

L3, option IMAGE

Analyse d'images – introduction

<http://perception.inrialpes.fr/people/Boyer/Teaching/L3/>

Elise Arnaud - Edmond Boyer

Université Joseph Fourier / INRIA Rhône-Alpes

elise.arnaud@inrialpes.fr edmond.boyer@inrialpes.fr

Analyse d'images (Image Analysis) :

utilisation d'un ordinateur pour interpréter le monde extérieur au travers d'images.



- Traitement d'images (Image Processing).
- Reconnaissance des formes (Pattern Recognition).
- Vision par ordinateur (Computer Vision).

Plan

(i) Qu'est ce qu'une image ?

- notions de base
- codage de l'information et formats des fichiers
- interprétation mathématique
- histogramme

(ii) Vers une chaîne de traitement

Qu'est ce qu'une image ?

- Une image est avant tout un **signal 2D**
- Souvent, cette image représente une **scène 3D**

D'un point de vue mathématique

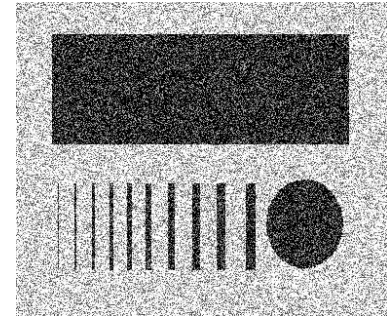
- une image est une matrice de nombres représentant un signal
- plusieurs outils permettent de manipuler ce signal

D'un point de vue humain

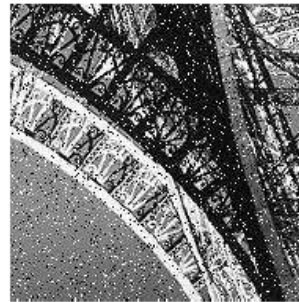
- une image contient plusieurs informations sémantiques
- il faut interpréter l'information au-delà de la valeur des nombres

Qu'est ce qu'une image ?

images naturelles vs images synthétiques



le bruit dans une image

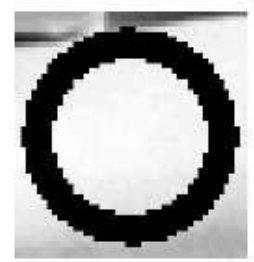
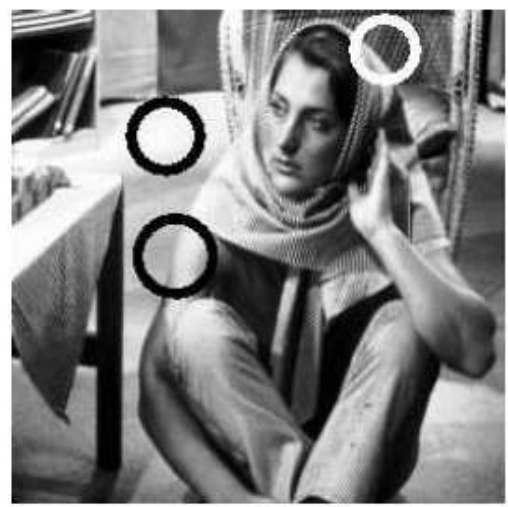


Le bruit est du à

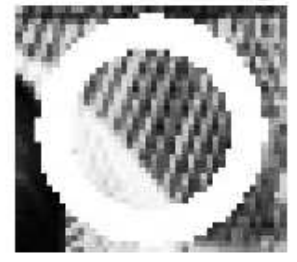
- la qualité de l'appareil d'acquisition
- les conditions de prise de vue : luminosité, mouvement de la scène, etc.

Qu'est ce qu'une image ?

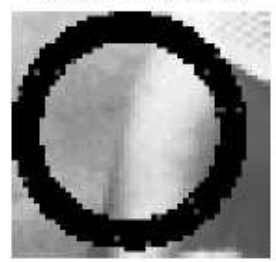
Quelques éléments simples d'une image



Une zone homogène

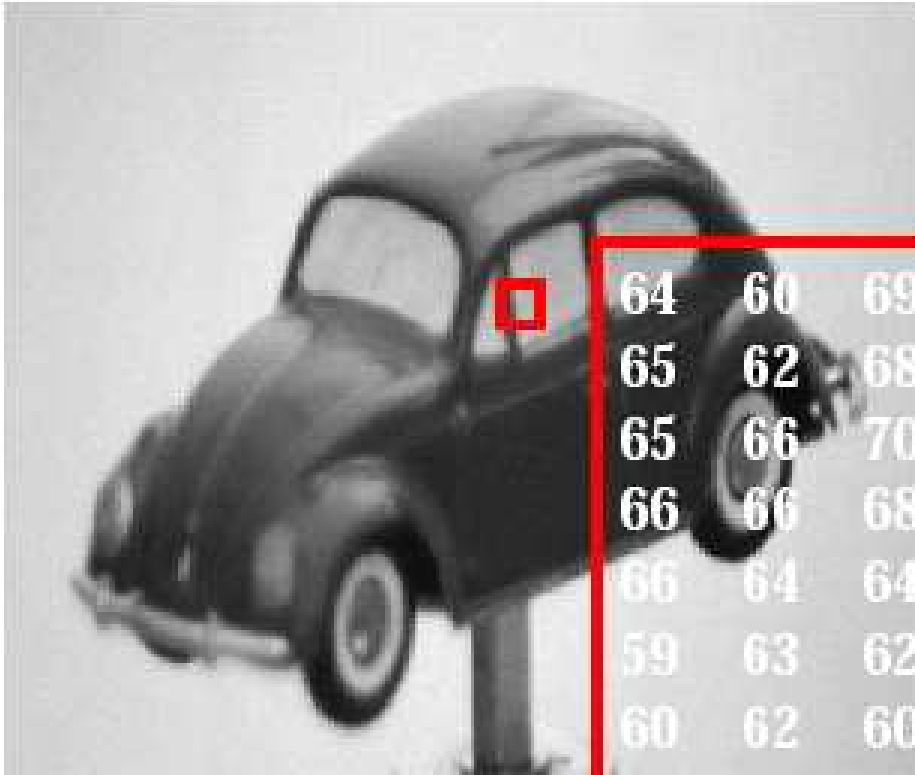


Une texture



Un contour

Qu'est ce qu'une image ?



Niveaux de gris - 8 bits:

0 - noir

255 - blanc

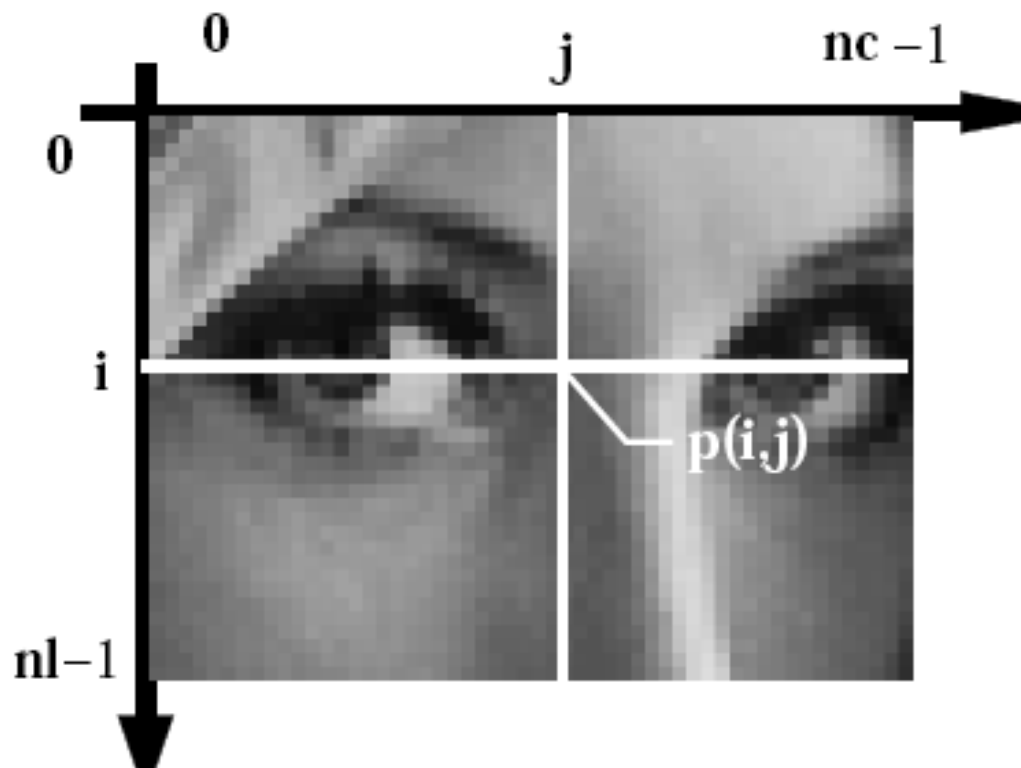
64	60	69	100	149	151	176	182	179
65	62	68	97	145	148	175	183	181
65	66	70	95	142	146	176	185	184
66	66	68	90	135	140	172	184	184
66	64	64	84	129	134	168	181	182
59	63	62	88	130	128	166	185	180
60	62	60	85	127	125	163	183	178
62	62	58	81	122	120	160	181	176
63	64	58	78	118	117	159	180	176

Qu'est ce qu'une image ?

Une image numérique est un tableau de pixel.

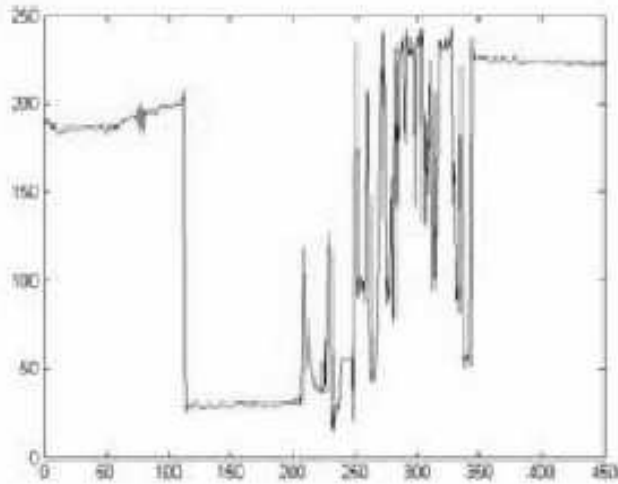
Un pixel s est décrit par :

- ses coordonnées dans l'image (i, j)
- sa valeur $I(i, j)$, représentant sa couleur (ou son niveau de gris)



Qu'est ce qu'une image ?

Un profil d'intensité d'une ligne dans une image est un signal 1D



Qu'est ce qu'une image ?

Résolution d'une image : finesse de la description spatiale



256x256



128x128



64x64



32x32

Quantification d'une image : plage de couleurs



6 bits



4 bits



3 bits



2 bits



1 bit

Le codage des couleurs

La valeur $I(i, j)$ d'un pixel $s = (i, j)$ représente son intensité lumineuse

En niveau de gris

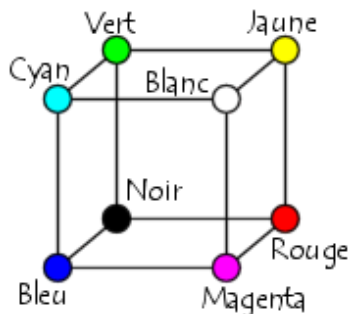
binaire : $I(i, j) = 0$ noir ou $I(i, j) = 1$ blanc

codage 8 bits : (le plus classique) $I(i, j) = 0, \dots, 255$ du plus foncé au plus clair

En couleur

codage dans l'espace **RGB** : trois intensités lumineuses rouge, vert, bleu.

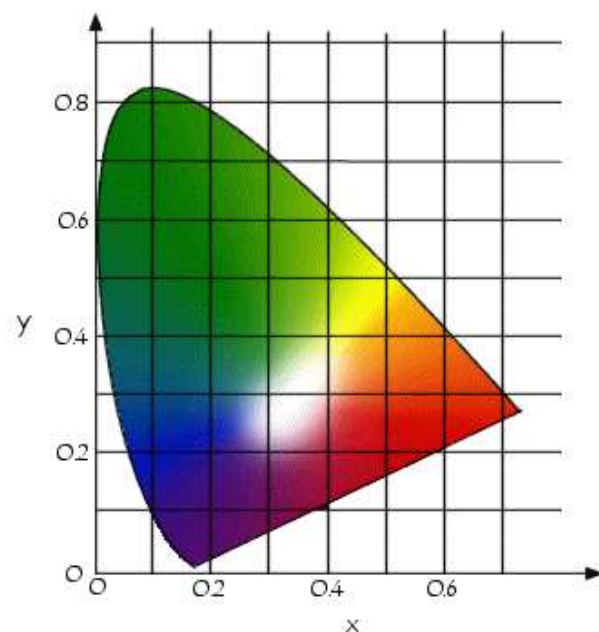
codage 24 bits : $I_R(i, j) = 0, \dots, 255$; $I_V(i, j) = 0, \dots, 255$; $I_B(i, j) = 0, \dots, 255$



Le codage des couleurs

l'espace Lab (espace à luminance séparée)

- L : la luminance, exprimée en pourcentage (0 pour le noir à 100 pour le blanc)
- a et b deux gammes de couleur allant respectivement du vert au rouge et du bleu au jaune avec des valeurs allant de -120 à +120.



Le codage des couleurs

l'espace HSL (Hue, Saturation, Luminance) (espace à luminance séparée)

- teinte : perception de la couleur (T-shirt mauve ou orange),
- saturation : pureté de la couleur (T-shirt neuf ou délavé)
- luminance : quantité de lumière de la couleur (T-shirt au soleil ou à l'ombre).

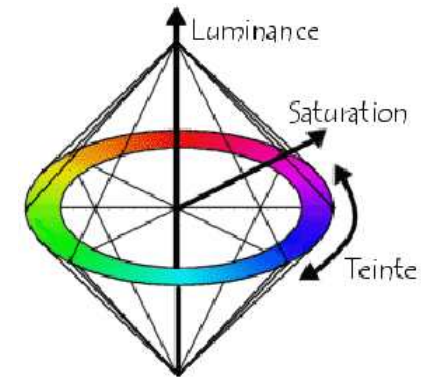


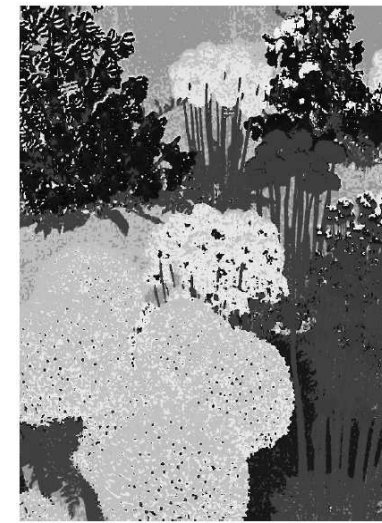
Image couleur



Composante valeur



Composante saturation



Composante teinte

L'image comme fonction

On peut voir l'image comme une fonction

$$I : S \rightarrow \Omega$$
$$(i, j) \rightarrow x = I(i, j)$$

En discret

$$S = \{0, 1, \dots, nl - 1\} \times \{0, 1, \dots, nc - 1\}$$
$$\Omega = \{0, 1, \dots, 255\}$$

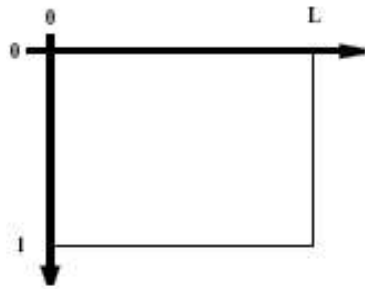
En continu

$$S = [0, nl - 1] \times [0, nc - 1]$$
$$\Omega = [0, P]$$

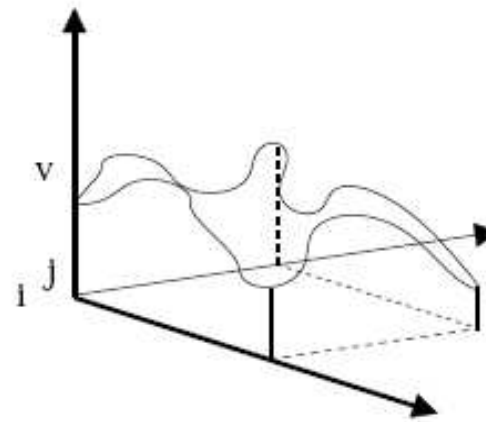
Avantage de la représentation continue : possibilité de dériver

L'image comme surface

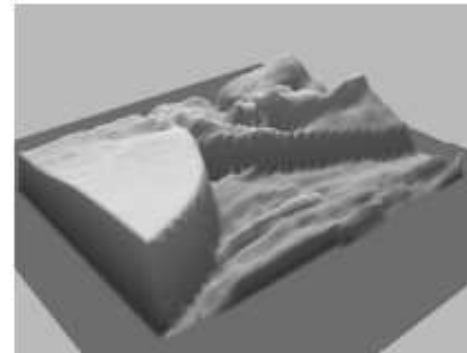
En utilisant la représentation précédente, on peut voir l'image comme une surface



devient



donne

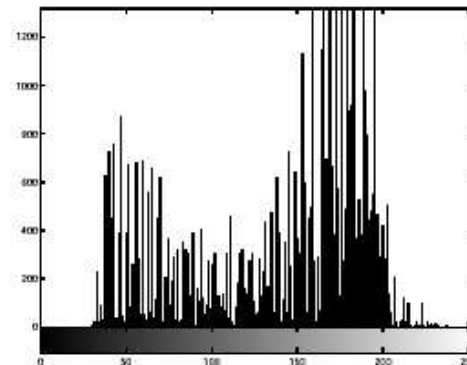
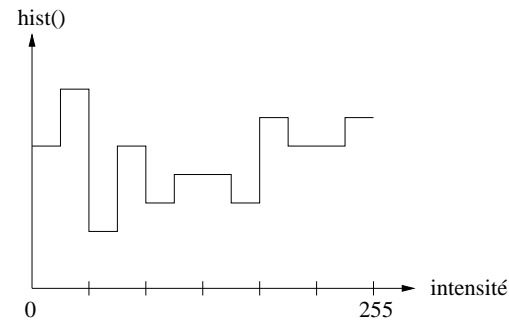


Histogramme d'une image

L'histogramme d'une image $h(x)$ est la fonction qui associe à une valeur d'intensité x le nombre de pixels dans l'image ayant cette valeur.

Algorithme

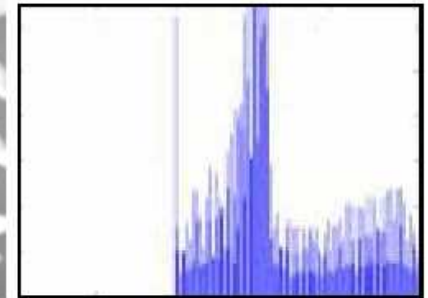
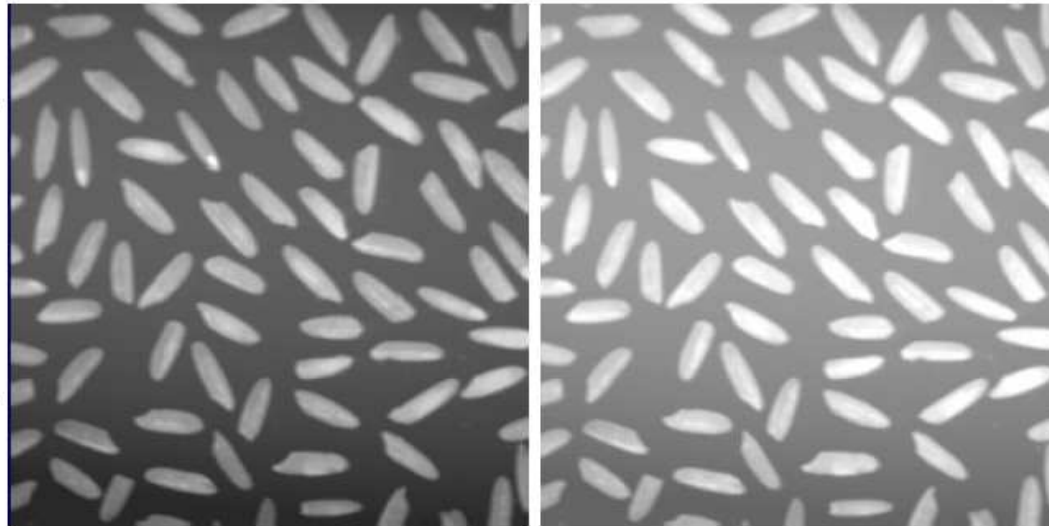
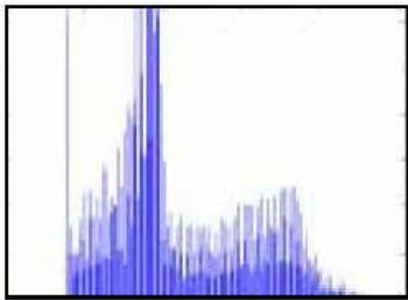
```
int row, col, rowmax, colmax;
int count[GREYMAX];
for(row = 0; row < rowmax; row++)
for(col = 0; col < colmax; col++)
    hist[image[row][col]]++;
```



(Pour une image couleur, il y a un histogramme par composante)

Histogramme d'une image

- **brillance d'un histogramme** : moyenne d'intensité des pixels de l'image
décalage de l'histogramme \Rightarrow augmentation de la brillance



Histogramme d'une image

- **histogramme normalisé** : proportion de pixel en fonction du niveau de gris (densité de probabilité)

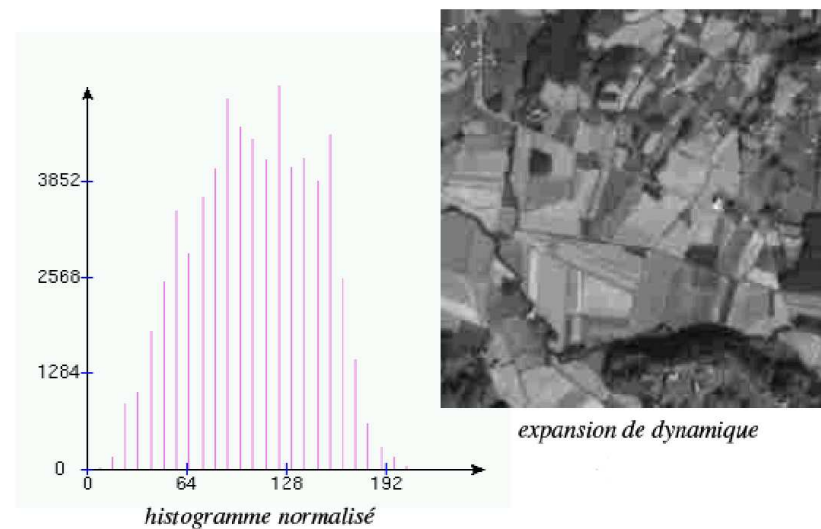
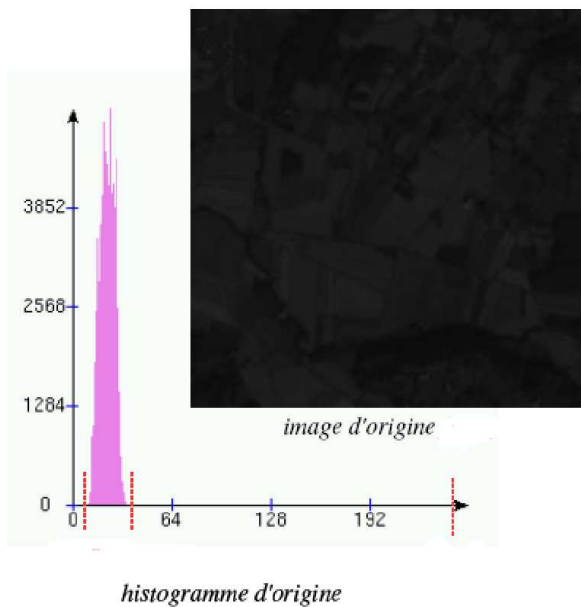
$$x \rightarrow h_n(x) = \frac{h(x)}{nb \text{ pixels}}$$

- **dynamique de l'image** : [min, max]
min : valeur minimale des niveaux de gris de l'image
max : valeur maximale des niveaux de gris de l'image

Histogramme d'une image

- **expansion de dynamique** : transformation des niveaux de gris de telle sorte que l'image utilise toute la dynamique (entre 0 et 255)

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



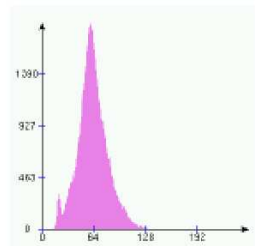
Histogramme d'une image

- **égalisation d'un histogramme** : transformation des niveaux de gris dont le but est d'équilibrer le mieux possible la distribution de pixels dans la dynamique
- l'histogramme cumulé $h_c(x) = \sum_{z \leq x} h_n(z)$ doit être le plus linéaire possible

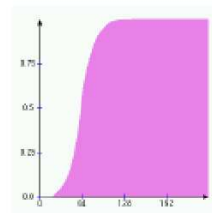
$$I_{Ega}(i, j) = 255 \times h_c(I(i, j))$$



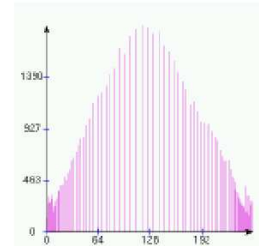
Original $f[x,y]$



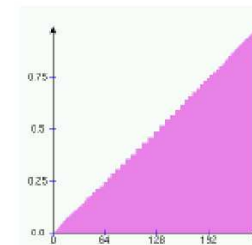
Histogramme



Histogramme cumulé



Histogramme



Histogramme cumulé



Après égalisation

Histogramme d'une image

La normalisation et l'égalisation sont des **transformations d'histogramme**

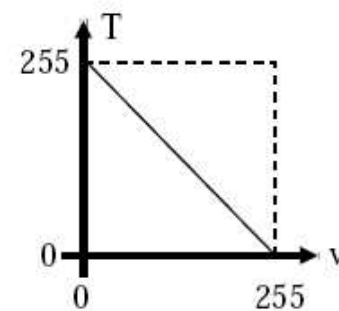
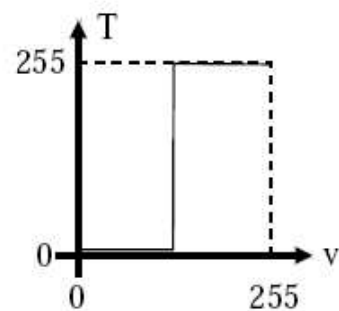
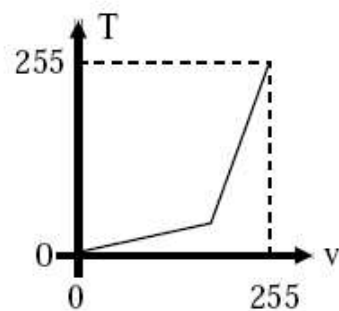
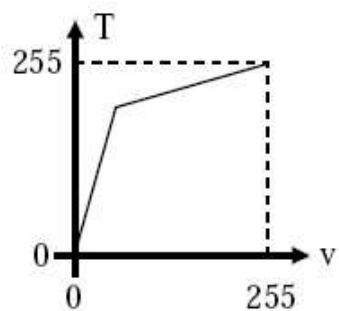
Principe algorithmique :

- parcourir les pixels de l'image
- pour chaque pixel (i, j)
 - (i) lire la valeur x
 - (ii) remplacer x par $T(x)$

La fonction $T(.)$ est représenté par son tracé. Le choix de $T(.)$ permet de modifier les caractéristiques d'une image (contraste, brillance, etc.)

Histogramme d'une image

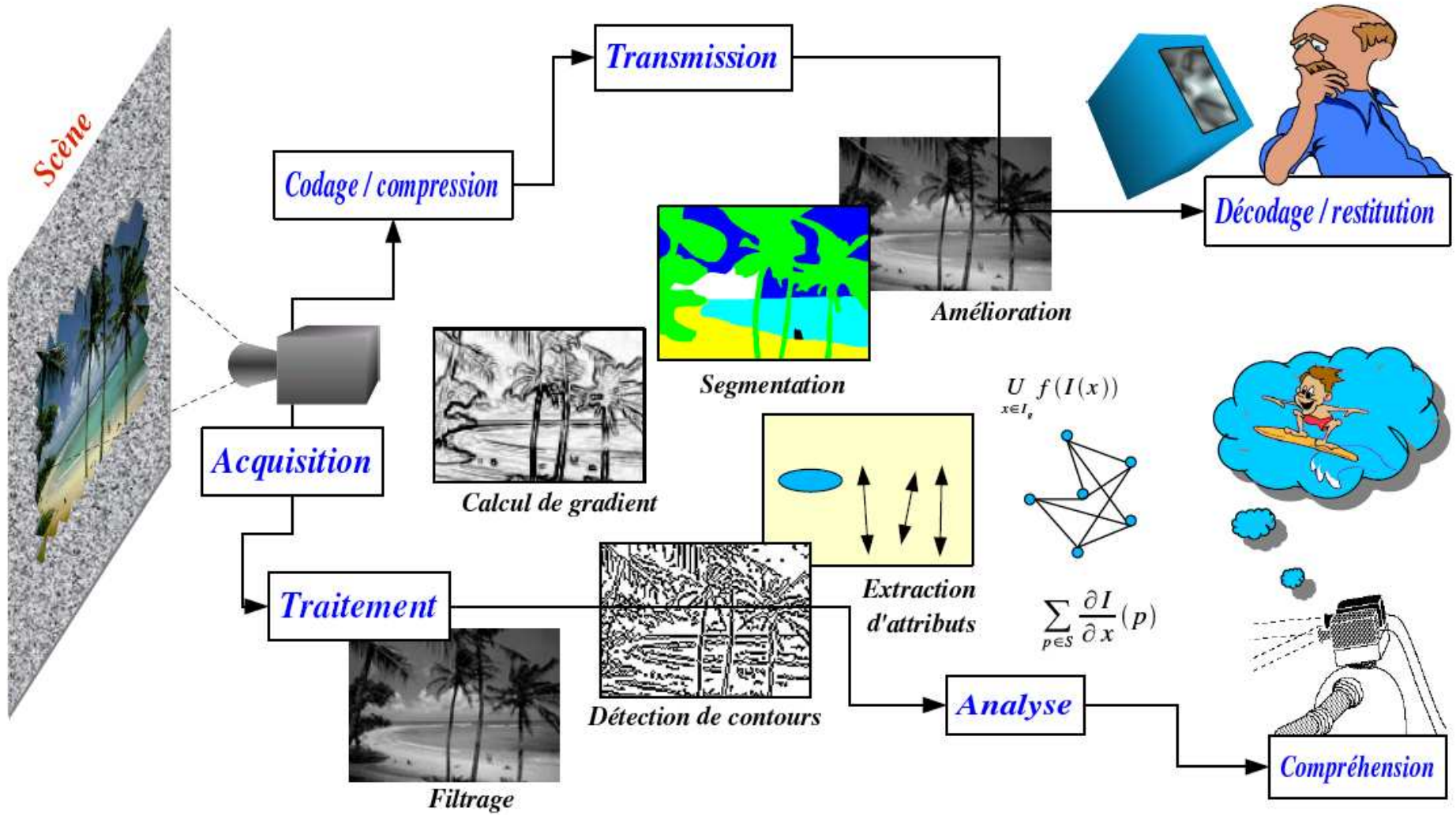
transformations d'histogramme

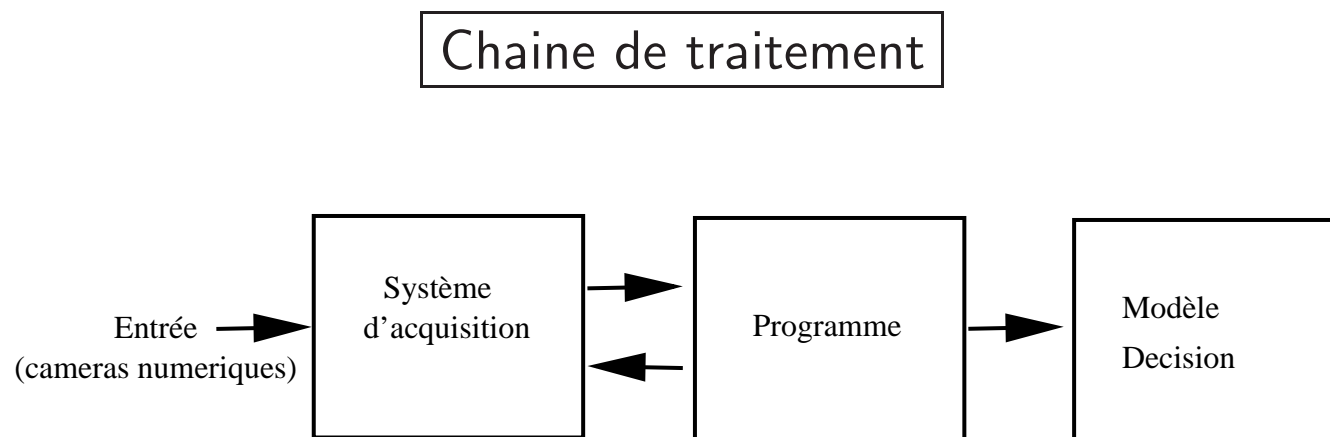


Plan

- (i) Quelles applications ?
- (ii) Complexité de la vision humaine
- (iii) Qu'est ce qu'une image ?
- (iv) **Vers une chaîne de traitement**

Chaîne de traitement





Les différentes phases de l'analyse d'une image :

- (i) Acquisition
- (ii) Traitement de bas niveaux : filtrage et extraction d'indices.
- (iii) Traitement de haut niveaux : reconnaissance des formes, reconstruction, calculs de positions, calculs de mouvements

Les éléments impliqués :

- (i) des primitives images : points d'intérêts, des segments, des contours.
- (ii) des caractéristiques photo-métriques : niveaux de gris, de couleurs.
- (iii) des caractéristiques géométriques : caméras, mouvements.

Chaîne de traitement

Différent niveaux d'analyse

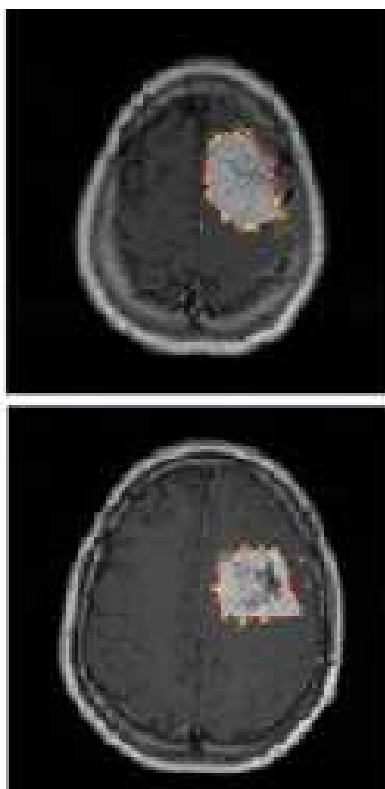
Exemple bas niveau : améliorer les caractéristiques d'une image



Chaîne de traitement

Différent niveaux d'analyse

Exemples bas niveau : détection de contours, détection de points d'intérêt



Chaîne de traitement

Différent niveaux d'analyse

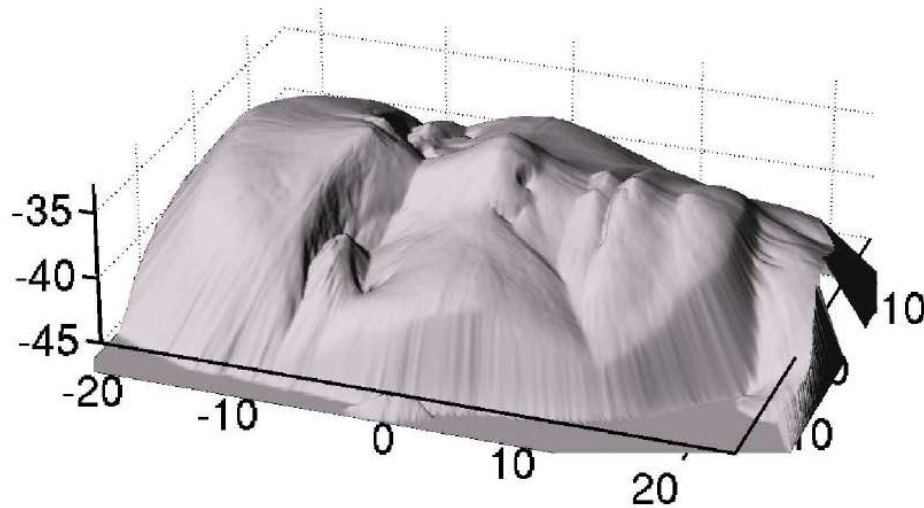
Exemples moyennant niveau : suivi 2D



Chaîne de traitement

Différent niveaux d'analyse

Exemples haut niveau : shape from shading



Chaîne de traitement

Différents niveaux d'analyse

Exemples haut niveau : suivi et reconstruction 3D

