

LE TRAITEMENT D'IMAGES

- HISTOGRAMME -

Jonathan Fabrizio

<http://jo.fabrizio.free.fr>

Version : Thu Feb 13 09:51:02 2025

Histogramme

L'histogramme dans le traitement des images (en niveaux de gris)

Applications directes

- Amélioration du contraste

- Autres corrections

Passage à la couleur

Applications

- Amélioration du contraste

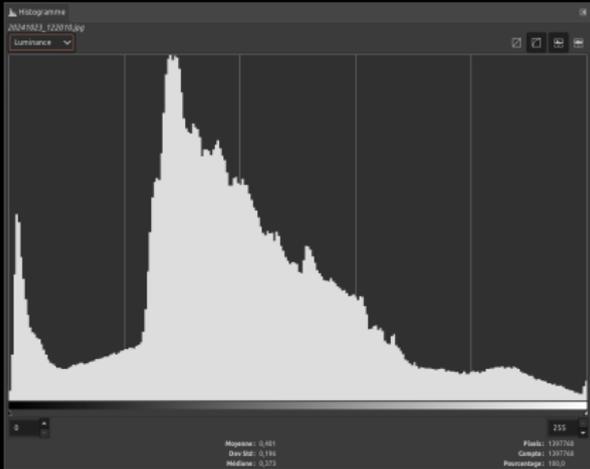
- Autres applications

- Réduction du nombre de couleur et binarisation

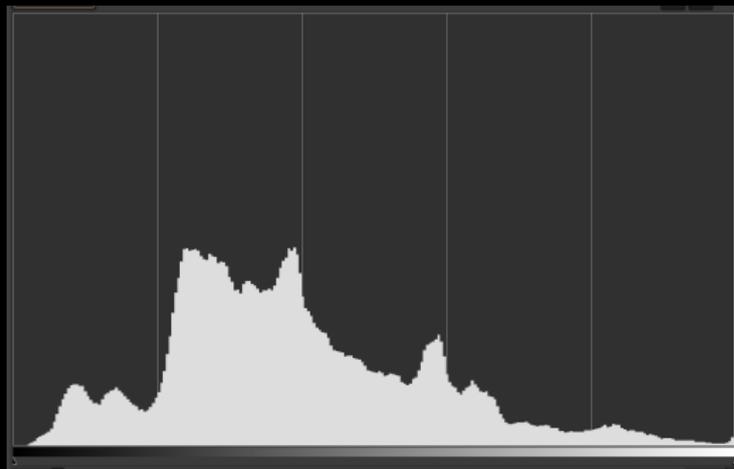
Conclusion

- L'histogramme :
 - Nombre d'occurrences par niveau
 - Recense les occurrences de chaque couleur
 - Donne une information globale sur l'image
 - Permet la réalisation de petits traitements globaux
 - Peut être calculé sur une image en couleur

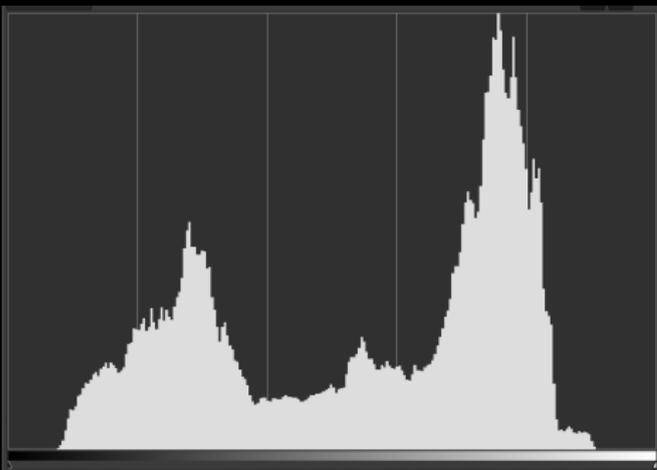
Exemples



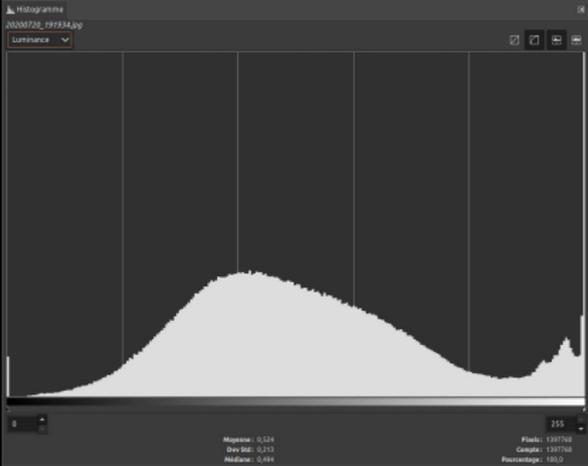
Exemples



Exemples



Exemples

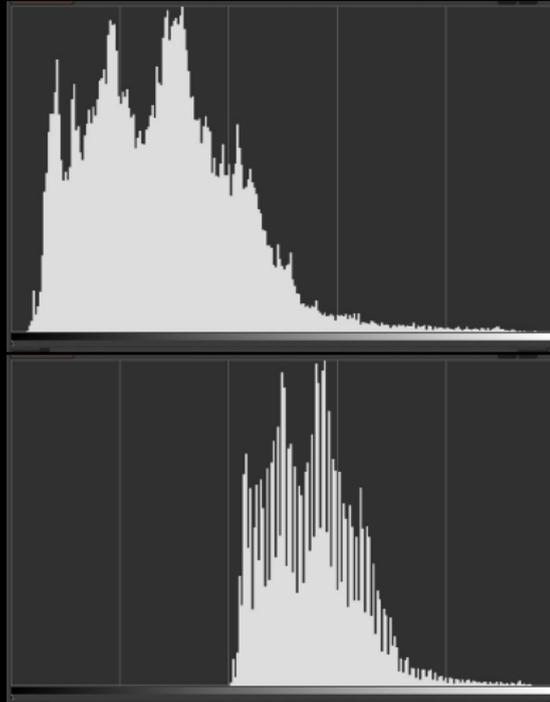


Calcul de l'histogramme

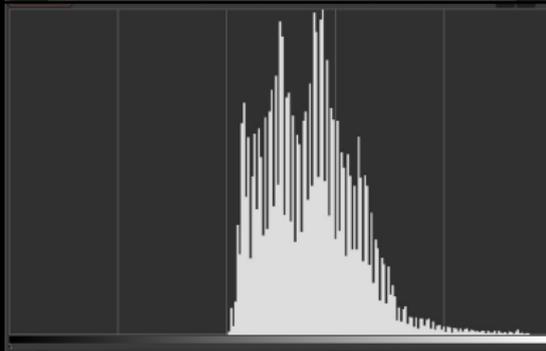
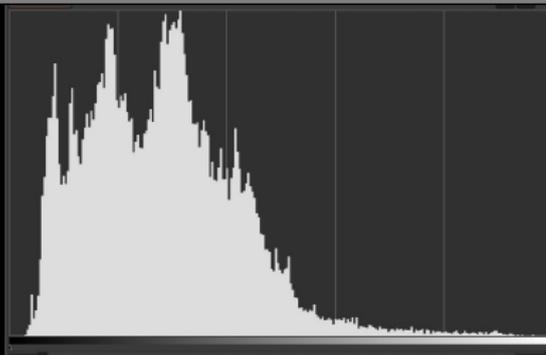
histogramme : tableau initialisé à 0
image : l'image sous forme d'un vecteur

```
for (offset=0;offset<sx*sy;++offset) {  
    histogramme[image[offset]]++;  
}
```

Informations sur l'image



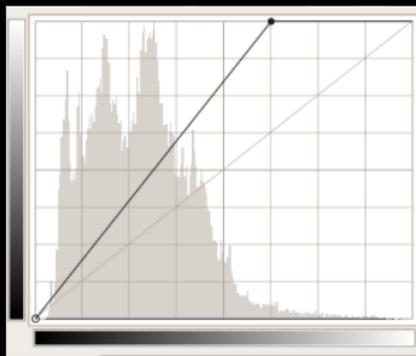
Informations sur l'image



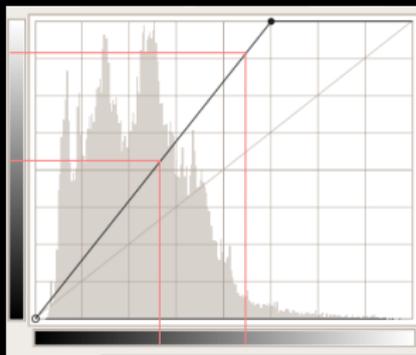
Amélioration du contraste

Amélioration du contraste

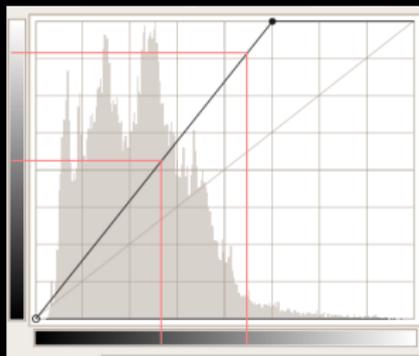
Amélioration du contraste



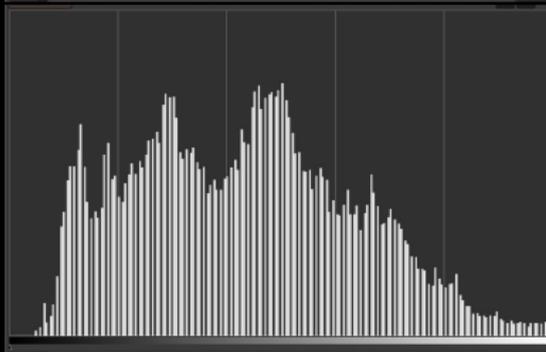
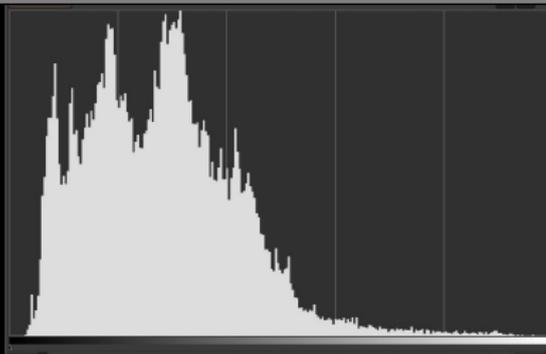
Amélioration du contraste



Amélioration du contraste

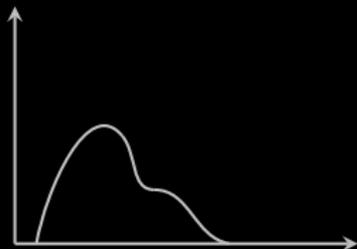


Amélioration du contraste



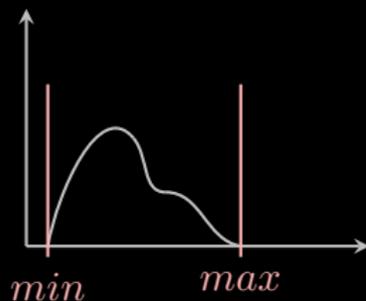
Amélioration du contraste

Généralisation : **Étirement d'histogramme**



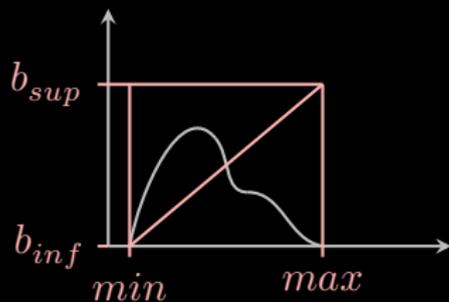
Amélioration du contraste

Généralisation : **Étirement d'histogramme**



Amélioration du contraste

Généralisation : **Étirement d'histogramme**

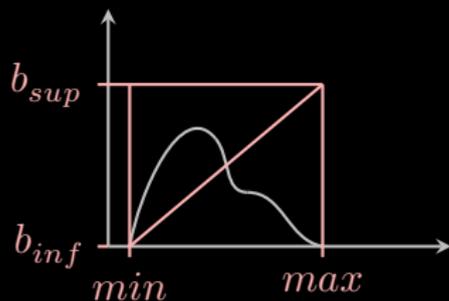


Amélioration du contraste

Généralisation : **Étirement d'histogramme**

Fonction de correction :

- $f(x) = ax + b$

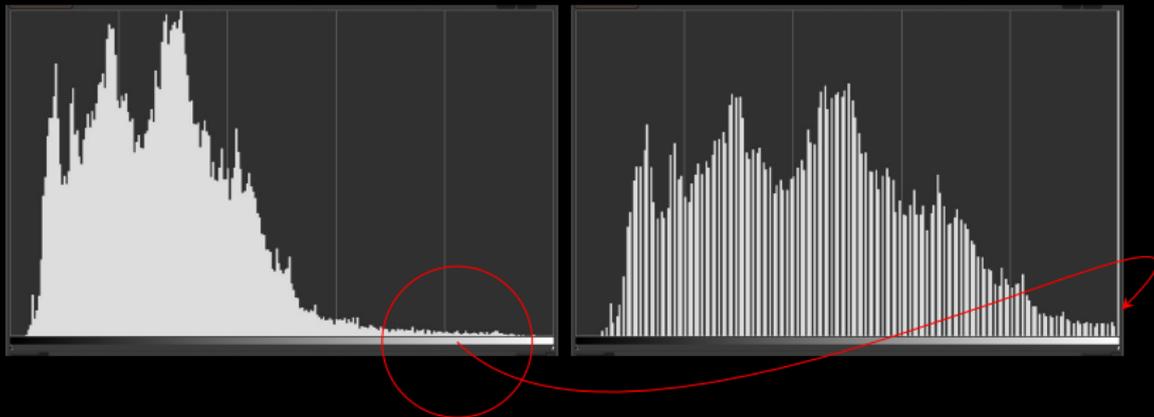


$$a = \frac{b_{sup} - b_{inf}}{max - min} \quad (1)$$

$$b = b_{inf} - ax \quad (2)$$

souvent $b_{inf} = 0$

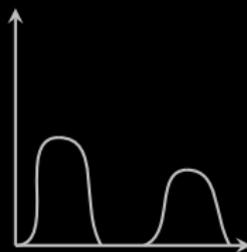
Amélioration du contraste : Étirement d'histogramme



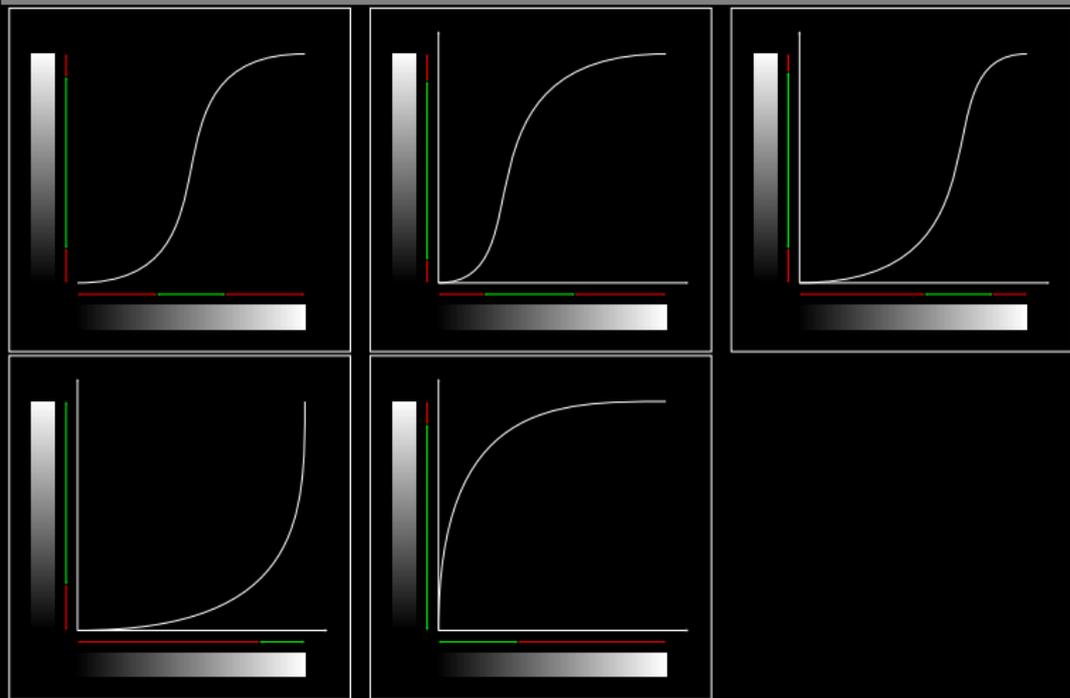
⚠ On a amélioré le contraste mais on a potentiellement perdu une partie de l'information !

Amélioration du contraste : Étirement d'histogramme

- peut ne rien apporter si l'histogramme (par exemple si les deux extrémités sont utilisées).
- peut dégrader le signal même si visuellement l'image semble meilleurs.



Diverses corrections



Diverses corrections

Choix de f

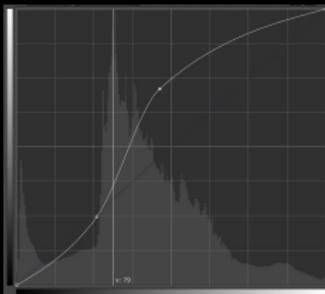
- log
 - si $x \neq 0$ $f(x) = \ln(x) / \ln(max) * max$
 - si $x = 0$ $f(x) = 0$
 - l'intervalle des zones sombres est augmentée.
- exp
 - L'intervalle des zones claires est augmentée
 - L'image est assombrie
 - Attention aux plages de valeurs ($\exp(255)...$)
- ...

Code de la correction

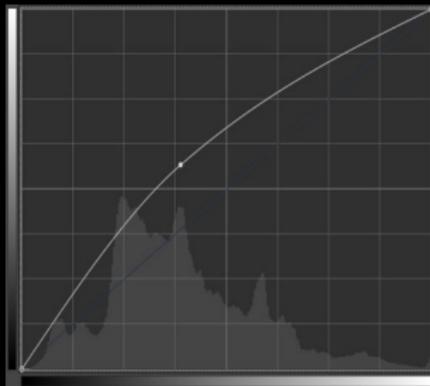
image : l'image sous forme d'un vecteur

```
for (offset=0; offset<sx*sy;++offset)  
image[offset]=f(image[offset]);
```

Amélioration du contraste/de la luminosité



Amélioration du contraste/de la luminosité



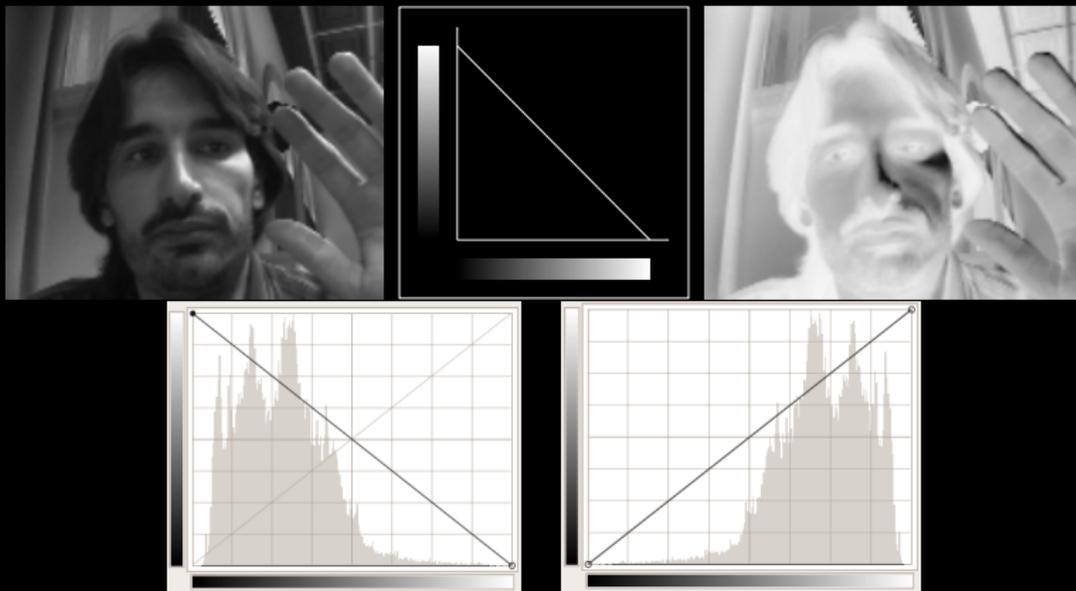
calcul du négatif

- Quelle fonction f prendre ?



calcul du négatif

- Quelle fonction f prendre? $f(x) = b_{sup} - x$



Amélioration du contraste

Histogramme cumulé $hc(x)$ de l'histogramme h :

- Si $x > 0$, $hc(x) = hc(x - 1) + h(x)$
- Si $x = 0$, $hc(x) = h(x)$

une idée :

- pour que le contraste soit optimal, il faut que l'histogramme soit plat
- cela induit que l'histogramme cumulé soit linéaire

Amélioration du contraste

- Essayer d'uniformiser la répartition des niveaux de gris dans l'histogramme
- Cela revient à essayer de rendre l'histogramme cumulé linéaire

Quelle fonction f prendre ?

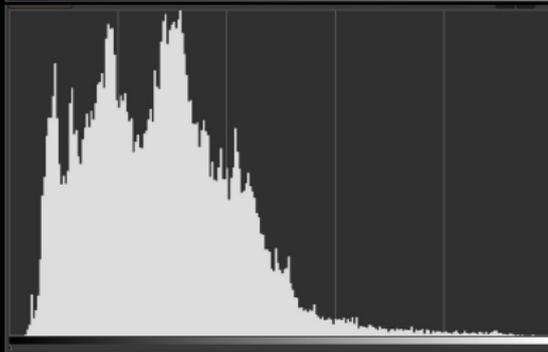
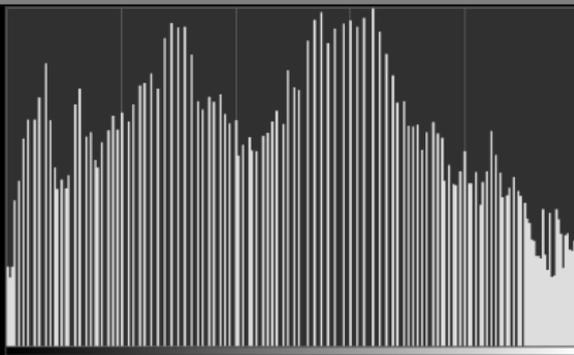
Amélioration du contraste

- Essayer d'uniformiser la répartition des niveaux de gris dans l'histogramme
- Cela revient à essayer de rendre l'histogramme cumulé linéaire

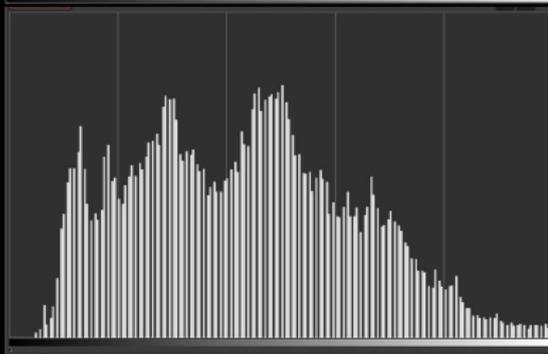
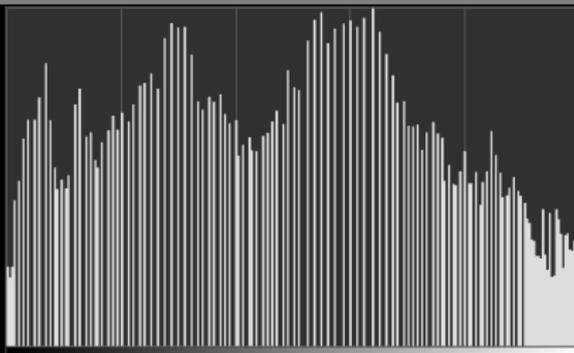
Quelle fonction f prendre ? $f(x) = b_{sup} * \frac{hc(x)}{nb_pix}$

Égalisation d'histogramme

Amélioration du contraste : Égalisation d'histogramme



Amélioration du contraste : Égalisation d'histogramme



Amélioration du contraste : Égalisation d'histogramme



Bilan

- Simple
- Rapide
- Efficace pour du pré-traitement

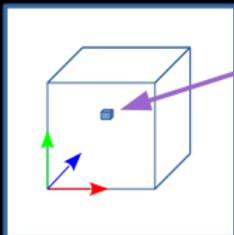
Pour aller plus loin :

- CLAHE
- Spécification d'histogramme (Imposer la forme de l'histogramme - comme pour l'égalisation qui donne un histogramme plat)
- ...

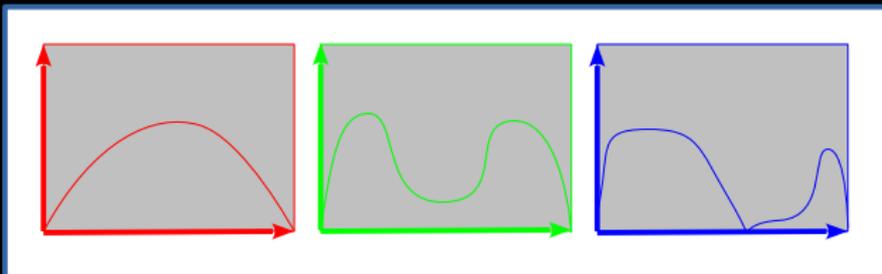
Images couleurs ?

Images couleurs ?

- Global



- Par plan



Images couleurs ?

Calcul :

- Global
- Par plan

Traitement :

- Indépendant sur chaque canal

Amélioration du contraste, égalisation d'histogramme

- comment faire ?



Source : wikipedia

Amélioration du contraste, égalisation d'histogramme



Source : wikipedia

Amélioration du contraste, égalisation d'histogramme



Source : wikipedia



égalisation sur chaque plan RGB

Images couleurs ?

Calcul :

- Global
- Par plan

Traitement :

- Indépendant sur chaque canal
- Sur le canal L de HLS ou V...

Amélioration du contraste, égalisation d'histogramme



Source : wikipedia

Amélioration du contraste, égalisation d'histogramme



Source : wikipedia



égalisation sur chaque plan RGB

Amélioration du contraste, égalisation d'histogramme



Source : wikipedia



égalisation sur chaque plan RGB



égalisation sur L

Amélioration du contraste, égalisation d'histogramme



Source : wikipedia



égalisation sur chaque plan RGB



égalisation sur L



égalisation sur V

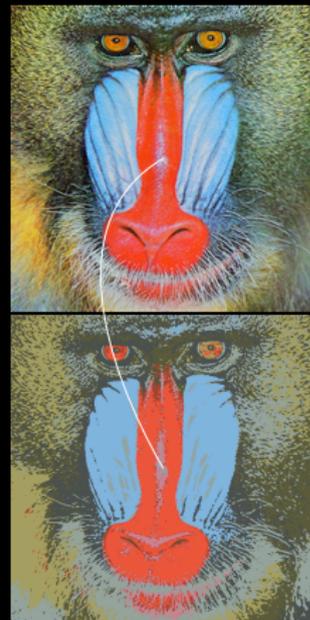
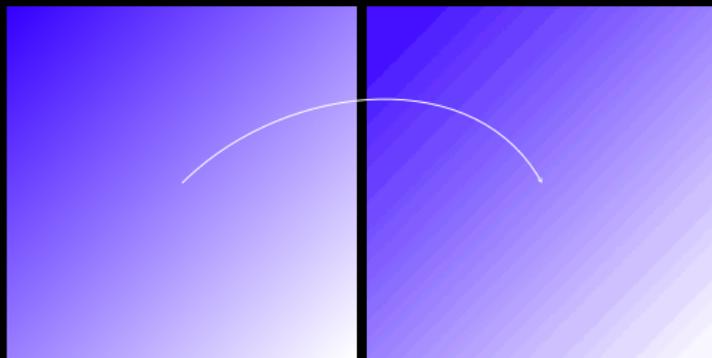
Comparaisons

- Indexation/comparaison (Distance entre histogrammes)
 - Comparaison d'images
 - Segmentation automatique en plan de séquences
 - Différences entre images consécutives
- Distances :
 - Bin-by-bin distances
 - Distance de Hellinger Bhattacharyya
 - ...
 - Cross-bin distances
 - Earth Mover's Distance
 - ...

Réduction du nombre de couleurs

- Pourquoi diminuer le nombre de couleurs ?
 - Simplifier l'image
 - Diminuer l'espace nécessaire au stockage
 - Focaliser sur les éléments qui nous intéressent
 - Effet artistique
- Pourquoi plus précisément passer de la couleur aux niveaux de gris ?
 - Traitement de la couleur pas toujours aisée
 - Plusieurs canaux
 - Pas vraiment de relation d'ordre utilisable avec la couleur
- Pourquoi plus précisément passer en noir et blanc ?
 - Focaliser sur les éléments qui nous intéressent
 - Séparation fond/forme (O.C.R. ...)

Objectif : Réduire le nombre de couleurs utilisées tout en conservant le plus possible une image proche de l'original



Algorithme du *Median cut*

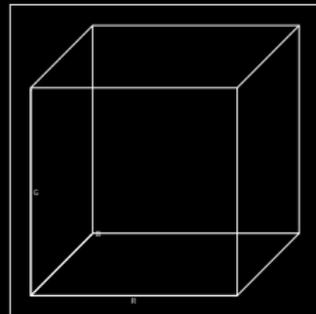
- Basée sur l'étude de l'histogramme

Digression : diffusion de l'erreur (adoucit certaines erreurs pour la visualisation).

Algorithme du *Median cut*

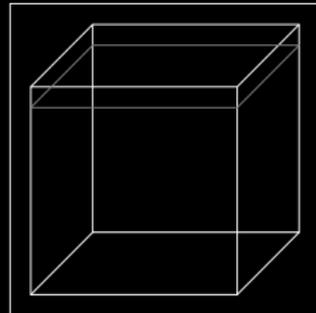
Algorithme du *Median cut*

- Calcul l'histogramme couleur de l'image



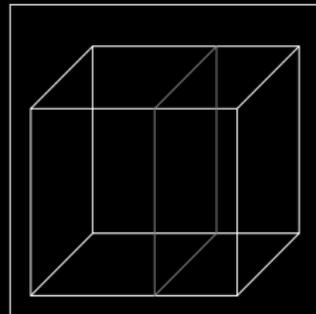
Algorithme du *Median cut*

- Calcul l'histogramme couleur de l'image
- Élimination des extrémités vides



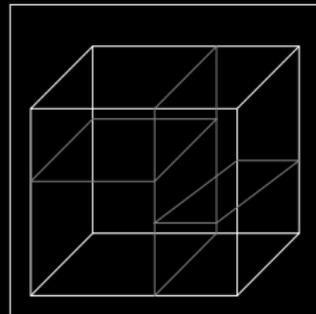
Algorithme du *Median cut*

- Calcul l'histogramme couleur de l'image
- Élimination des extrémités vides
- Découpage du parallélépipède restant en deux sous blocs contenant autant de points



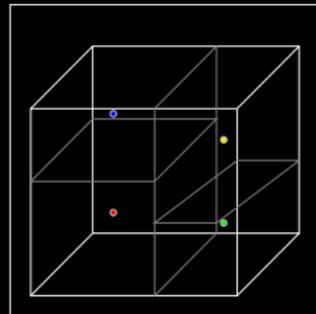
Algorithme du *Median cut*

- Calcul l'histogramme couleur de l'image
- Élimination des extrémités vides
- Découpage du parallélépipède restant en deux sous blocs contenant autant de points
- Pour chaque sous bloc, recommencer jusqu'à avoir autant de sous blocs que de couleurs souhaitées



Algorithme du *Median cut*

- Calcul l'histogramme couleur de l'image
- Élimination des extrémités vides
- Découpage du parallélépipède restant en deux sous blocs contenant autant de points
- Pour chaque sous bloc, recommencer jusqu'à avoir autant de sous blocs que de couleurs souhaitées
- Trouver pour chaque partie, une couleur représentante



Digression : diffusion de l'erreur

- Quand on substitue une couleur par son représentant, on commet une petite erreur.
- Le but de la diffusion de l'erreur est de compenser l'erreur commise sur un pixel en propageant cette erreur sur les pixels voisins.

Algorithme de *FloydSteinberg*

- On parcourt les pixels de haut en bas et de gauche à droite,
- Pour chaque pixel, on substitue la valeur d'un pixel par son représentant,
- On calcule l'erreur commise et on la répartit suivant :

	X	-7/16
-3/16	-5/16	- 1/16

Algorithme de *FloydSteinberg*

-

	16	18
12	15	21

Algorithme de *FloydSteinberg*

- | | | |
|----|----|----|
| | 16 | 18 |
| 12 | 15 | 21 |
- | | | |
|----|----|----|
| | 25 | 18 |
| 12 | 15 | 21 |

 (erreur 9)

Algorithme de *FloydSteinberg*

- | | | |
|----|----|----|
| | 16 | 18 |
| 12 | 15 | 21 |
- | | | |
|----|----|----|
| | 25 | 18 |
| 12 | 15 | 21 |

 (erreur 9)
- | | | |
|-------------------|-------------------|-------------------|
| | 25 | $18 - 9 * 7 / 16$ |
| $12 - 9 * 3 / 16$ | $15 - 9 * 5 / 16$ | $21 - 9 * 1 / 16$ |

Algorithme de FloydSteinberg

- | | | |
|----|----|----|
| | 16 | 18 |
| 12 | 15 | 21 |
- | | | |
|----|----|----|
| | 25 | 18 |
| 12 | 15 | 21 |

 (erreur 9)
- | | | |
|-------------------|-------------------|-------------------|
| | 25 | $18 - 9 * 7 / 16$ |
| $12 - 9 * 3 / 16$ | $15 - 9 * 5 / 16$ | $21 - 9 * 1 / 16$ |
- | | | |
|------|-------|-------|
| | 25 | 14 |
| 10.3 | 12.19 | 20.44 |

Algorithme de *FloydSteinberg*

- | | | |
|----|----|----|
| | 16 | 18 |
| 12 | 15 | 21 |
- | | | |
|----|----|----|
| | 25 | 18 |
| 12 | 15 | 21 |

 (erreur 9)
- | | | |
|-------------------|-------------------|-------------------|
| | 25 | $18 - 9 * 7 / 16$ |
| $12 - 9 * 3 / 16$ | $15 - 9 * 5 / 16$ | $21 - 9 * 1 / 16$ |
- | | | |
|------|-------|-------|
| | 25 | 14 |
| 10.3 | 12.19 | 20.44 |
- Pour le pixel voisin je considère donc la valeur 14 et non 18.

Algorithme de *FloydSteinberg*

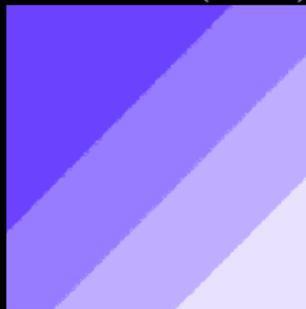
Résultats :



Image originale
(32Ko)



16 couleurs (4.6ko) / 16 couleurs + diffusion (21ko)

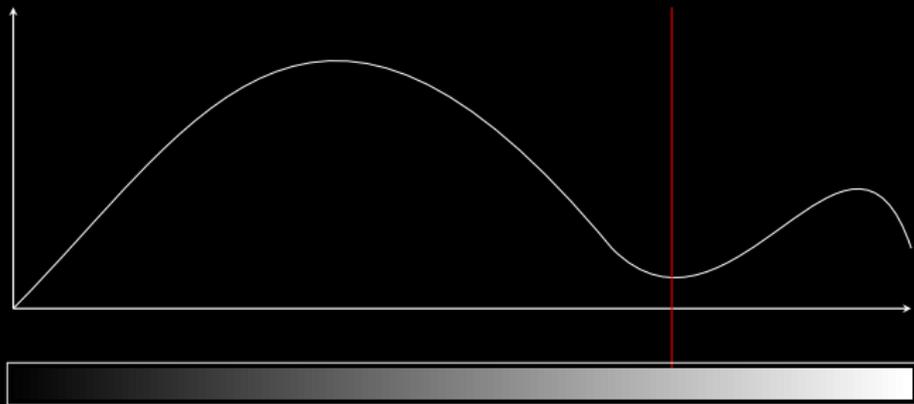


4 couleurs (2ko) / 4 couleurs + diffusion (14.9ko)

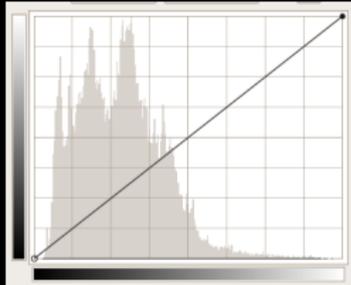
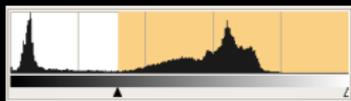
Passage en noir et blanc (binarisation)

- Séparation Fond/forme.

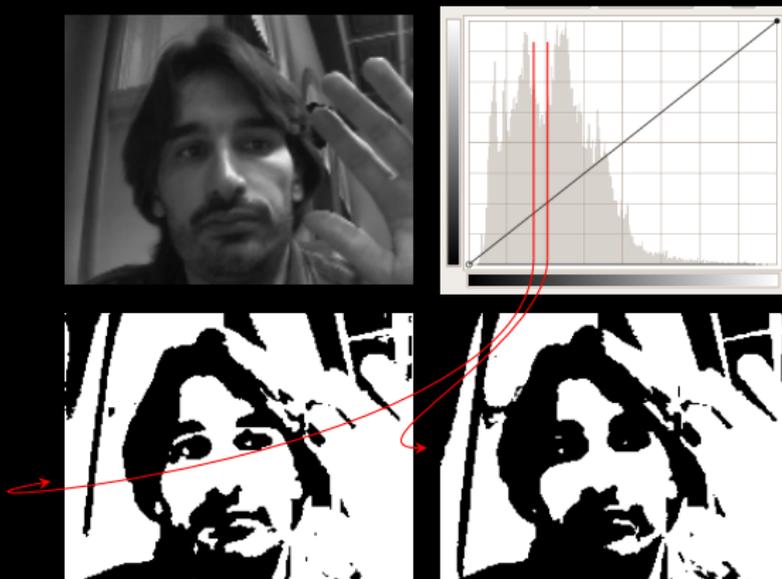
- Seuil global
 - Utilisation de l'histogramme
 - On suppose l'histogramme bi-modal (1 mod pour le fond et 1 pour la forme)
 - Trouver le niveau de gris à la jonction entre les deux



Seuil global



Seuil global



Seuil global

Comment trouver ce seuil automatiquement ?

Un algorithme simple :

Seuil global

Comment trouver ce seuil automatiquement ?

Un algorithme simple :

- Supposons un seuil T initial

Seuil global

Comment trouver ce seuil automatiquement ?

Un algorithme simple :

- Supposons un seuil T initial
- Calculons les moyennes m_1 et m_2 des ensembles des pixels d'intensité inférieur à T et supérieur ou égale à T respectivement

Seuil global

Comment trouver ce seuil automatiquement ?

Un algorithme simple :

- Supposons un seuil T initial
- Calculons les moyennes m_1 et m_2 des ensembles des pixels d'intensité inférieur à T et supérieur ou égale à T respectivement
- Corriger T avec $T = (m_1 + m_2) / 2$

Seuil global

Comment trouver ce seuil automatiquement ?

Un algorithme simple :

- Supposons un seuil T initial
- Calculons les moyennes m_1 et m_2 des ensembles des pixels d'intensité inférieur à T et supérieur ou égale à T respectivement
- Corriger T avec $T = (m_1 + m_2) / 2$
- Si $T > \Delta T$ continuer en 2

Seuil global : Le critère d'Ostu

Comment trouver ce seuil automatiquement ?

Un algorithme un peu plus recherché :

Seuil global : Le critère d'Ostu

Comment trouver ce seuil automatiquement ?

Un algorithme un peu plus recherché :

- On cherche deux classes

Seuil global : Le critère d'Ostu

Comment trouver ce seuil automatiquement ?

Un algorithme un peu plus recherché :

- On cherche deux classes
 - Minimiser la variance intra-classe
 - Maximiser la variance inter-classe

Seuil global : Le critère d'Ostu

Comment trouver ce seuil automatiquement ?

Un algorithme un peu plus recherché :

- On cherche deux classes
 - Minimiser la variance intra-classe
 - Maximiser la variance inter-classe
- $m_1(k)$ et $m_2(k)$ les moyennes des deux classes formées par le seuil k
- m_g la moyenne
- $p_1(k)$ et $p_2(k)$ les probabilités d'occurrence des deux classes formées par le seuil k
- Maximiser la variance inter-classe :
 - $$\sigma(k)^2 = p_1(k)(m_1(k) - m_g)^2 + p_2(k)(m_2(k) - m_g)^2$$

Seuil global : Le critère d'Ostu

- Maximiser la variance inter-classe :

$$\sigma(k)^2 = p_1(k)(m_1(k) - m_g)^2 + p_2(k)(m_2(k) - m_g)^2$$

$$\text{Or } p_1(k)m_1(k) + p_2(k)m_2(k) = m_g \text{ et } p_1(k) + p_2(k) = 1$$

$$\sigma(k)^2 = p_1(k)p_2(k)(m_1(k) - m_2(k))^2$$

$$= (m_g p_1(k) - m_1(k))^2 / (p_1(k)(1 - p_1(k)))$$

Reviens à chercher le k dans l'intervalle où $p_1(k)(1 - p_1(k)) \neq 0$ tel que $\sigma(k)^2$ est maximum (si plusieurs max, faire la moyenne).

Seuil global : Le critère d'Ostu

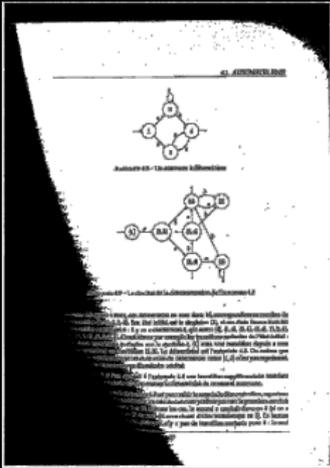
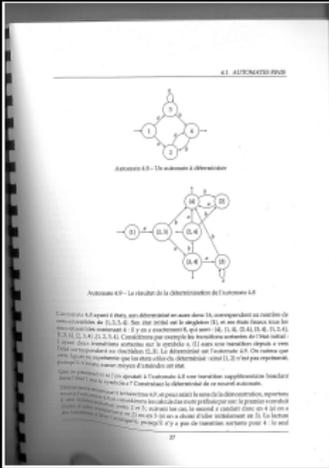
Résultat :

Seuil global : Le critère d'Ostu

Résultat :



Seuil global : Le critère d'Ostu
Résultat :



Seuil global : Le critère d'Ostu

- Rapide et simple
- Se calcul directement sur l'histogramme
- Dans la pratique pas toujours efficace selon le contexte (le seuil est global !)
- Régulièrement mal utilisé
- Moins utile (utilisation d'outils de segmentation issus du deep ou même pas de nécessité de séparer le fond et la forme)

Histogramme

Histogramme

- Simple à calculer
- Simple à modifier
- Rapide

Applications :

- pre-traitements (amélioration de contraste)
- diminution du nombre de couleurs
- indexation/comparaison rapide d'images
- découpage en plan séquence
- ...

Fin