation/étirement d'histogrammes

- ▶ Image originale



- ▶ Image "normalisée"



37 / 56

Égalisation d'histogramme

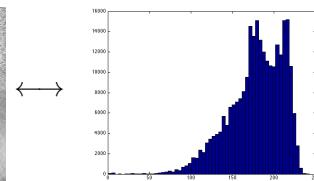
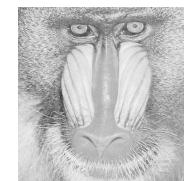
- ▶ ajustement automatique du contraste
- ▶ but : "étaler" les valeurs de l'image pour mieux répartir sur l'ensemble des valeurs possibles
- ▶ Exemple (wikipédia)



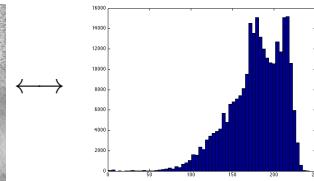
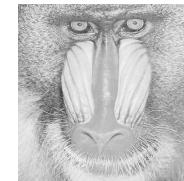
39 / 56

Normalisation/étirement d'histogrammes

- ▶ Image originale



- ▶ Image "normalisée"



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

38 / 56

Égalisation d'histogramme

Principes de l'algorithme

- ▶ image à valeurs dans $\{0, \dots, L - 1\}$
- ▶ intensités des pixels ($f[n, m]$) : réalisations d'une variable aléatoire F à valeurs dans $[0, L - 1]$ (continue)
- ▶ $p_f(f) = P_f(F = f)$: sa densité de probabilité
- ▶ Quelle transformation ϕ doit on appliquer à f pour que p_g , la densité de probabilité associée à l'image $g = \phi(f)$, soit uniforme ?
- ▶ On peut montrer qu'il faut prendre

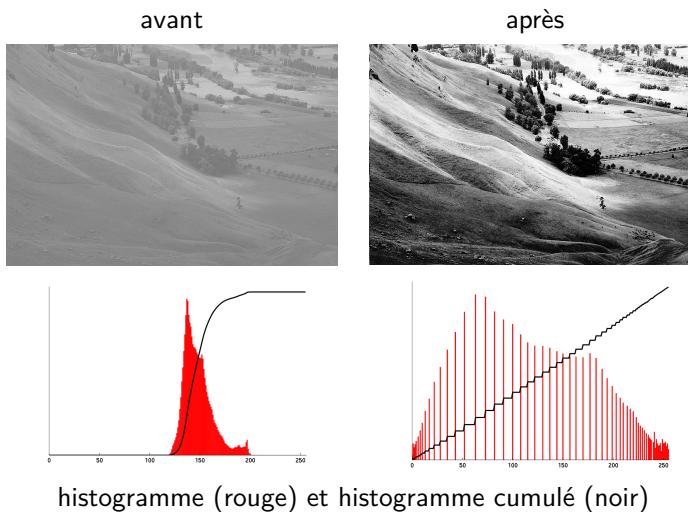
$$\phi(f) = (L - 1) \int_0^f p_f(w) dw \quad \rightarrow \text{Fonction de répartition de } F$$

- ▶ En pratique : histogrammes discrets. Impossible d'obtenir un histogramme plat mais on suit la même idée $\phi(n) = (L - 1) \sum_{k=0}^n p(f = k)$, avec $p(f = k)$ la probabilité empirique, ie la proportion de pixels de valeur k dans l'image. C'est l'**histogramme cumulé**.

40 / 56

Égalisation d'histogramme

Résultats :

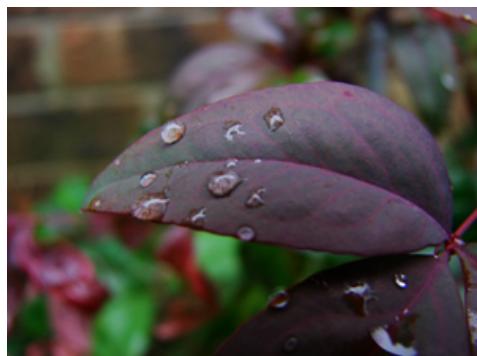


41 / 56

Réduction du nombre de couleurs (quantification)

But

Compression ou adaptation au système d'affichage

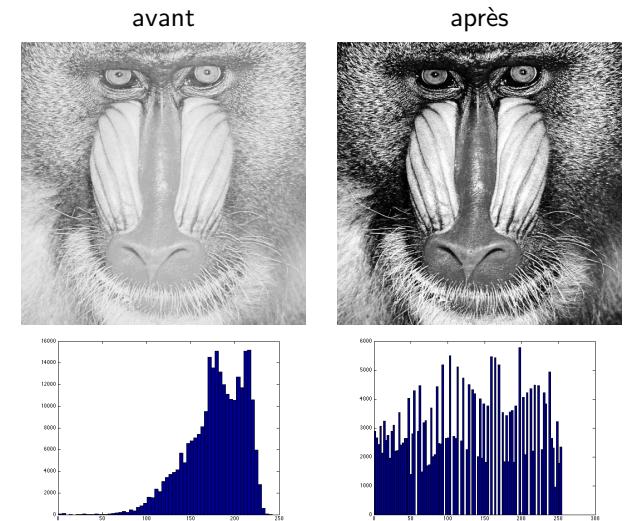


24 bits par pixel (bpp)
16 millions de couleurs

43 / 56

Égalisation d'histogramme

Résultats :



42 / 56

Réduction du nombre de couleurs (quantification)



8 bpp (256 couleurs)



4 bpp (16 couleurs)



2 bpp (4 couleurs)



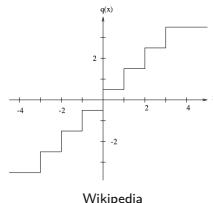
1 bpp (2 couleurs)

44 / 56

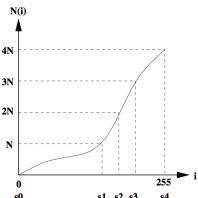
Réduction du nombre de couleurs (quantification)

Réduction n.d.g. → n.d.g.

- ▶ Quantification scalaire uniforme :



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



45 / 56

Transformations géométriques

But

Déformer une image (sans modifier ses valeurs)

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

Définition

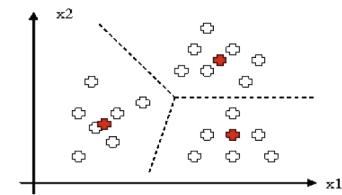
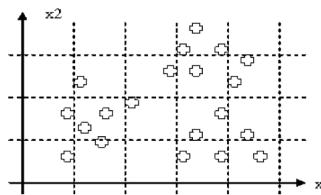
Une transformation $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

47 / 56

Réduction du nombre de couleurs (quantification)

Réduction RGB → RGB

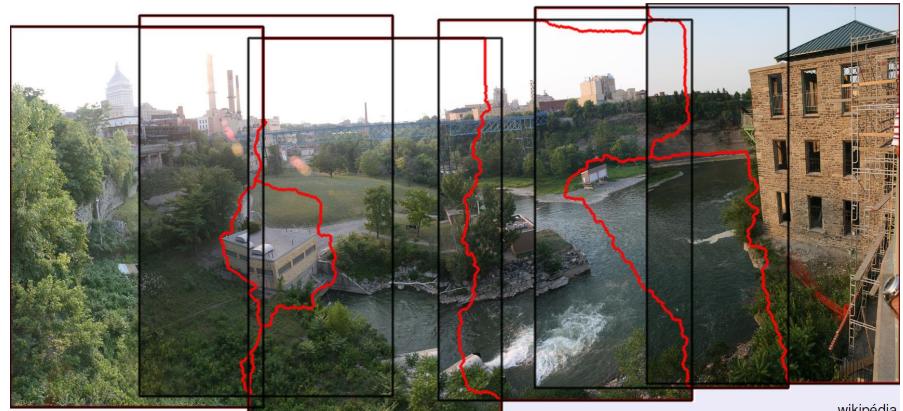
- ▶ On peut appliquer une quantification scalaire sur chaque composante (R, G et B) séparément → pas optimal
- ▶ Optimal au sens de l'erreur quadratique moyenne : appliquer les k-means dans \mathbb{R}^3 . On cherche des représentants de chaque classe directement dans RGB.



46 / 56

Exemple d'application

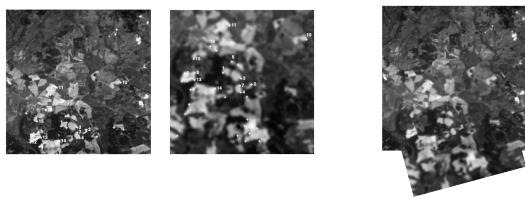
- ▶ Création de panoramas



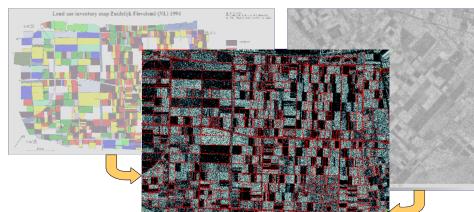
48 / 56

Exemple d'application

- ▶ Images satellites



- ▶ Télédétection : Carte de végétation et image satellitaire

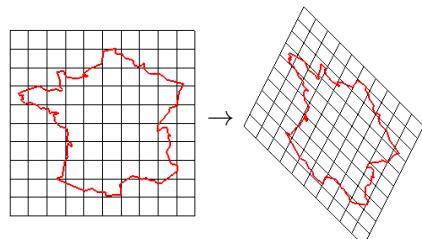


49 / 56

Transformations affines

Définition

$$T(x, y) = \underbrace{\begin{pmatrix} a & b \\ c & d \end{pmatrix}}_A \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}.$$

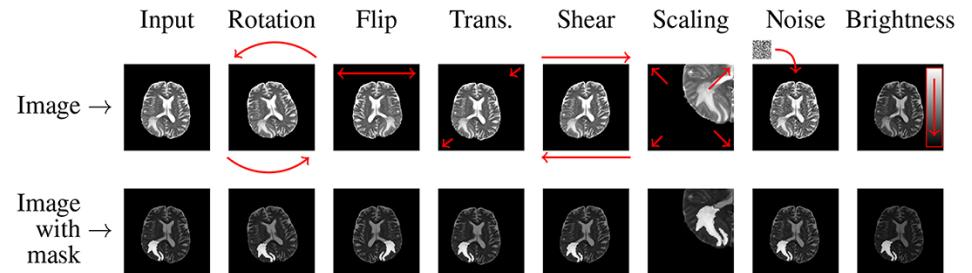


Exemple : translations ($A = I_2$), rotations, (A orthogonale), changement d'échelle (A diagonale). Conserve le parallélisme.

51 / 56

Exemple d'application

Data augmentation

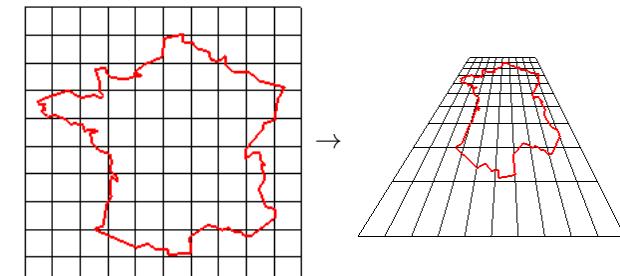


50 / 56

Transformations projectives

Caractérisation

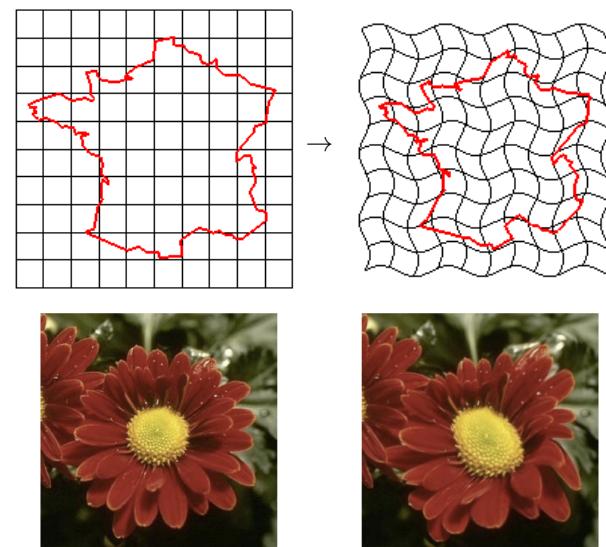
- ▶ Préserve les droites, plans, etc
- ▶ Obtenue lorsqu'on change de point de vue → très utilisée en vision par ordinateur



52 / 56

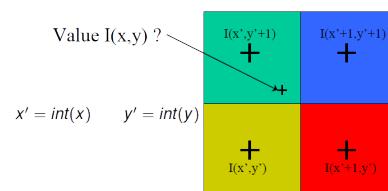
Transformations non-linéaires

Utilisées en recalage non-rigide



53 / 56

Interpolation d'images



Deux approches "basiques"

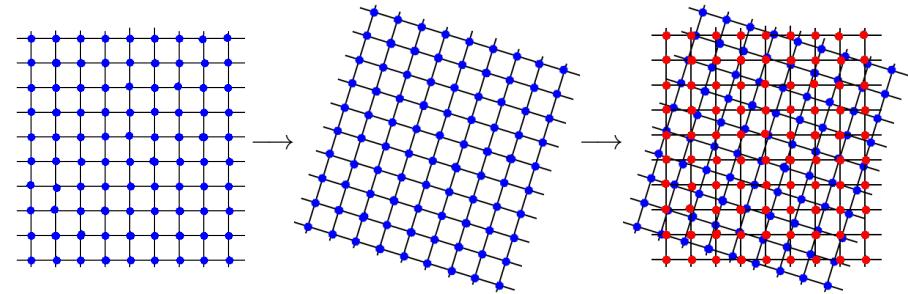
- interpolation au plus proche voisins : $I(x, y) = I(\text{round}(x), \text{round}(y))$

- interpolation bi-linéaire

55 / 56

Cas d'images discrètes

- $W(x, y)$ 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)

54 / 56

Plan du cours

1. Introduction à l'imagerie

56 / 56

Transformée discrète

TFD 2D

TFD d'une image discrète $f[m, n]$, $m = 0 \cdots M - 1$, $n = 0 \cdots N - 1$:

$$\hat{f}[u, v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n] e^{-2i\pi(u \frac{m}{M} + v \frac{n}{N})}$$

Inversion habituelle :

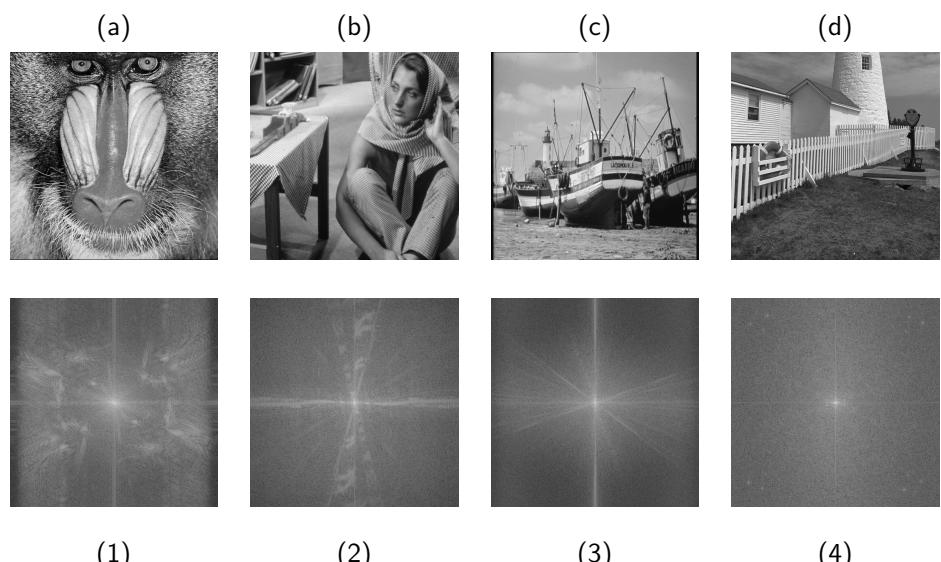
$$f[m, n] = \frac{1}{NM} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}[u, v] e^{2i\pi(m \frac{u}{M} + n \frac{v}{N})}$$

Propriétés:

- ▶ symétrie (pour une image réelle)
- ▶ FFT à temps de calcul : $\mathcal{O}(MN \log(MN))$

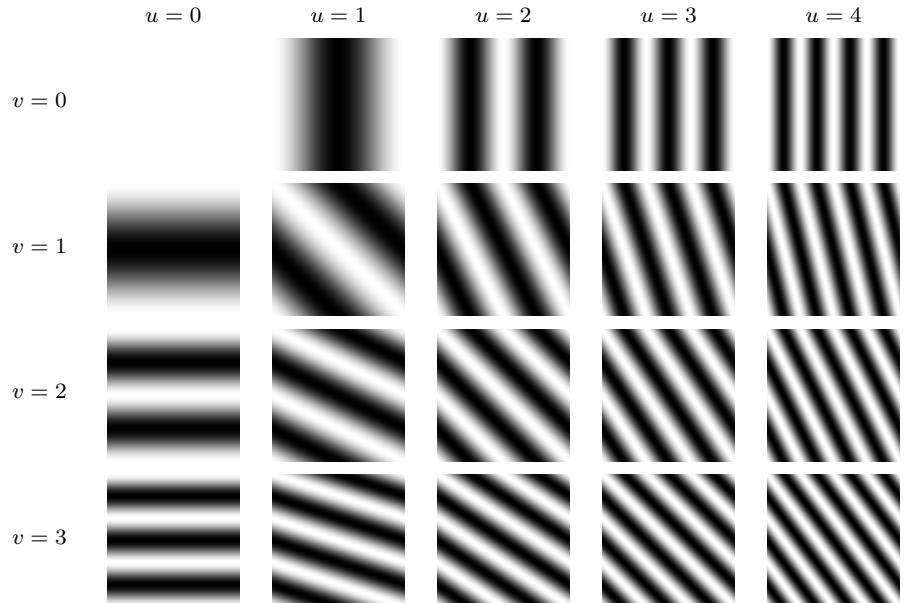
57 / 56

TFs d'images réelles



59 / 56

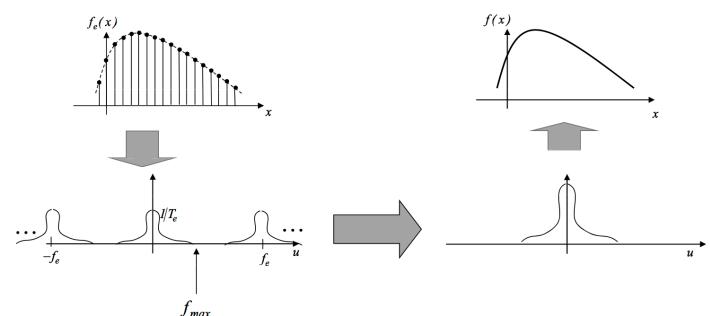
Base de Fourier 2D



58 / 56

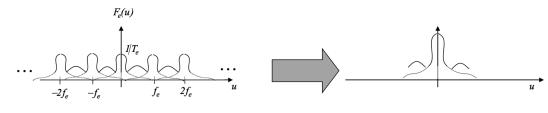
Reconstruction d'un signal échantillonné

- ▶ Étant donné $f_e(x)$ il est donc possible de retrouver $f(x)$



- ▶ À condition que $f_e \geq 2 f_{max}$ (fréquence de Nyquist)

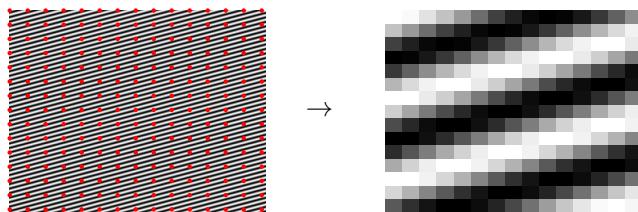
- ▶ Si $f_e < 2 f_{max}$, problème de repliement de spectre → aliasing



61 / 56

Aliasing

Si $f_e < 2f_{max}$, problème de repliement de spectre → **aliasing**



62 / 56

Convolution et filtrage linéaire

Opérateur linéaire invariant par translation

Un opérateur L tel que $\forall n_1, n_2, p_1, p_2 \in \mathbb{Z}$,

$$L(f[\cdot - p_1, \cdot - p_2]) [n_1, n_2] = L(f)[n_1 - p_1, n_2 - p_2].$$

- L est entièrement caractérisé par sa **réponse impulsionnelle**

$$h[n_1, n_2] = L\delta[n_1, n_2] \text{ avec } \delta \text{ le Dirac discret 2D}$$

- L'opérateur L est simplement une convolution :

$$Lf = h * f.$$

- L s'écrit dans le domaine fréquentiel avec la fonction de transfert \hat{h} :

$$\widehat{Lf} = \hat{f} \cdot \hat{h}.$$

64 / 56

Convolution discrète

Convolution discrète de f et $h \in \ell_1(\mathbb{R}^2)$:

$$(f * h)[n_1, n_2] = \sum_{p_1 \in \mathbb{Z}} \sum_{p_2 \in \mathbb{Z}} f[p_1, p_2]h[n_1 - p_1, n_2 - p_2].$$

Images finies, effets de bords

Lorsque f et h sont finies (disons des images de taille $N_1 \times N_2$) :

- Prolongement par 0 ("zero-padding")
- Convolution circulaire : prolongement périodique
- Autres : répétition bord, symétrie miroir, etc

Convolution circulaire et TF

Prolongement périodiques de f et h : $\forall n_1, n_2 \in \mathbb{Z}$, $\tilde{f}[n_1, n_2] = f[n_1 \bmod N_1, n_2 \bmod N_2]$.

$$f * h = \tilde{f} * \tilde{h}. \text{ Convolution circulaire et TFD : } \widehat{f * h} = \hat{f} \cdot \hat{h}.$$

63 / 56

Pourquoi filtrer?

- Éliminer des défauts dans une image (bruit d'acquisition, flou) → chap 4
- Déetecter des contours → chap 7
- Simplifier une image pour un futur traitement
- Améliorer la qualité d'une image



65 / 56

Filtrage dans le domaine spatial

- ▶ Défini par un masque de convolution fini h
- ▶ Souvent, masque de taille impaire → convolution centrée
- ▶ Filtres moyenneurs : $h_{ij} \geq 0$ et $\sum h_{ij} = 1$ (**passee-bas**)
- ▶ Filtres dérivateurs : $\sum h_{ij} = 0$ (**passee-haut**)
- ▶ Attention aux bords

▶ Un exemple :

100	100	100	100	100
100	100	100	100	100
100	100	150	100	100
100	100	100	100	100
100	100	100	100	100

*

0	-1	0
-1	5	-1
0	-1	0

=

100	100	100	100	100
100	100	50	100	100
100	50	350	50	100
100	100	50	100	100
100	100	100	100	100

66 / 56

Filtres passe-bas

- ▶ Aussi appelés filtres moyenneurs
- ▶ Ils gomment les différences d'intensité et génèrent visuellement un flou
- ▶ En général ils sont normalisés, séparables et isotropes : $h = h_1^T h_1$.

$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

filtre uniforme

$$\frac{1}{81} \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

filtre triangulaire

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 34 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

filtre binomial

Filtre gaussien : importance particulière. Pour l'implantation discrète, on utilise en général un filtre binomial.

68 / 56

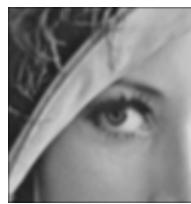
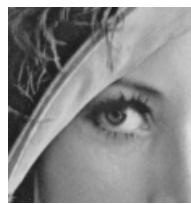
Un petit quizz

À quoi correspondent les masques de convolution suivants? À quoi servent-ils?

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

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

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

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


67 / 56

Filtres dérivateurs (passee-haut)

- ▶ Du premier ordre (gradients discrets) :
 - ▶ La convolution avec $h = [-1 1]$ approche $\frac{\partial}{\partial x_1}$ (différences finies)
 - ▶ Idem pour la convolution avec $[-1 1]^T$ et $\frac{\partial}{\partial x_2}$
 - ▶ On préfère souvent le filtre centré $[-1 0 1]$ et son transposé. Formes 2D correspondantes :

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ et } \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

- ▶ Au second ordre :

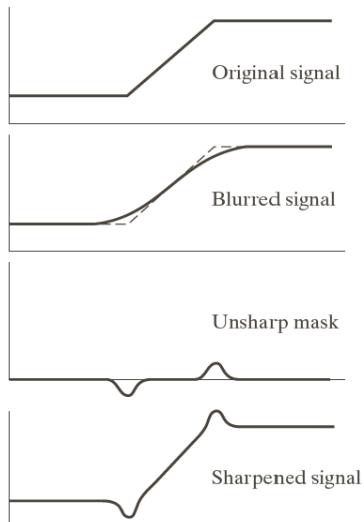
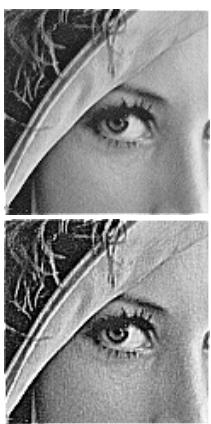
- ▶ La convolution avec $h = [1 -2 1]$ ou son transposé approchent $\frac{\partial^2}{\partial x_1^2}$ et $\frac{\partial^2}{\partial x_2^2}$
- ▶ Le Laplacien $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$ est approché par:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ ou } \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

69 / 56

Filtres réhausseurs de contours

- Principe : rajouter à l'image f des hautes fréquences. $h = \delta_0 - \gamma g$, avec g un filtre Laplacien.
- Illustration :



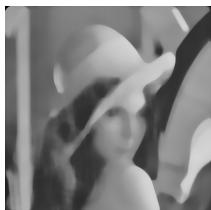
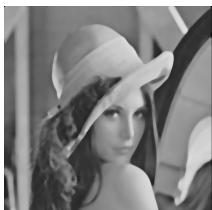
70 / 56

Filtres non-linéaires

- Par extension, on appelle filtrage non-linéaire une opération non-linéaire locale dans le domaine spatial
- Intérêt : élimination de bruit impulsionnel (cf. chap 4), lissage préservant les contours, réhaussement sans amplifier le bruit, etc
- Exemple : le filtre médian

$$g(x) = \text{med}\{f(z) \mid z \in S_x\},$$

S_x étant une fenêtre centrée en x , souvent carrée de taille fixée.



Image

Médian taille 5

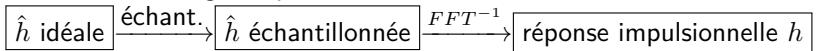
Médian taille 10

Médian taille 25

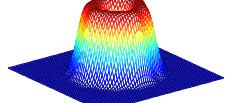
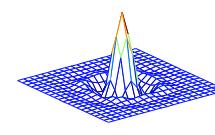
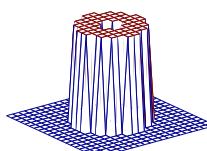
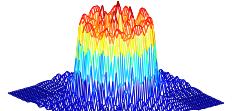
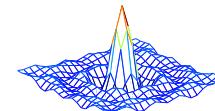
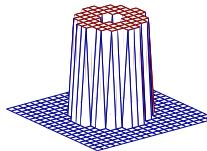
Synthèse de filtres RIF

Étant donnée \hat{h} idéale donnée, estimer la rep. imp. correspondante de taille fixée

- Par échantillonnage fréquentiel :



- Avec fenêtrage : on ajoute une étape de fenêtrage : $\tilde{h}(x) = h(x)w(x)$, avec en général une fenêtre circulaire ($w(x) = w_1(|x|)$) ou séparable ($w(x) = w_1(x_1)w_2(x_2)$).



71 / 56

Plan du cours

1. Introduction à l'imagerie

Représentation des images : motivation

- ▶ Transformation linéaire $T : x \mapsto y = T(x)$
- ▶ Cas particulier : changement de base orthonormée, T unitaire
- ▶ Exemple : la transformée de Fourier discrète

Pourquoi changer de représentation?

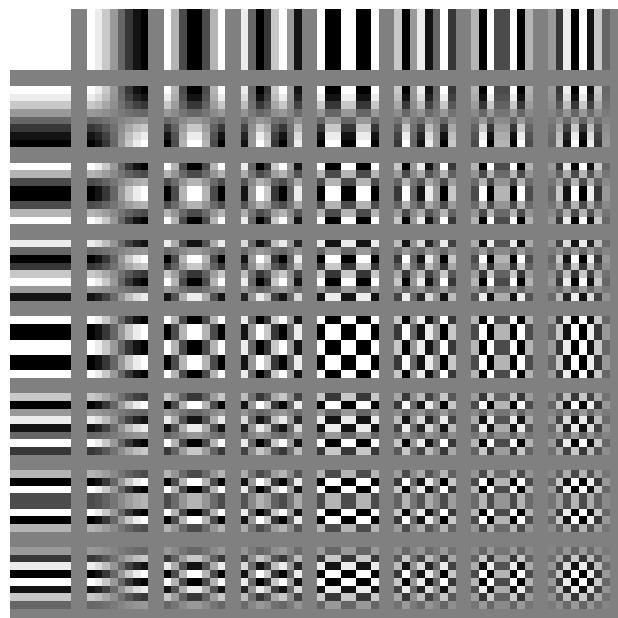
- ▶ Pour compresser. Exemple : image RGB sur 20 Mpixels = 60 Mo
- ▶ Pour estimer, et donc restaurer, reconstruire, analyser, etc
- ▶ ...

Comment choisir T ? But :

- ▶ Connaître ou prédire la statistique de y
- ▶ Avoir une transformée locale
- ▶ Rendre les coefficients y_k indépendants, ou du moins peu corrélés
- ▶ Rendre les coefficients **parcimonieux** : "concentrer" l'information dans peu de coefficients "utiles", les autres ≈ 0

74 / 56

TCD - fonctions de base ($N = M = 8$)



76 / 56

Transformée en cosinus discrète (DCT)

Soit une image à valeurs réelles $f : \llbracket 0, M-1 \rrbracket \times \llbracket 0, N-1 \rrbracket \rightarrow \mathbb{R}$, sa DCT $\tilde{f} : \llbracket 0, M-1 \rrbracket \times \llbracket 0, N-1 \rrbracket \rightarrow \mathbb{R}$ est définie par :

$$\tilde{f}[u, v] = \frac{\alpha_u}{\sqrt{M}} \frac{\alpha_v}{\sqrt{N}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n] \cos\left(\frac{\pi(2m+1)u}{2M}\right) \cos\left(\frac{\pi(2n+1)v}{2N}\right),$$

avec $\alpha_u = 1$ si $u = 0$, $\sqrt{2}$ sinon.

Propriétés

Similaires à la DFT :

- ▶ transformation séparable
- ▶ calcul rapide en $\mathcal{O}(NM \log(NM))$
- ▶ interprétation fréquentielle

En plus : transformation réelle, meilleure concentration de l'énergie

75 / 56

TCD – Illustration



Image originale



Image symétrisée



TCD (valeur absolue, log)

77 / 56

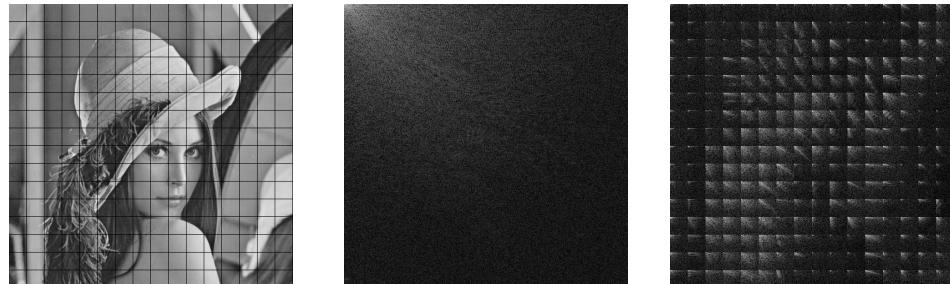
TCD par blocs

Problème de la non-stationnarité

On a déjà vu que les images sont fortement non-stationnaires : les propriétés statistiques ne peuvent pas être supposées constantes en espace.
→ besoin de représentations/transformations **locales**

DCT par blocs

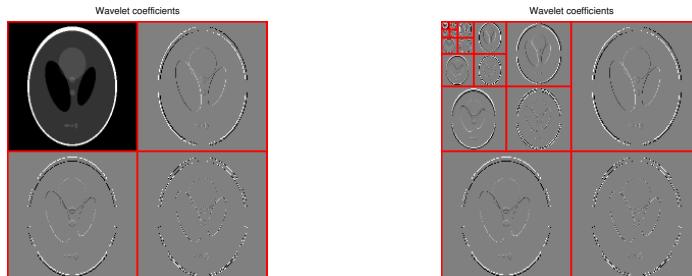
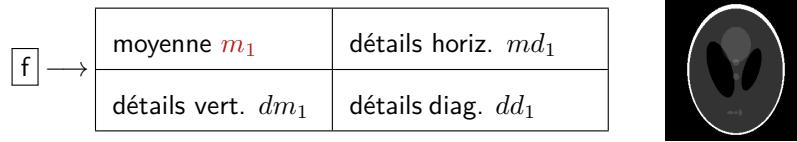
- ▶ Découpage de l'image en blocs (souvent : 8x8)
- ▶ DCT 2D sur chaque bloc



78 / 56

Base de Haar (2D)

On généralise en 2D :



81 / 56

Base de Haar (1d)

On considère un signal $s \in \mathbb{R}^{2^N}$. On pose $m_0 = s$, et on définit $\forall j = 0 \cdots N - 1$, $\forall t \in 0 \cdots 2^{N-j} - 1$:

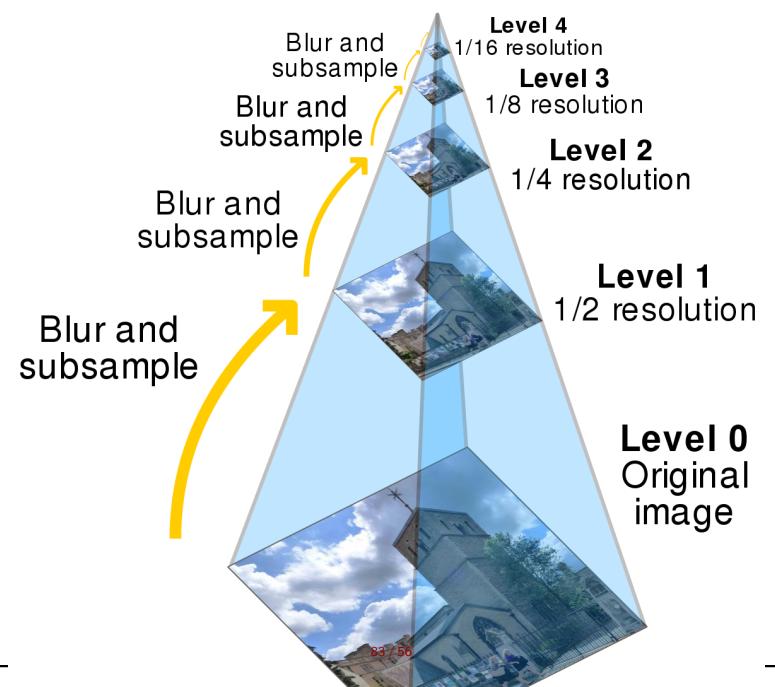
$$\begin{cases} d_{j+1}[t] &= \frac{1}{2}(m_j[2t+1] - m_j[2t]) \text{ (détails à l'échelle } j) \\ m_{j+1}[t] &= \frac{1}{2}(m_j[2t+1] + m_j[2t]) \text{ (approximation à l'échelle } j) \end{cases}$$

(d_1, \dots, d_J, m_J) est une décomposition complète (inversible) et multi-échelle de s .

niveau 0	[3	8	1	0	7	2	12	11]
niveau 1	$\frac{1}{2}[5$	-1	-5	-1]	$\frac{1}{2}[11$	1	9	23]
niveau 2	$\frac{1}{2}[5$	-1	-5	-1]	$\frac{1}{4}[-10$	14]	$\frac{1}{4}[12$	32]
niveau 3	$\frac{1}{2}[5$	-1	-5	-1]	$\frac{1}{4}[-10$	14]	$\frac{1}{8}[20$	$\frac{1}{8}[44$

80 / 56

Pyramides multi-échelles



Pourquoi comprimer?

Indispensable pour le stockage et la transmission dans de nombreuses situations!

Exemples de signaux numériques et leur taille :

- ▶ Photographie : 10 Megapixel sur 8 bits couleur = 30 Mo
- ▶ Image hyperspectrale, 256x660, 242 bandes spectrales, 12 bits = 61 Mo
- ▶ Vidéo : 2 heures de film, 800x600, 25 images/sec = 260 Go
- ▶ Audio : 10 min de musique en stéréo à 44.1 kHz sur 16 bits : 105 Mo

→ problématique commune à tous les signaux numériques

84 / 56

Critères de performance

- ▶ Taux de compression

$$T = \frac{\text{volume final}}{\text{volume initial}}$$

- ▶ Débit de codage

$$R = \frac{\text{volume final}}{\text{nombre de pixels}} \text{ bpp}$$

- ▶ Erreur quadratique moyenne

$$\text{MSE} = \frac{1}{NM} \sum_{n,m} (f_{n,m} - \tilde{f}_{n,m})^2$$

- ▶ Rapport signal sur bruit crête :

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right)$$

- ▶ Mesures de qualité perceptuelles : SSIM, analyse expérimentale, etc

86 / 56

Comment comprimer?

- ▶ Redondance statistique (pour toute classe de signaux) : répétitions, motifs récurrents, etc
- ▶ Autres redondances : zones homogènes, zones régulières (basse fréquence), à faible dynamique, etc

Spécificités de l'image

Prendre en compte le système visuel humain, comme la moindre sensibilité aux hautes fréquences des composantes de chrominance.

Avec ou sans perte?

- ▶ Sans : faible compression, souvent pas suffisant
- ▶ Avec : compromis entre qualité et compression

85 / 56

Principe de la compression par transformée

Idée générale

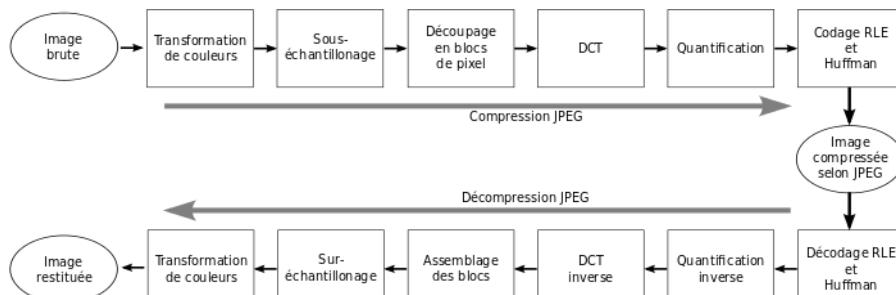
1. Trouver une **bonne** représentation
2. Quantifier efficacement les coefficients → avec pertes
3. Coder les coefficients quantifiés → théorie de l'information

Qu'est-ce qu'une **bonne** transformée?

1. Concentre de l'information (beaucoup de coefficients ≈ 0)
2. Décorrèle les données : des données indépendantes peuvent être codées séparément de manière optimale (quantification scalaire)

87 / 56

JPEG : principe général



"Compression JPEG" par Stéphane Brunner via Wikimedia Commons

88 / 56

Quantification

Étape cruciale de la compression. But : compresser tout en ne perdant pas trop d'information. Quantification uniforme, adaptative?

JPEG : quantification non uniforme

On utilise une matrice de quantification Q , et la quantification suivante :

$$F^*[u, v] = \begin{bmatrix} F[u, v] \\ Q[u, v] \end{bmatrix}$$

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \longrightarrow F^* = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

90 / 56

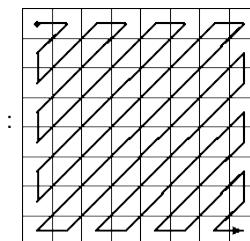
Pré-traitements et transformation

1. Décomposition luminance-chrominances YCbCr
2. Sous-échantillonnage des composantes de chrominance (souvent, facteur 2)
3. Découpage en blocs 8x8 pour chaque composante
4. DCT sur chaque bloc, exemple :

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix} \longrightarrow F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

89 / 56

Codage



Parcours du bloc F^* en Z :

Dans notre exemple : 79, 0, -2, -1, -1, 0, 0, -1, EOB.

Codage (sans pertes) :

- ▶ RLE (Run-length encoding) sur 0, pour coder les répétitions de 0
- ▶ Codage entropique : Huffman ou arithmétique

91 / 56