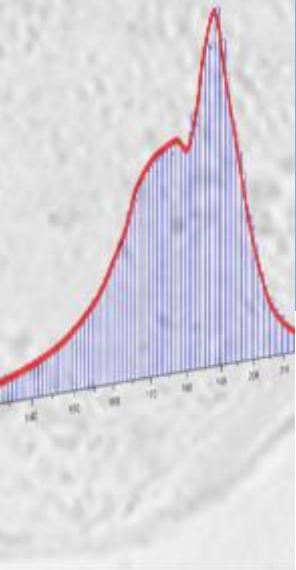




Cours de segmentation d'images

Partie 3 – Méthodes par histogrammes



Master M2TI - Paris V

2015-2016

John Chaussard

LAGA – Université Paris 13

chaussard@math.univ-paris13.fr

Les histogrammes en niveaux de gris

Construire un histogramme en niveaux de gris

L'histogramme d'une image en niveaux de gris consiste à **recenser toutes les valeurs des pixels** d'une image.

1	5	4	4
1	2	2	1
4	1	2	2
1	4	2	5

Par exemple, considérons cette image de 4x4 pixels, avec des valeurs allant de 1 à 5.

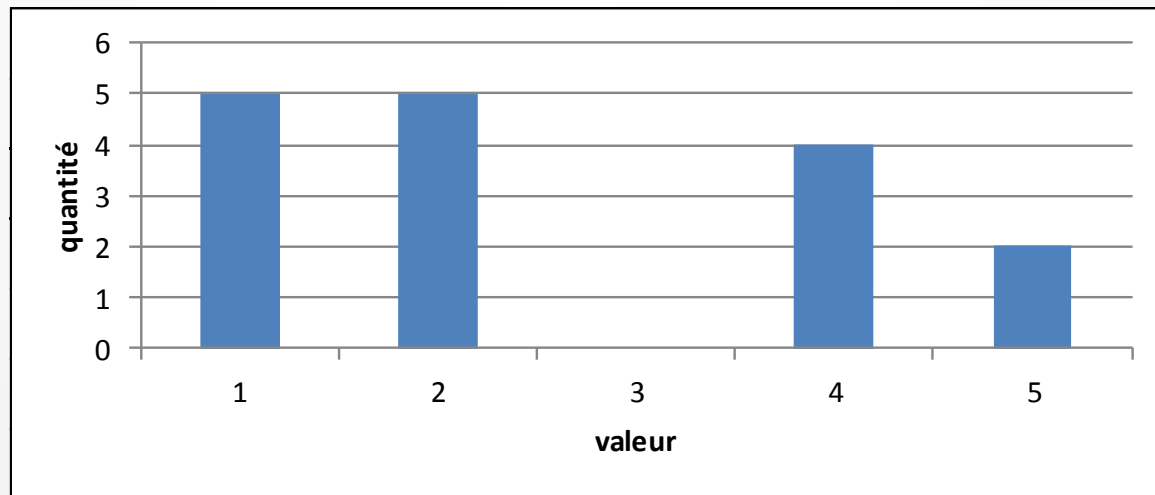
- Il y a :
- . 5 pixels de valeur 1
 - . 5 pixels de valeur 2
 - . 0 pixel de valeur 3
 - . 4 pixels de valeur 4
 - . 2 pixels de valeur 5

Représentation d'histogramme

On peut représenter les résultats à l'aide d'un tableau

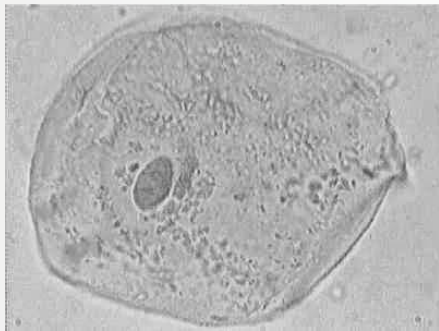
valeur	1	2	3	4	5
quantité	5	5	0	4	2

Ou bien à l'aide d'un graphe

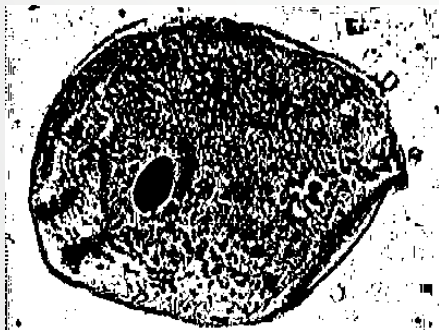


Représentation d'histogramme

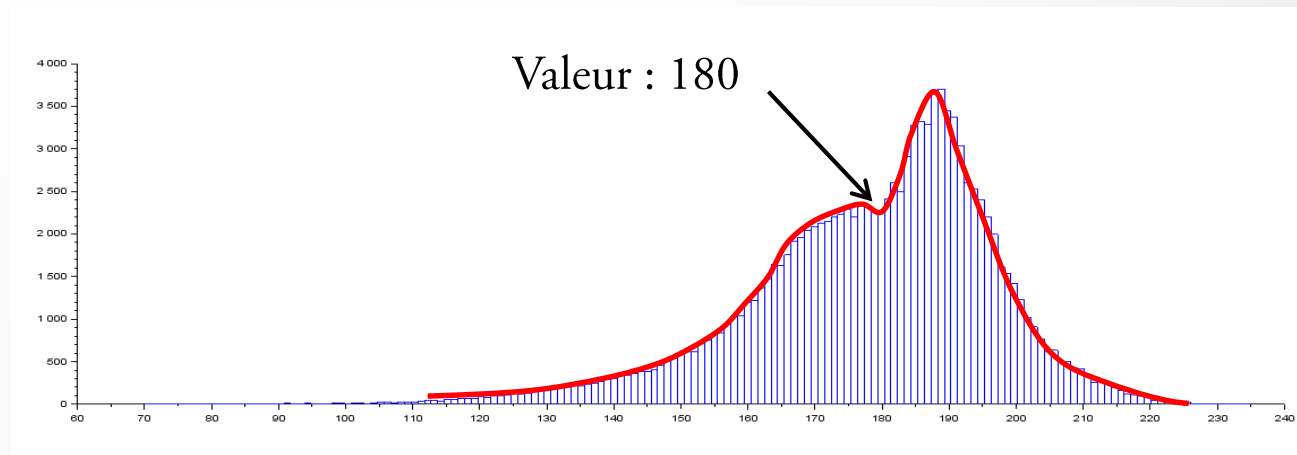
L'intérêt de l'histogramme est qu'il peut apporter de précieuses informations sur l'image.



^ Image originale



< Seuil à 180



Représentation d'histogramme

En général, on ne regarde pas si un pixel vaut une valeur précise, **mais s'il est dans un intervalle donné.**

1	5	4	4
1	2	2	1
4	1	2	2
1	4	2	5

Par exemple, on pourrait vouloir connaître le nombre de pixels dans les intervalles $[0;1]$, $[2;3]$ et $[4;5]$:

Il y a :

- . 5 pixels entre 0 et 1
- . 5 pixels entre 2 et 3
- . 6 pixels entre 4 et 5

Ceci permet d'étendre la notion d'histogrammes aux images à valeur réelles.

Représentation d'histogramme

En général, on **spécifie le nombre d'intervalles de valeurs que l'on souhaite étudier** (appelés *classes*, ou *bins de l'histogramme*) : les intervalles seront régulièrement espacés entre les valeurs extrémales de l'image.

Les valeurs des pixels de cette image vont de 3 à 14.

Un histogramme en **quatre classes** donne :

- . 6 pixels entre 3 et 5
- . 4 pixels entre 6 et 8
- . 4 pixels entre 9 et 11
- . 2 pixels entre 12 et 14

Un histogramme en **trois classes** donne :

- . 6 pixels entre 3 et 6
- . 6 pixels entre 7 et 10
- . 4 pixels entre 11 et 14

10	14	4	3
8	8	9	7
4	5	5	8
12	3	11	11

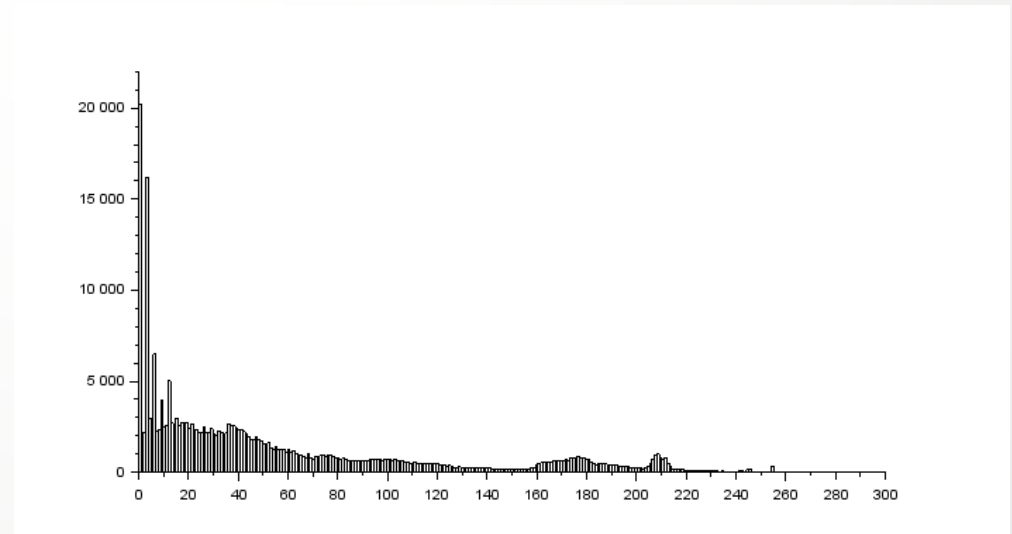
Egalisation d'histogramme (échauffement)

Egaliser un histogramme

Certaines images peuvent avoir une plage de valeurs réduite, rendant leur visualisation difficile. On souhaite trouver une **transformation permettant d'améliorer le contraste** de l'image.



Image sous exposée

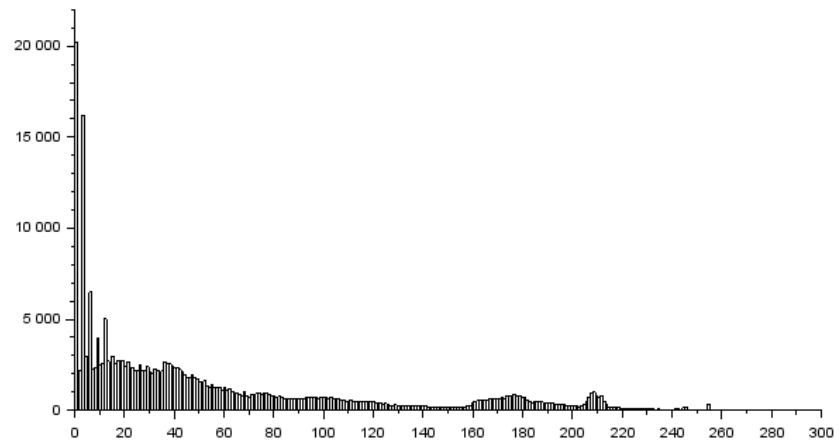


Histogramme de l'image

Egaliser un histogramme

Certaines images peuvent avoir une plage de valeurs réduite, rendant leur visualisation difficile. On souhaite trouver une **transformation permettant d'améliorer le contraste** de l'image.

Les niveaux de gris de l'image sont concentrés vers des valeurs basses...
 Il est nécessaire de corriger ce défaut.



Histogramme de l'image

Approche par le continu

Soient :

P l'image d'entrée (avec le mauvais contraste),

Q l'image de sortie (avec le contraste rehaussé),

N le nombre de pixels des deux images.

On pose :

H_P l'histogramme de l'image d'entrée,

H_Q l'histogramme de l'image de sortie.

$H_P(x)$ est le nombre de pixels de l'image d'entrée possédant le niveau de gris x .

$H_Q(y)$ est le nombre de pixels de l'image de sortie possédant le niveau de gris y .

Approche par le continu

On pose H_P l'histogramme de l'image d'entrée, et H_Q l'histogramme de l'image de sortie.

On souhaite trouver la transformation T , qui à un niveau de gris x de l'image d'entrée P associe un niveau de gris y de l'image de sortie q :

$$T(x)=y$$

Bref, T nous indique comment transformer les valeurs des pixels de P afin d'obtenir une « jolie » image Q .

Approche par le continu

Pour « faire plus simple », on imagine que **l'image est continue**, c'est à dire que c'est une application de \mathbb{R}^n dans \mathbb{R} .

On imagine que l'image P possède des valeurs allant de p_{\min} jusqu'à p_{\max} , tandis que l'image Q possèdera des valeurs allant de q_{\min} jusqu'à q_{\max} .

Approche par le continu

$H_P(x)$ nous donne le nombre de points, dans P , possédant le niveau de gris x .

Question : Que représente

$$\int_{P_{\min}}^x H_P(s) ds$$

Il s'agit du nombre de points de l'image P qui possèdent un niveau de gris inférieur ou égal à x .

Approche par le continu

$$\int_{p_{\min}}^x H_P(s) ds$$

est le nombre de points de l'image P qui possèdent un niveau de gris inférieur ou égal à x .

Question : comment écrire cela dans un cadre discret (une image de \mathbb{N}^n dans \mathbb{N}) ?

$$\sum_{s=p_{\min}}^x H_P(s)$$

Approche par le continu

Cadre continu

$$\int_{p_{\min}}^x H_P(s) ds$$

Cadre discret

$$\sum_{s=p_{\min}}^x H_P(s)$$

En général, pour passer d'un cadre continu à un cadre discret, on remplace les intégrales par des sommes...

Approche par le continu

On souhaite trouver la transformation T , qui à un niveau de gris x de l'image d'entrée P associe un niveau de gris y de l'image de sortie q :

$$T(x)=y$$

La transformation T devra préserver l'ordre des niveaux de gris, c'est à dire être croissante :

$$x_1 \leq x_2 \rightarrow T(x_1) \leq T(x_2)$$

On en déduit que, dans le continu, le nombre de points de l'image P possédant un niveau de gris inférieur ou égal à x_1 est égal au nombre de points de l'image Q possédant un niveau de gris inférieur ou égal à $T(x_1)$.

A la recherche de la transformation T

On en déduit que le nombre de points de l'image P possédant un niveau de gris inférieur ou égal à x_1 est égal au nombre de points de l'image Q possédant un niveau de gris inférieur ou égal à $T(x_1)$.

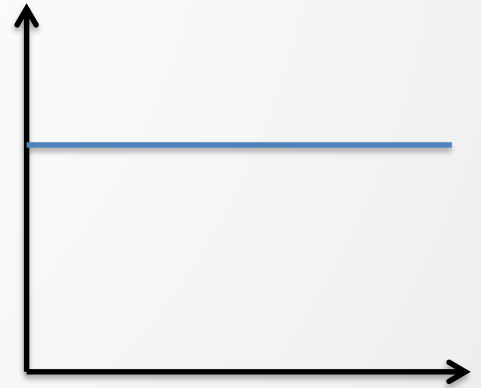
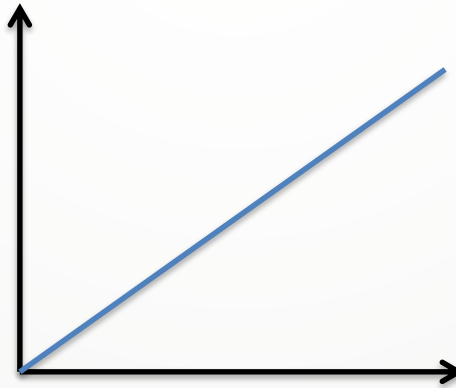
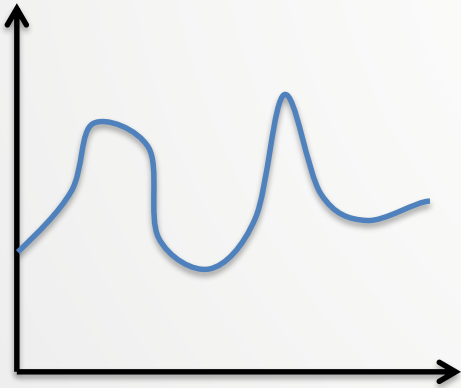
Question : Comment exprimer cette relation en termes d'histogrammes ?

$$\int_{p_{\min}}^x H_P(s) ds = \int_{q_{\min}}^{T(x)} H_Q(u) du$$

A la recherche de la transformation T

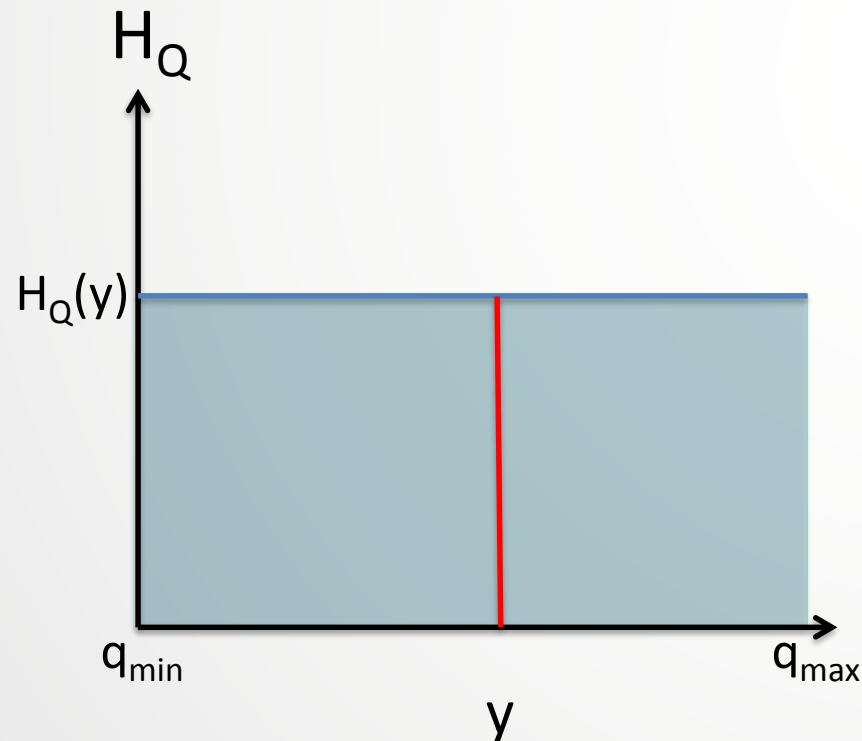
On souhaite obtenir un histogramme égalisé/uniformisé.

Question : Lequel de ces histogrammes est égalisé ?



A la recherche de la transformation T

Question : Que vaut $H_Q(y)$?



On pose :

$$\int_{q_{\min}}^{q_{\max}} H_Q(u) du = N$$

$$(q_{\max} - q_{\min}) H_Q(y) = N$$

$$H_Q(y) = \frac{N}{q_{\max} - q_{\min}}$$

A la recherche de la transformation T

Question : trouver l'expression de $T(x)$ en fonction de x , en considérant que

$$\int_{q_{\min}}^{T(x)} H_Q(u) du = \int_{p_{\min}}^x H_P(s) ds$$

$$H_Q(y) = \frac{N}{(q_{\max} - q_{\min})}$$

A la recherche de la transformation T

$$\int_{q_{\min}}^{T(x)} H_Q(u) du = \int_{p_{\min}}^x H_P(s) ds$$

$$\int_{q_{\min}}^{T(x)} \frac{N}{(q_{\max} - q_{\min})} du = \int_{p_{\min}}^x H_P(s) ds$$

$$N \frac{T(x) - q_{\min}}{q_{\max} - q_{\min}} = \int_{p_{\min}}^x H_P(s) ds$$

$$T(x) = \frac{q_{\max} - q_{\min}}{N} \int_{p_{\min}}^x H_P(s) ds + q_{\min}$$

A la recherche de la transformation T

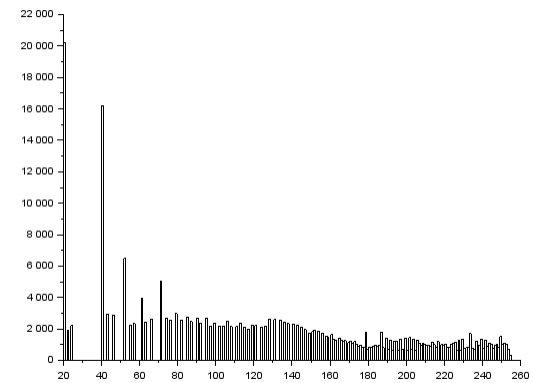
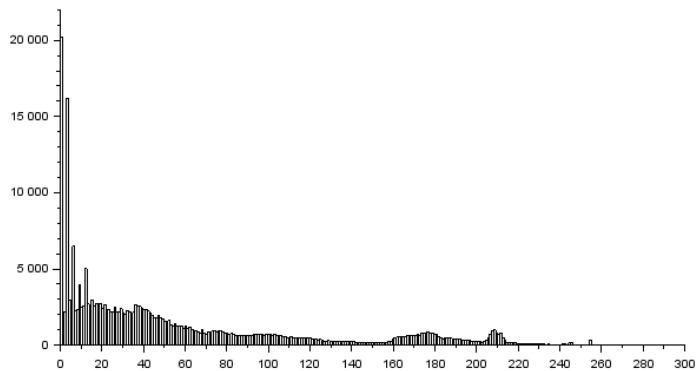
On a donc :

$$T(x) = \frac{q_{\max} - q_{\min}}{N} \int_{p_{\min}}^x H_P(s) ds + q_{\min}$$

Ce qui donne dans le discret :

$$T(x) = \frac{q_{\max} - q_{\min}}{N} \sum_{s=p_{\min}}^x H_P(s) + q_{\min}$$

Résultat



Seuil automatique d'Otsu

Qu'est-ce qu'un seuil

Réaliser un seuil (threshold en anglais) d'une image consiste à placer tous les pixels au-dessus d'un certain niveau de gris T à la valeur blanc (255), et tous les autres à la valeur noir (0).



Seuil à 70



Seuil à 220



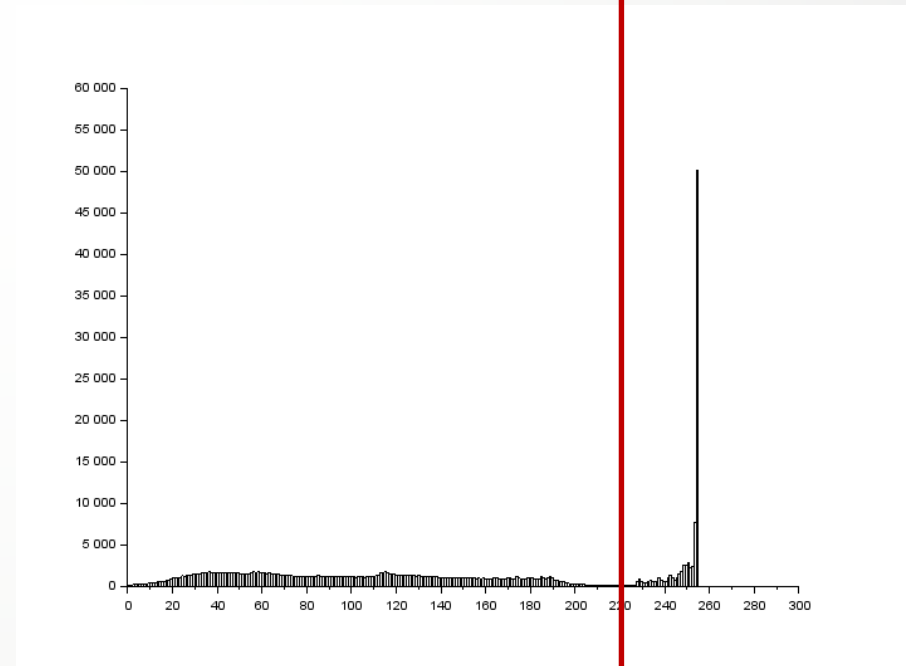
Seuil à 150

Ne pas se fier à son instinct

En examinant l'histogramme d'une image, **on peut parfois voir un seuil « évident »** apparaître :



Seuil à 220

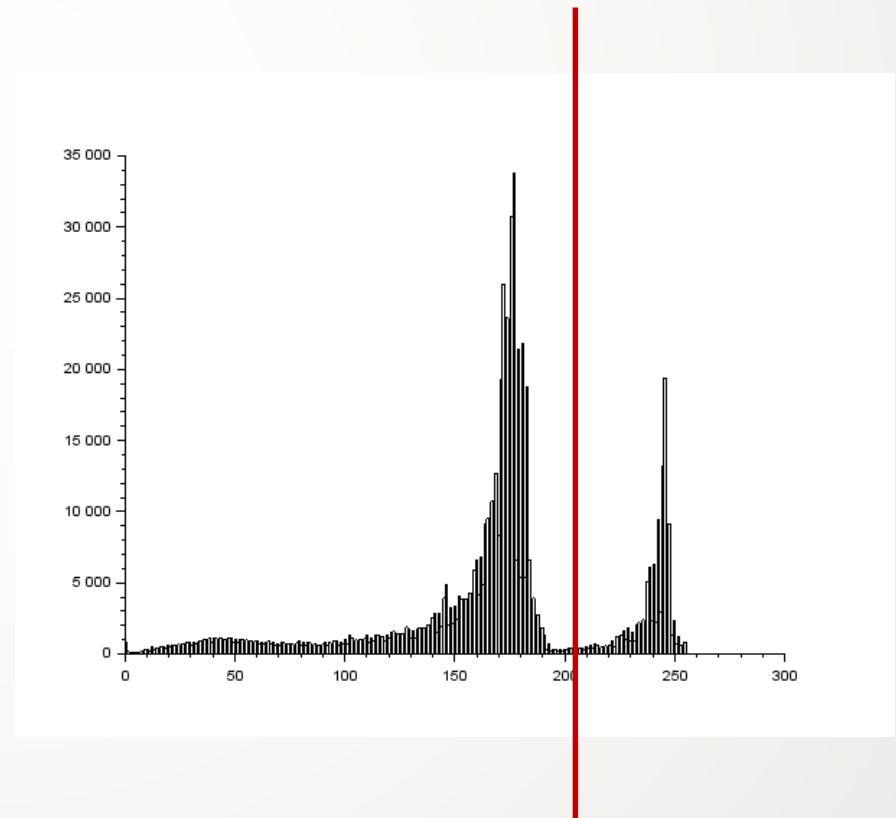


Ne pas se fier à son instinct

En examinant l'histogramme d'une image, **on peut parfois voir un seuil « évident »** apparaître :



Seuil à 205



Ne pas se fier à son instinct

En examinant l'histogramme d'une image, **on peut parfois voir un seuil « évident »** apparaître :

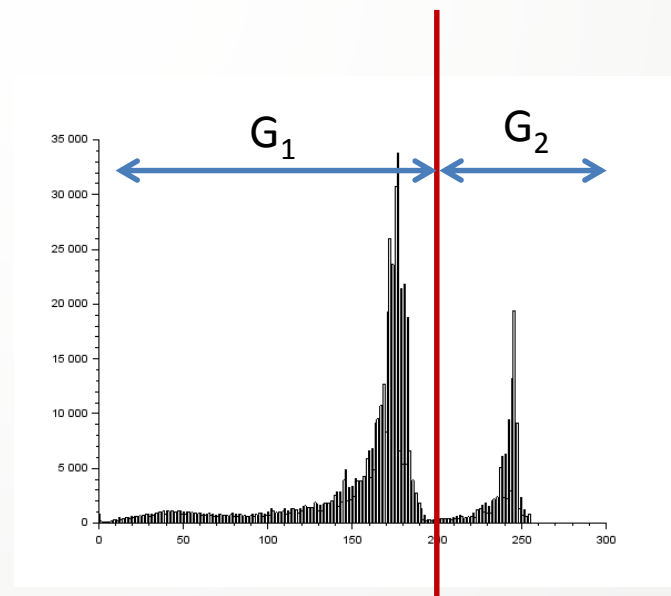


L'examen visuel du seuil, même s'il fait apparaître des seuils évidents, ne fait pas forcément apparaître des seuils intéressants.

Trouver un seuil automatiquement

Une méthode permet de trouver un seuil « sympathique » automatiquement : elle a été proposée par **Nobuyuki Otsu en 1979**.

Quand on choisit un seuil T , on divise l'histogramme H de l'image en deux groupes : G_1 et G_2

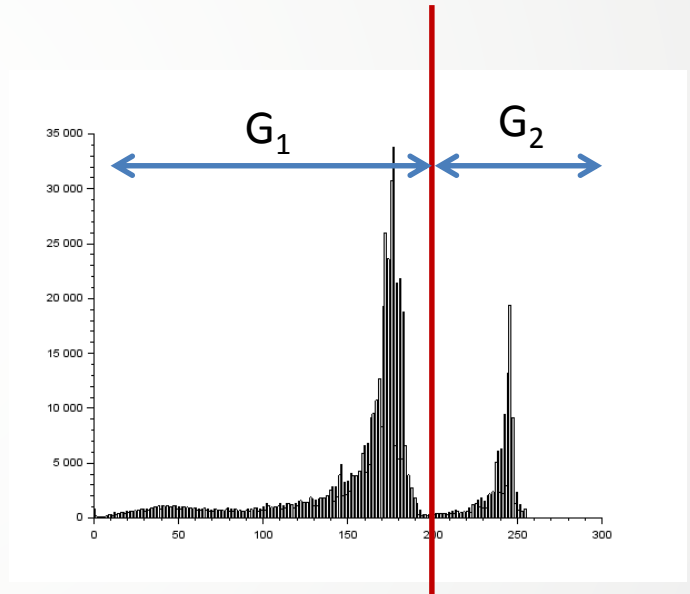


Trouver un seuil automatiquement

Quand on choisit un seuil T , on divise l'histogramme H de l'image en deux groupes : G_1 et G_2

On calcule la variance V_1 et V_2 de ces deux groupes.

On cherche le seuil T qui produit les deux groupes dont les variances sont les plus faibles possibles.



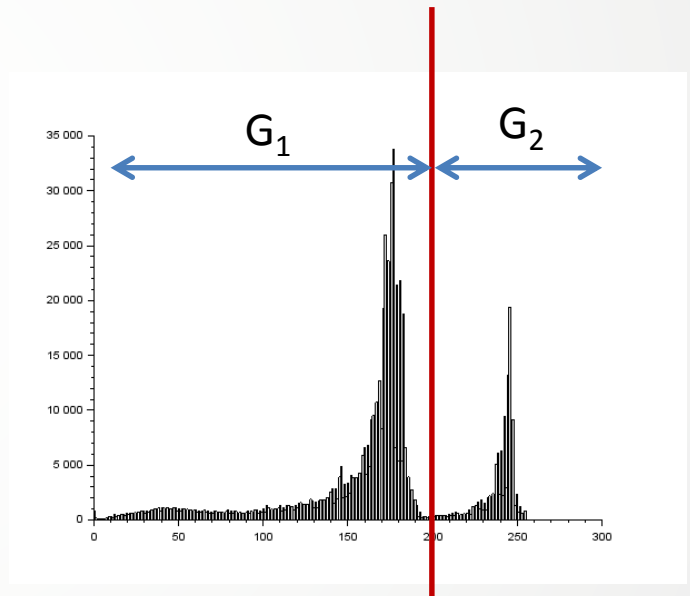
Cela revient à produire deux groupes où « les pixels se ressemblent au sein d'un même groupe » : cela s'appelle **minimiser la variance intra classe**.

Trouver un seuil automatiquement (2)

Quand on choisit un seuil T , on divise l'histogramme H de l'image en deux groupes : G_1 et G_2

On calcule la variance V_1 et V_2 de ces deux groupes.

On peut aussi chercher le seuil T qui produit les deux groupes dont les moyennes sont les plus éloignées.



Cela revient à produire deux groupes les plus dissemblables possible : cela s'appelle **maximiser la variance inter classe**.

Trouver un seuil automatiquement (3)

L'**analyse discriminante** est une technique statistique où l'on cherche à regrouper des observations dans des groupes.

Si on pose V_{inter} la variance inter-classe, et V_{intra} la variance intra classe, on cherche, en analyse discriminante, à maximiser la quantité :

$$\lambda = \frac{(V_{inter})}{(V_{intra})}$$

C'est de cette méthode qu'Otsu s'est inspiré pour proposer son seuil automatique en 1979.

Que maximiser/minimiser ?

Otsu a proposé que, pour maximiser λ , il suffisait de maximiser V_{inter} ou de minimiser V_{intra} .

Si on pose V la variance totale de l'image, on a en effet la relation suivante :

$$V = V_{inter} + V_{intra}$$

V étant fixe, maximiser V_{inter} revient à minimiser V_{intra} , et inversement.

Démontrons cette relation...

Trouver un seuil automatiquement

Posons N le nombre de pixels de l'image I .

- On peut exprimer la **moyenne des pixels de I** comme étant

$$\mu = \frac{1}{N} \sum_{x \in I} I(x)$$

- On peut exprimer la **moyenne des pixels du groupe j** comme étant

$$\mu_j = \frac{1}{|G_j|} \sum_{x \in G_j} I(x)$$

Trouver un seuil automatiquement

Posons N le nombre de pixels de l'image I .

- On peut exprimer la **variance totale des pixels de I** comme étant

$$V = \frac{1}{N} \sum_{x \in I} (I(x) - \mu)^2$$

- On peut exprimer la **variance des pixels du groupe j** comme étant

$$V_j = \frac{1}{|G_j|} \sum_{x \in G_j} (I(x) - \mu_j)^2$$

Trouver un seuil automatiquement

Posons N le nombre de pixels de l'image I . On se place dans le cas de deux groupes.

- La **variance inter classe** s'exprime comme

$$V_{inter} = \frac{|G_1|}{N} V_1 + \frac{|G_2|}{N} V_2$$

- La **variance intra classe** s'exprime comme

$$V_{intra} = \frac{|G_1|}{N} \frac{|G_2|}{N} (\mu_1 - \mu_2)^2$$

Exprimer le problème différemment

On peut décomposer la variance selon les deux groupes de pixels :

$$V = \frac{1}{N} \sum_{x \in I} (x - \bar{x})^2 = \frac{1}{N} \left(\sum_{x_1 \leq T} (x_1 - \bar{x})^2 + \sum_{x_2 > T} (x_2 - \bar{x})^2 \right)$$

On pose :

$$\bar{x}_{<} = \frac{1}{|T_{<}|} \sum_{x_1 \leq T} x_1$$

$$\bar{x}_{>} = \frac{1}{|T_{>}|} \sum_{x_2 > T} x_2$$

La moyenne des pixels des groupes $T_{<}$ et $T_{>}$.

Exprimer le problème différemment

$$\begin{aligned}
 \sum_{x_1 \leq T} (x_1 - \bar{x})^2 &= \sum_{x_1 \leq T} ((x_1 - \bar{x}_{<}) + (\bar{x}_{<} - \bar{x}))^2 \\
 &= \sum_{x_1 \leq T} ((x_1 - \bar{x}_{<})^2 + (\bar{x}_{<} - \bar{x})^2 + 2(x_1 - \bar{x}_{<})(\bar{x}_{<} - \bar{x})) \\
 &= \sum_{x_1 \leq T} (x_1 - \bar{x}_{<})^2 + \sum_{x_1 \leq T} (\bar{x}_{<} - \bar{x})^2 + 2 \sum_{x_1 \leq T} (x_1 - \bar{x}_{<})(\bar{x}_{<} - \bar{x}) \\
 &= \sum_{x_1 \leq T} (x_1 - \bar{x}_{<})^2 + \sum_{x_1 \leq T} (\bar{x}_{<} - \bar{x})^2 + 2(\bar{x}_{<} - \bar{x}) \sum_{x_1 \leq T} (x_1 - \bar{x}_{<})
 \end{aligned}$$

Exprimer le problème différemment

$$= \sum_{x_1 \leq T} (x_1 - \bar{x}_<) ^2 + \sum_{x_1 \leq T} (\bar{x}_< - \bar{x}) ^2 + 2(\bar{x}_< - \bar{x}) \sum_{x_1 \leq T} (x_1 - \bar{x}_<)$$

$$= \sum_{x_1 \leq T} (x_1 - \bar{x}_<) ^2 + \sum_{x_1 \leq T} (\bar{x}_< - \bar{x}) ^2$$

On a donc :

$$V = \frac{1}{N} \left(\sum_{x_1 \leq T} (x_1 - \bar{x}) ^2 + \sum_{x_2 > T} (x_2 - \bar{x}) ^2 \right)$$

$$= \frac{1}{N} \left(\sum_{x_1 \leq T} (x_1 - \bar{x}_<) ^2 + \sum_{x_2 > T} (x_2 - \bar{x}_>) ^2 + \sum_{x_1 \leq T} (\bar{x}_< - \bar{x}) ^2 + \sum_{x_2 > T} (\bar{x}_> - \bar{x}) ^2 \right)$$

Exprimer le problème différemment

$$V = \frac{1}{N} \left(\sum_{x1 \leq T} (x_1 - \bar{x}_{<})^2 + \sum_{x2 > T} (x_2 - \bar{x}_{>})^2 + \sum_{x1 \leq T} (\bar{x}_{<} - \bar{x})^2 + \sum_{x2 > T} (\bar{x}_{>} - \bar{x})^2 \right)$$

$$= \frac{1}{N} \left(|T_{<}| \times V_{T_{<}} + |T_{>}| \times V_{T_{>}} + \sum_{x1 \leq T} (\bar{x}_{<} - \bar{x})^2 + \sum_{x2 > T} (\bar{x}_{>} - \bar{x})^2 \right)$$

$$= \frac{1}{N} \left(|T_{<}| \times V_{T_{<}} + |T_{>}| \times V_{T_{>}} + |T_{<}| (\bar{x}_{<} - \bar{x})^2 + |T_{>}| (\bar{x}_{>} - \bar{x})^2 \right)$$

Exprimer le problème différemment

$$|T_{<}|(\bar{x}_{<} - \bar{x})^2 + |T_{>}|(\bar{x}_{>} - \bar{x})^2$$

$$= |T_{<}|\bar{x}_{<}^2 + |T_{>}|\bar{x}_{>}^2 + N\bar{x}^2 - 2\bar{x}(|T_{<}|\bar{x}_{<} + |T_{>}|\bar{x}_{>})$$

$$= |T_{<}|\bar{x}_{<}^2 + |T_{>}|\bar{x}_{>}^2 - N\bar{x}^2$$

$$= |T_{<}|\bar{x}_{<}^2 + |T_{>}|\bar{x}_{>}^2 - N \left(\frac{|T_{<}|\bar{x}_{<} + |T_{>}|\bar{x}_{>}}{N} \right)^2$$

$$= |T_{<}|\bar{x}_{<}^2 + |T_{>}|\bar{x}_{>}^2 - \frac{|T_{<}|^2\bar{x}_{<}^2 + |T_{>}|^2\bar{x}_{>}^2 + 2|T_{<}||T_{>}|\bar{x}_{<}\bar{x}_{>}}{N}$$

Exprimer le problème différemment

$$\text{Or, } |T_{<}| + |T_{>}| = N$$

$$= |T_{<}|\overline{x_{<}}^2 + |T_{>}|\overline{x_{>}}^2 - \frac{|T_{<}|^2\overline{x_{<}}^2 + |T_{>}|^2\overline{x_{>}}^2 + 2|T_{<}||T_{>}|\overline{x_{<}}\overline{x_{>}}}{N}$$

$$= \frac{N|T_{<}|\overline{x_{<}}^2 - |T_{<}|^2\overline{x_{<}}^2}{N} + \frac{N|T_{>}|\overline{x_{>}}^2 - |T_{>}|^2\overline{x_{>}}^2}{N} - \frac{2|T_{<}||T_{>}|\overline{x_{<}}\overline{x_{>}}}{N}$$

$$= \frac{|T_{<}||T_{>}|\overline{x_{<}}^2}{N} + \frac{|T_{<}||T_{>}|\overline{x_{>}}^2}{N} - \frac{2|T_{<}||T_{>}|\overline{x_{<}}\overline{x_{>}}}{N}$$

Exprimer le problème différemment

La variance totale s'écrit donc

$$= \frac{|T_{<}||T_{>}|\bar{x}_{<}^2}{N} + \frac{|T_{<}||T_{>}|\bar{x}_{>}^2}{N} - \frac{2|T_{<}||T_{>}|\bar{x}_{<}\bar{x}_{>}}{N}$$

$$= \frac{|T_{<}||T_{>}|}{N} (\bar{x}_{<} - \bar{x}_{>})^2$$

$$V = \frac{|T_{<}|}{N} V_{T_{<}} + \frac{|T_{>}|}{N} V_{T_{>}} + \frac{|T_{<}||T_{>}|}{N^2} (\bar{x}_{<} - \bar{x}_{>})^2$$

Résultat

La variance totale s'écrit donc

$$V = \frac{|G_1|}{N} V_{1<} + \frac{|G_2|}{N} V_2 + \frac{|G_1||G_2|}{N^2} (\mu_1 - \mu_2)^2$$

Valeur fixe

On peut aussi maximiser cette quantité, qui est la variance inter-classe

Variance intra-classe que l'on cherche à minimiser

Algorithme

Il est un peu plus simple de calculer la variance inter classe que la variance intra classe, car la première n'est basée que sur des moyennes.

Pour deux classes, on **implémentera cet algorithme** en testant tous les seuils possibles et en retenant celui qui maximise la variance inter classe.

Algorithme

Soit H l'histogramme de l'image I , dont les valeurs des pixels vont de 0 à M .

La phase d'initialisation

```
mu <= moyenne des pixels de l'image I

g1 = H(0)           //Le nombre de pixels de valeur inf ou égale à 0
mu1 <= 0           //La moyenne des pixels de valeur inf ou égale à 0

g2 = N - H(0)      //Le nombre de pixels de valeur sup à 0
mu2 <= mu*N/|G2|   //La moyenne des pixels de valeur sup à 0

best_var_inter = (g1 * g2) / N2 * (mu1 - mu2)2
best_seuil = 0
```

Algorithme

Soit H l'histogramme de l'image I , dont les valeurs des pixels vont de 0 à M .

Le déroulement

Pour tous les seuils T allant de 1 à M (exclu)

$$\mu_1 = \mu_1 * g_1 + T * H(T)$$

$$\mu_2 = \mu_2 * g_2 - T * H(T)$$

$$g_1 = g_1 + H(T)$$

$$g_2 = g_2 - H(T)$$

$$\mu_1 = \mu_1 / g_1$$

$$\mu_2 = \mu_2 / g_2$$

Si $(g_1 * g_2) / N^2 * (\mu_1 - \mu_2)^2 > \text{best_var}$

$$\text{best_var} = (g_1 * g_2) / N^2 * (\mu_1 - \mu_2)^2$$

$$\text{best_seuil} = T$$

Résultat



Seuil à 154

Résultat



Seuil à 130

Plus de deux classes

Dans le cas de k classes, on chercherait les classes qui maximisent :

$$\sum_{\substack{i=1 \\ j=1 \\ i \neq j}}^k \frac{|G_i|}{N} \frac{|G_j|}{N} (\mu_i - \mu_j)^2$$

C'est une **formule assez peu pratique**, car chaque classe contribue à différents termes de la somme. De plus, chaque terme dépend de deux classes, et non d'une seule : on ne peut pas aisément séparer les termes par classe.

Plus de deux classes

On retiendra donc cette formule équivalente (obtenue lors de la démonstration précédente) que l'on cherchera à maximiser :

$$\sum_{i=1}^k \frac{|G_i|}{N} (\mu_i - \mu)^2$$

Chaque terme de la somme dépend d'une seule classe, et chaque classe agit sur un seul terme de la somme. On peut maximiser cette valeur grâce à un algorithme efficace.

Plus de deux classes : algorithme sur un exemple

Pour comprendre l'algorithme, imaginons que l'on ait six classes en tout, donc que l'on cherche cinq seuils T_1 , T_2 , T_3 , T_4 et T_5 tels que

$$T_1 < T_2 < T_3 < T_4 < T_5$$

La classe 1 sera composée des pixels entre 0 et T_1-1

La classe 2 sera composée des pixels entre T_1 et T_2-1 ,

La classe 3 sera composée des pixels entre T_2 et T_3-1 ,

La classe 4 sera composée des pixels entre T_3 et T_4-1 ,

La classe 5 sera composée des pixels entre T_4 et T_5-1 ,

et la classe 6 sera composée des pixels de valeurs supérieure ou égale à T_5

Plus de deux classes : algorithme sur un exemple

Commençons par chercher la classe 1... Pour chaque seuil possible T_1 , on calcule la contribution de la classe 1 au terme général à maximiser, c'est à dire

$$\text{Contrib1}(T_1) = \frac{|G_1|}{N} (\mu_1 - \mu)^2$$

on rappelle que $|G_1|$ et μ_1 dépendent de T_1 .

On peut donc réécrire cette formule comme

$$\text{Contrib1}(T_1) = \frac{|G_1(T_1)|}{N} (\mu_1(T_1) - \mu)^2$$

On n'a pas trouvé le seuil T_1 , mais on connaît sa contribution pour chaque valeur de T_1 possible.

Plus de deux classes : algorithme sur un exemple

Pour chaque seuil possible T_2 , on calcule la contribution de la classe 2 au terme général à maximiser.

Cette contribution dépend de T_2 mais aussi de T_1 : on recherchera donc, pour chaque seuil T_2 , le meilleur seuil T_1 possible.

Quel est le meilleur seuil T_1 étant donné une valeur de T_2 ? C'est le seuil qui maximise la contribution de la classe 1 et de la classe 2 :

$$BestT_1(T_2) = \operatorname{argmax}_{T_1 < T_2} \left(\frac{|G_2(T_2, T_1)|}{N} (\mu_2(T_2, T_1) - \mu)^2 + \frac{|G_1(T_1)|}{N} (\mu_{1(T_1)} - \mu)^2 \right)$$

On calcule, pour chaque T_2 possible, le meilleur T_1 pour maximiser la contribution de la classe 2 et de la classe 1 ensemble au terme général.

Plus de deux classes : algorithme sur un exemple

Pour chaque seuil possible T_2 , on calcule la contribution de la classe 2 au terme général à maximiser.

Cette contribution dépend de T_2 mais aussi de T_1 : on recherchera donc, pour

Le nombre de pixels de la classe 2, c'est-à-dire ayant une valeur entre T_1 et T_2-1

La moyenne des pixels de la classe 2

$$BestT_1(T_2) = \operatorname{argmax}_{T_1 < T_2} \left(\frac{|G_2(T_2, T_1)|}{N} (\mu_2(T_2, T_1) - \mu)^2 + \frac{|G_1(T_1)|}{N} (\mu_{1(T_1)} - \mu)^2 \right)$$

On calcule, pour chaque T_2 possible, le meilleur T_1 pour maximiser la contribution de la classe 2 et de la classe 1 ensemble au terme général.

Plus de deux classes : algorithme sur un exemple

On peut réécrire cette formule :

$$BestT_1(T_2) = \underset_{T_1 < T_2}{\operatorname{argmax}_{T_1}} \left(\frac{|G_2(T_2, T_1)|}{N} (\mu_2(T_2, T_1) - \mu)^2 + \operatorname{Contrib1}(T_1) \right)$$

On a donc, pour chaque seuil T_2 , le meilleur seuil T_1 pour maximiser la contribution de la classe 2 et de la classe 1 au terme général.

Pour chaque seuil T_2 , on enregistre cette « meilleure » contribution :

$$\operatorname{Contrib2}(T_2) = \frac{|G_2(T_2, BestT_1(T_2))|}{N} (\mu_2(T_2, BestT_1(T_2)) - \mu)^2 + \operatorname{Contrib1}(BestT_1(T_2))$$

Plus de deux classes : algorithme sur un exemple

On recommence la même chose pour la classe 3, c'est à dire que l'on cherche le seuil T_3 qui maximise la contribution de la classe 3, de la classe 2 et de la classe 1 au terme général.

On cherche donc, pour chaque seuil T_3 , les meilleurs seuils T_2 et T_1 possibles :

$$BestT_2(T_3) = \underset{\substack{T_2, T_1 \\ T_2 < T_3 \\ T_1 < T_2}}{\operatorname{argmax}} \left(\begin{array}{l} \frac{|G_3(T_3, T_2)|}{N} (\mu_3(T_3, T_2) - \mu)^2 \\ + \\ \frac{|G_2(T_2, T_1)|}{N} (\mu_2(T_2, T_1) - \mu)^2 \\ + \\ \frac{|G_1(T_1)|}{N} (\mu_1(T_1) - \mu)^2 \end{array} \right)$$

Il faudrait, pour trouver le meilleur T_2 pour un T_3 donné, faire varier à la fois T_1 et T_2 ? Cela risque de donner un algorithme assez lent...

Plus de deux classes : algorithme sur un exemple

On recommence la même chose pour la classe 3, c'est à dire que l'on cherche le seuil T_3 qui maximise la contribution de la classe 3, de la classe 2 et de la classe 1 au terme général.

On cherche donc, pour chaque seuil T_3 , les meilleurs seuils T_2 et T_1 possibles :

$$BestT_2(T_3) = \underset{\substack{T_2, T_1 \\ T_2 < T_3 \\ T_1 < T_2}}{\operatorname{argmax}} \left(\begin{array}{c} \frac{|G_3(T_3, T_2)|}{N} (\mu_3(T_3, T_2) - \mu)^2 \\ + \\ \frac{|G_2(T_2, T_1)|}{N} (\mu_2(T_2, T_1) - \mu)^2 \\ + \\ \frac{|G_1(T_1)|}{N} (\mu_1(T_1) - \mu)^2 \end{array} \right)$$

Ceci est la contribution, étant donné T_2 , de la classe 2 et 1 au terme général...

Sa plus forte valeur est $Contrib2(T_2)$, calculé à l'étape précédente

On n'a donc pas besoin de faire varier T_1 car on a déjà calculé le meilleur T_1 pour un T_2 donné, et on peut réécrire le terme plus simplement... 59

Plus de deux classes : algorithme sur un exemple

On recommence la même chose pour la classe 3, c'est à dire que l'on cherche le seuil T_3 qui maximise la contribution de la classe 3, de la classe 2 et de la classe 1 au terme général.

On cherche donc, pour chaque seuil T_3 , les meilleurs seuils T_2 et T_1 possibles :

$$BestT_2(T_3) = \underset_{T_2 < T_3}{\operatorname{argmax}_{T_2}} \left(\begin{array}{c} \frac{|G_3(T_3, T_2)|}{N} (\mu_3(T_3, T_2) - \mu)^2 \\ + \\ \operatorname{Contrib}(T_2) \end{array} \right)$$

$$\operatorname{Contrib}3(T_3) = \frac{|G_3(T_3, BestT_2(T_3))|}{N} (\mu_3(T_3, BestT_2(T_3)) - \mu)^2$$

Plus de deux classes : algorithme sur un exemple

On recommence la même chose pour la classe 4 ; on cherche, pour chaque T_4 , le meilleur T_3 :

$$BestT_3(T_4) = \underset{\substack{T_3 \\ T_3 < T_4}}{\operatorname{argmax}} \left(\begin{array}{c} \frac{|G_4(T_4, T_3)|}{N} (\mu_4(T_4, T_3) - \mu)^2 \\ + \\ Contrib(T_3) \end{array} \right)$$

$$Contrib4(T_4) = \frac{|G_4(T_4, BestT_3(T_4))|}{N} (\mu_4(T_4, BestT_3(T_4)) - \mu)^2$$

Plus de deux classes : algorithme sur un exemple

Le dernier seuil est un petit peu particulier, car il permet de « fixer » deux classes :

$$BestT_4(T_5) = \underset{\substack{T_4 \\ T_4 < T_5}}{\operatorname{argmax}} \left(\begin{array}{c} \frac{|G_6(T_5)|}{N} (\mu_6(T_5) - \mu)^2 \\ + \\ \frac{|G_5(T_5, T_4)|}{N} (\mu_5(T_5, T_4) - \mu)^2 \\ + \\ \operatorname{Contrib}(T_4) \end{array} \right)$$

Plus de deux classes : algorithme sur un exemple

On retrouve tous les seuils en cherchant d'abord le meilleur seuil T_5 :

$$T_5 = \underset{T_5}{\operatorname{argmax}} \operatorname{Best}T_4(T_5)$$

Puis, on retrouve les autres seuils facilement :

$$T_4 = \operatorname{Best}T_4(T_5)$$

$$T_3 = \operatorname{Best}T_3(T_4)$$

$$T_2 = \operatorname{Best}T_2(T_3)$$

$$T_1 = \operatorname{Best}T_1(T_2)$$

Plus de deux classes : algorithme sur un exemple

Pour l'implémentation de l'algorithme, il est souhaitable de pré-calculer, pour chaque $T_i < T_j$, la quantité

$$|G(T_j, T_i)|$$

Ceci peut-être réalisé à l'aide d'un histogramme cumulé HC de l'image I :

$H(x)$ = nombre de pixels de I ayant comme valeur x

$HC(x)$ = nombre de pixels de I ayant une valeur inférieure ou égale à x

On remarque alors que

$$|G(T_j, T_i)| = HC(T_j) - HC(T_i)$$

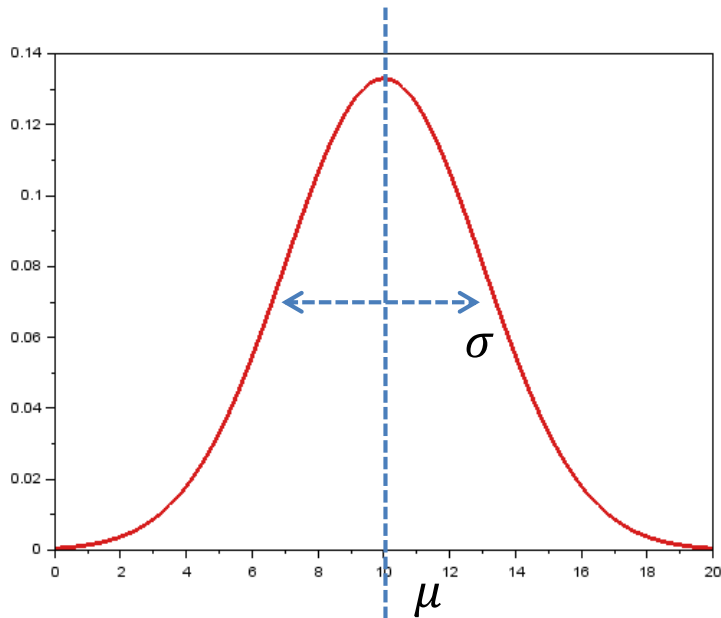
Par une méthode similaire, on peut pré-calculer $\mu(T_j, T_i)$

Mélange de gaussiennes (1D)

Rappel sur la gaussienne

La densité de probabilité de la loi normale s'écrit

$$f(x) = \frac{1}{\sigma * \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



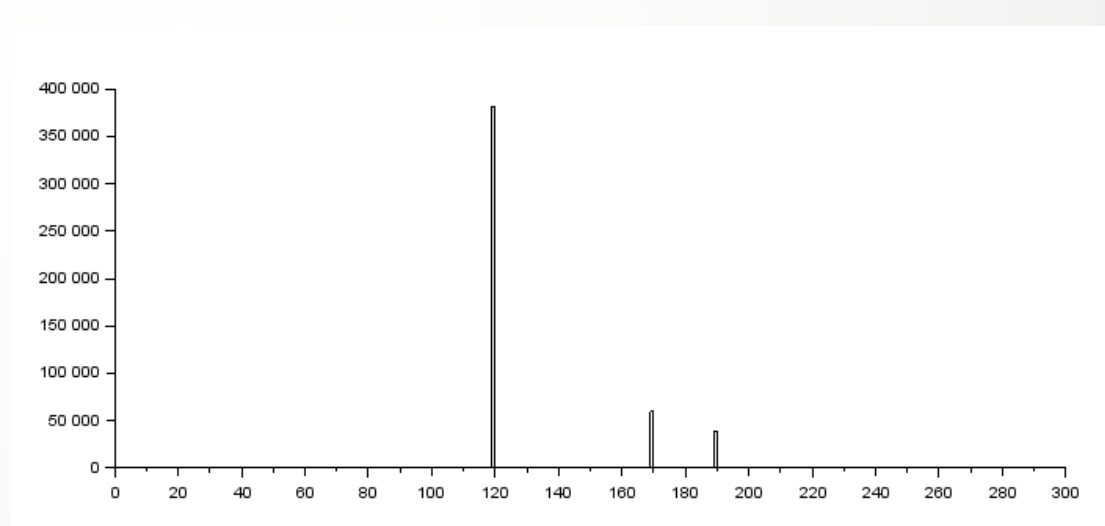
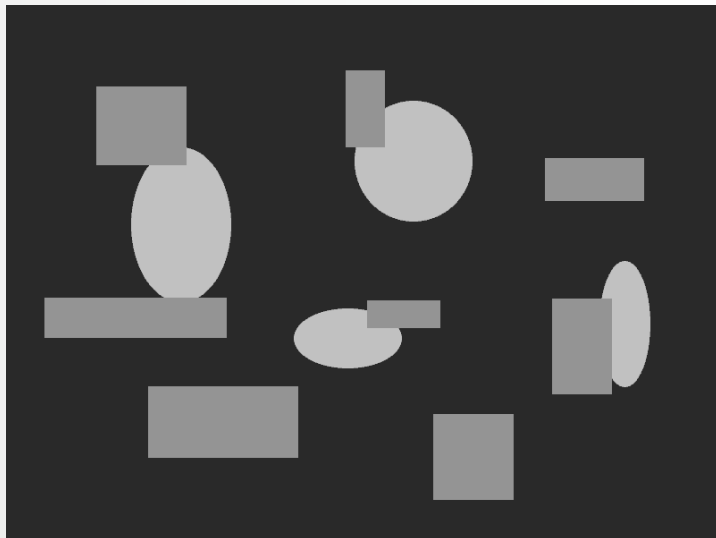
Une loi normale est caractérisée par deux paramètres : μ et σ .

Pour faire plus court, on écrira :

$$f(x) = \mathcal{N}_{\mu, \sigma}(x)$$

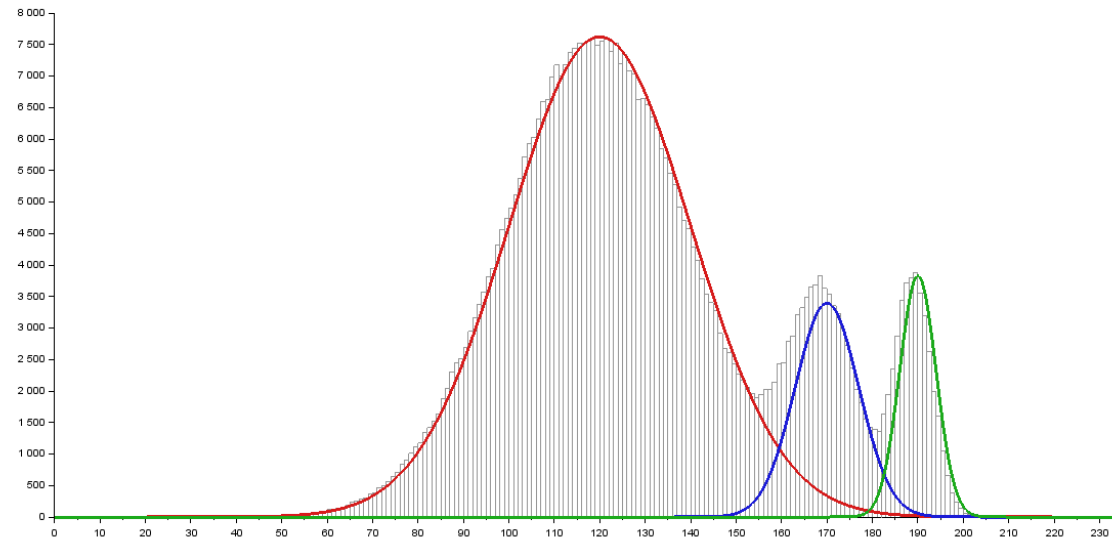
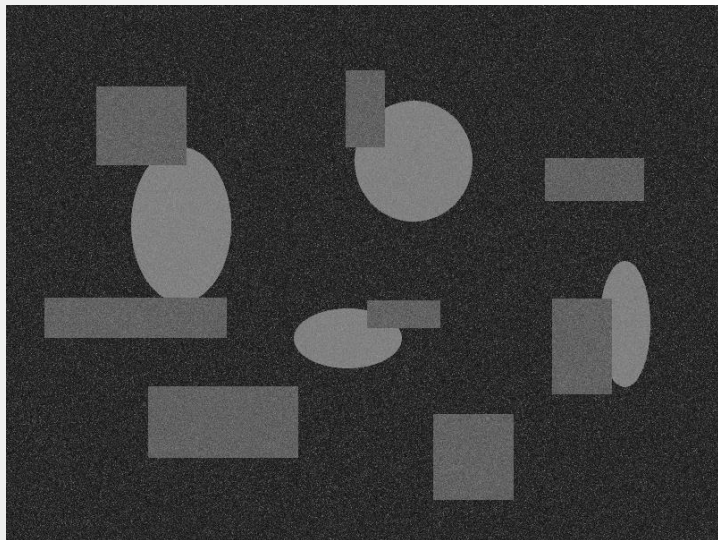
Cadre du problème

On imagine avoir une image avec trois éléments : des éléments circulaires clairs (190), des rectangles gris foncés (170) et un fond très foncé (120).



Cadre du problème

L'image est perturbée par du bruit gaussien, c'est-à-dire que chaque pixel verra sa valeur modifiée selon une loi gaussienne (différente selon la valeur du pixel) :



L'histogramme montre la présence de trois gaussiennes.

Cadre du problème

La question que l'on se pose est la suivante :

Peut-on retrouver les paramètres des trois gaussiennes à partir de l'image de sortie (ou de son histogramme) ?

On possède les données suivantes :

- . **La loi que suit chacun des processus aléatoires** que l'on essaye de reconstruire (ici, loi gaussienne pour chacun),
- . **Le nombre k de processus** (gaussiens) que l'on suspecte être à l'origine de notre observation,
- . **Une observation de la réalisation de ces processus** (ici, l'image bruitée Q).

Poser le problème

Si l'on considère que **l'image est le résultat d'un seul processus gaussien de paramètre mu et sigma** (la valeur de chaque pixel est le résultat d'un même tirage aléatoire gaussien), on a

$$p(Q(q) = n \mid \mu, \sigma) = \mathcal{N}_{\mu, \sigma}(n)$$

Etant donné que le niveau de gris observé du pixel q est $Q(q)$, on a

$$p(Q(q) \mid \mu, \sigma) = \mathcal{N}_{\mu, \sigma}(Q(q))$$

Poser le problème

L'image est en réalité le résultat de k processus gaussiens aléatoires, chacun avec une moyenne et une variance particulière.

On a alors dans ce cas :

$$\mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_k \end{bmatrix}, \sigma = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_k \end{bmatrix}$$

$$p(Q(q) | \mu, \sigma) = \sum_{i=1}^k \mathcal{N}_{\mu_i, \sigma_i}(Q(q))$$

La probabilité qu'un pixel q possède une valeur $Q(q)$ est donc la somme des probabilités que chaque processus gaussien ait pu générer q avec le niveau de gris $Q(q)$.

Poser le problème

Question : Quelle est la probabilité que $Q(q)$ soit une valeur entre moins l'infini et plus l'infini ?

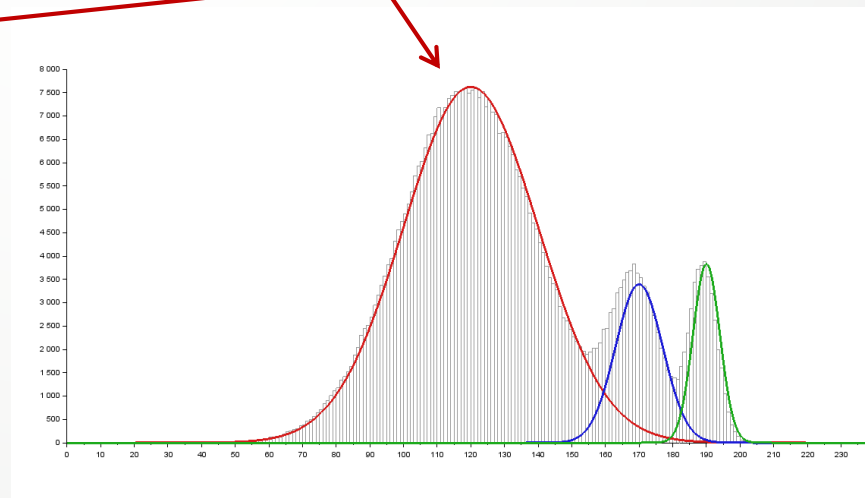
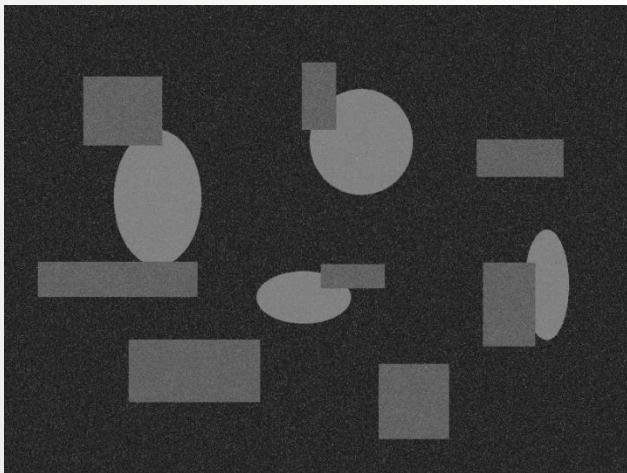
$$\begin{aligned} p(Q(q) \in \mathbb{R} \mid \mu, \sigma) &= \int_{-\infty}^{+\infty} \sum_{i=1}^k \mathcal{N}_{\mu_i, \sigma_i}(Q(q)) dQ(q) \\ &= \sum_{i=1}^k \int_{-\infty}^{+\infty} \mathcal{N}_{\mu_i, \sigma_i}(Q(q)) dQ(q) \\ &= \sum_{i=1}^k 1 = k > 1 \end{aligned}$$

Poser le problème

Une probabilité supérieure à 1 : ce n'est pas très bon...

En vérité, **chaque processus gaussien n'a pas le même poids** : comme on le voit sur l'exemple de l'image, chaque processus gaussien s'exerçant sur le fond de l'image se produit beaucoup plus souvent que le processus gaussien des rectangles ou des ellipses...

Processus de bruit sur le fond de l'image



Poser le problème

Une probabilité supérieure à 1 : ce n'est pas très bon...

En vérité, **chaque processus gaussien n'a pas le même poids** : comme on le voit sur l'exemple de l'image, chaque processus gaussien s'exerçant sur le fond de l'image se produit beaucoup plus souvent que le processus gaussien des rectangles ou des ellipses...

On a alors dans ce cas :

Le poids du processus gaussien i dans tout notre processus...

$$p(Q(q) | \mu, \sigma, \Pi) = \sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q))$$

avec

$$\sum_{i=1}^k \Pi_i = 1$$

Poser le problème

Question : Quelle est la probabilité que $Q(q)$ soit une valeur entre moins l'infini et plus l'infini ?

$$\begin{aligned} p(Q(q) \in \mathbb{R} \mid \mu, \sigma, \Pi) &= \int_{-\infty}^{+\infty} \sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q)) dQ(q) \\ &= \sum_{i=1}^k \Pi_i * \int_{-\infty}^{+\infty} \mathcal{N}_{\mu_i, \sigma_i}(Q(q)) dQ(q) \\ &= \sum_{i=1}^k \Pi_i = 1 \end{aligned}$$

Poser le problème

On a donc, pour un pixel q en particulier :

$$p(Q(q) \mid \mu, \sigma, \Pi) = \sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q))$$

Pour tous les pixels q_1, \dots, q_N de l'image, on a

$$p(Q \mid \mu, \sigma, \Pi) = \prod_{j=1}^N \sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q_j))$$

La probabilité d'observer l'image Q (l'image bruitée que l'on a)

Poser le problème

Notre problème consiste à trouver les $3k$ paramètres (k μ , k σ et k π) qui maximisent la probabilité d'observer l'image Q que l'on a...

$$p(Q \mid \mu, \sigma, \Pi) = \prod_{j=1}^N \sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q_j))$$

On souhaite maximiser ça

Ca s'appelle **maximiser la vraisemblance**, c'est-à-dire maximiser la probabilité d'observer ce que l'on observe bel et bien (car on imagine que ce que l'on observe, c'est-à-dire l'image bruitée, n'est pas le fruit d'un immense coup de chance)...

Poser le problème

Pour plus de simplicité dans les calculs (si si !), on souhaiterait se débarrasser du produit au début de l'expression et le transformer en somme...

$$p(Q | \mu, \sigma, \Pi) = \prod_{j=1}^N \sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q_j))$$

Question : comment transformer notre expression afin de se débarrasser du produit ?

Le logarithme étant croissant, maximiser cela revient à maximiser cela

On étudie le logarithme de la probabilité :

$$\ln(p(Q | \mu, \sigma, \Pi)) = \sum_{j=1}^N \ln \left(\sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q_j)) \right)$$

Poser le problème

Pour résumer, on cherche à trouver μ , σ et π qui maximisent

$$\ln(p(Q | \mu, \sigma, \Pi)) = \sum_{j=1}^N \ln \left(\sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q_j)) \right)$$

On trouvera la solution en posant, **pour tout m entre 1 et k** :

$$\frac{d}{d\mu_m} \ln(p(Q | \mu, \sigma, \Pi)) = 0$$

$$\frac{d}{d\Pi_m} \ln(p(Q | \mu, \sigma, \Pi)) = 0$$

$$\frac{d}{d\sigma_m} \ln(p(Q | \mu, \sigma, \Pi)) = 0$$

Quelques notations

Pour la suite, on pose les notations suivantes :

$p(q \in z_i)$ La probabilité que le pixel \mathbf{q} ait eut sa valeur attribuée par le processus gaussien i
 $= \Pi_i$

$p(Q(q))$ La probabilité que le pixel \mathbf{q} ait la valeur $Q(\mathbf{q})$
 $= \sum_{j=1}^k \Pi_j * \mathcal{N}_{\mu_j, \sigma_j}(Q(q))$

$p(Q(q) | q \in z_i)$ La probabilité que le pixel \mathbf{q} ait la valeur $Q(\mathbf{q})$ sachant que sa valeur a été attribuée par le processus gaussien i .
 $= \mathcal{N}_{\mu_i, \sigma_i}(Q(q))$

Quelques notations

Question : que représente

$p(q \in z_i | Q(q))$ La probabilité que le pixel q ait eut sa valeur attribuée par le processus gaussien i sachant que sa valeur est $Q(q)$.

Question : combien cela vaut-il (théorème de Bayes) ?

$$p(q \in z_i | Q(q)) = \frac{p(Q(q) | q \in z_i) * p(q \in z_i)}{p(Q(q))}$$
$$= \frac{\Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q))}{\sum_{j=1}^k \Pi_j * \mathcal{N}_{\mu_j, \sigma_j}(Q(q))}$$


Petit rappel

Question : si on considère une fonction $u(x)$, que vaut

$$\frac{d \ln(u(x))}{dx} = \frac{u'(x)}{u(x)}$$

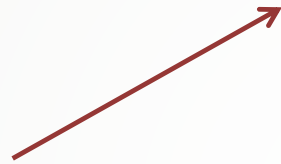
Remarque :

Forme très similaire à u'/u

$$p(q \in z_i | Q(q)) = \frac{\Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q))}{\sum_{j=1}^k \Pi_j * \mathcal{N}_{\mu_j, \sigma_j}(Q(q))}$$


Résoudre le problème

Question : Calculer la valeur de $\frac{d}{d\mu_m} \ln(p(Q(q) | \mu, \sigma, \Pi))$



Attention, ce n'est pas le vecteur μ
ici, mais sa m -ième composante

Résoudre le problème

Question : Calculer la valeur de μ_m si $\frac{d \ln(p(Q(q) | \mu, \sigma, \Pi))}{d\mu_m} = 0$

$$\frac{d \ln(p(Q(q) | \mu, \sigma, \Pi))}{d\mu_m} = \frac{1}{p(Q(q) | \mu, \sigma, \Pi)} * \frac{d p(Q(q) | \mu, \sigma, \Pi)}{d\mu_m}$$

$$p(Q(q) | \mu, \sigma, \Pi) = \sum_{i=1}^k \Pi_i * \mathcal{N}_{\mu_i, \sigma_i}(Q(q))$$

$$\frac{d p(Q(q) | \mu, \sigma, \Pi)}{d\mu_m} = \Pi_m * \frac{d \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{d\mu_m} = \Pi_m * \mathcal{N}_{\mu_m, \sigma_m}(Q(q)) * \frac{Q(q) - \mu_m}{\sigma_m^2}$$

Résoudre le problème

$$\frac{d p(Q(q) | \mu, \sigma, \Pi)}{d\mu_m} = \Pi_m * \mathcal{N}_{\mu_m, \sigma_m}(Q(q)) * \frac{Q(q) - \mu_m}{\sigma_m^2}$$

$$\frac{d \ln(p(Q(q) | \mu, \sigma, \Pi))}{d\mu_m} = \frac{1}{p(Q(q) | \mu, \sigma, \Pi)} * \frac{d p(Q(q) | \mu, \sigma, \Pi)}{d\mu_m}$$

$$= \frac{\Pi_m * \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{j=1}^k \Pi_j * \mathcal{N}_{\mu_j, \sigma_j}(Q(q))} * \frac{Q(q) - \mu_m}{\sigma_m^2}$$

$$= p(q \in z_m | Q(q)) * \frac{Q(q) - \mu_m}{\sigma_m^2}$$

Résoudre le problème

Question : Calculer la valeur de μ_m qui maximise $\ln(p(Q | \mu, \sigma, \Pi))$

On trouve

$$\mu_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q)) \times Q(q)}{\sum_{q \in Q} p(q \in z_m | Q(q))}$$

Résoudre le problème

Si on souhaite généraliser le calcul à l'image entière, il suffit de sommer les dérivées sur chaque point de l'image :

$$\frac{\partial \ln(p(Q|\mu, \sigma, \Pi))}{\partial \mu_m} = \frac{\partial \sum_{q \in Q} \ln(p(Q(q)|\mu, \sigma, \Pi))}{\partial \mu_m} = 0$$

$$\Leftrightarrow \sum_{q \in Q} p(q \in z_m | Q(q)) \times \frac{Q(q) - \mu_m}{\sigma_m^2} = 0$$

$$\Leftrightarrow \sum_{q \in Q} p(q \in z_m | Q(q)) \times \frac{Q(q) - \mu_m}{\sigma_m^2} = 0$$

Résoudre le problème

$$\Leftrightarrow \sum_{q \in Q} p(q \in z_m | Q(q)) \times \frac{Q(q) - \mu_m}{\sigma_m^2} = 0$$

$$\Leftrightarrow \mu_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q)) \times Q(q)}{\sum_{q \in Q} p(q \in z_m | Q(q))}$$

Résoudre le problème

Question : Calculer la valeur de $\frac{d}{d\sigma_m} \ln(p(Q(q) | \mu, \sigma, \Pi))$,

Puis calculer la valeur de σ_m qui maximise $\ln(p(Q | \mu, \sigma, \Pi))$

On trouve

$$\sigma_m = \sqrt{\frac{\sum_{q \in Q} p(q \in z_m | Q(q)) \times (Q(q) - \mu_m)^2}{\sum_{q \in Q} p(q \in z_m | Q(q))}}$$

Résoudre le problème

Question : Calculer la valeur de σ_m si $\frac{\partial \ln(p(Q|\mu, \sigma, \Pi))}{\partial \sigma_m} = 0$

$$\begin{aligned} \frac{\partial \ln(p(Q(q)|\mu, \sigma, \Pi))}{\partial \sigma_m} &= \frac{\partial \ln\left(\sum_{j=1}^k \Pi_j N_{\mu_j, \sigma_j}(Q(q))\right)}{\partial \sigma_m} \\ &= \frac{1}{\sum_{j=1}^k \Pi_j N_{\mu_j, \sigma_j}(Q(q))} \times \frac{\partial \sum_{j=1}^k \Pi_j N_{\mu_j, \sigma_j}(Q(q))}{\partial \sigma_m} \end{aligned}$$

Résoudre le problème

$$\begin{aligned}
 \frac{\partial \ln(p(Q(q)|\mu, \sigma, \Pi))}{\partial \sigma_m} &= \frac{1}{\sum_{j=1}^k \Pi_j N_{\mu_j, \sigma_j}(Q(q))} \times \frac{\partial \sum_{j=1}^k \Pi_j N_{\mu_j, \sigma_j}(Q(q))}{\partial \sigma_m} \\
 &= \frac{\Pi_m}{\sum_{j=1}^k \Pi_j N_{\mu_j, \sigma_j}(Q(q))} \times \frac{\partial N_{\mu_m, \sigma_m}(Q(q))}{\partial \sigma_m}
 \end{aligned}$$

Résoudre le problème

$$\begin{aligned}
 \frac{\partial N_{\mu_m, \sigma_m}(Q(q))}{\partial \sigma_m} &= -\frac{1}{\sigma_m^2 \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{Q(q) - \mu_m}{\sigma_m} \right)^2} + \frac{1}{\sigma_m \sqrt{2\pi}} \frac{(Q(q) - \mu_m)^2}{\sigma_m^3} e^{-\frac{1}{2} \left(\frac{Q(q) - \mu_m}{\sigma_m} \right)^2} \\
 &= -\frac{1}{\sigma_m} N_{\mu_m, \sigma_m}(Q(q)) + \frac{(Q(q) - \mu_m)^2}{\sigma_m^3} N_{\mu_m, \sigma_m}(Q(q)) \\
 &= N_{\mu_m, \sigma_m}(Q(q)) \times \left(\frac{(Q(q) - \mu_m)^2}{\sigma_m^3} - \frac{1}{\sigma_m} \right)
 \end{aligned}$$

Résoudre le problème

$$\frac{\partial \ln(p(Q(q)|\mu, \sigma, \Pi))}{\partial \sigma_m} = \frac{\Pi_m}{\sum_{j=1}^k \Pi_j N_{\mu_j, \sigma_j}(Q(q))} \times N_{\mu_m, \sigma_m}(Q(q)) \times \left(\frac{(Q(q) - \mu_m)^2}{\sigma_m^3} - \frac{1}{\sigma_m} \right)$$

$$= p(q \in z_m | Q(q)) \times \left(\frac{(Q(q) - \mu_m)^2}{\sigma_m^3} - \frac{1}{\sigma_m} \right)$$

Résoudre le problème

Donc, en généralisant à tout Q :

$$\frac{\partial \ln(p(Q|\mu, \sigma, \Pi))}{\partial \sigma_m} = \sum_{q \in Q} p(q \in z_m | Q(q)) \times \left(\frac{(Q(q) - \mu_m)^2}{\sigma_m^3} - \frac{1}{\sigma_m} \right) = 0$$

$$= \sum_{q \in Q} p(q \in z_m | Q(q)) \times \left(\frac{(Q(q) - \mu_m)^2}{\sigma_m^3} - \frac{1}{\sigma_m} \right) = 0$$

$$\Leftrightarrow \sigma_m = \sqrt{\frac{\sum_{q \in Q} p(q \in z_m | Q(q)) \times (Q(q) - \mu_m)^2}{\sum_{q \in Q} p(q \in z_m | Q(q))}}$$

Résoudre le problème

Question : Calculer la valeur de Π_m si $\frac{d}{d\Pi_m} \ln(p(Q | \mu, \sigma, \Pi)) = 0$

Ici, on a une contrainte concernant les Π_k :
$$\sum_{i=1}^k \Pi_i = 1$$

Il faut utiliser la technique des **multiplicateurs de Lagrange** pour intégrer la contrainte au calcul d'optimum (on ne veut pas trouver le véritable optimum de la fonction, mais l'optimum sous contrainte) :

$$F(\Pi_m, \lambda) = \ln(p(Q | \mu, \sigma, \Pi)) + \lambda \left(\sum_{i=1}^k (\Pi_i) - 1 \right)$$

Ça fait 0



Résoudre le problème

On veut donc trouver Π_m tel que $\frac{d}{d\Pi_m} F(\Pi_m, \lambda) = 0$

On trouve

$$\Pi_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q))}{N}$$

Résoudre le problème

On veut donc trouver Π_m tel que $\frac{d}{d\Pi_m} F(\Pi_m, \lambda) = 0$

$$\frac{dF(\Pi_m, \lambda)}{d\Pi_m} = \sum_{q \in Q} \left(\frac{\mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \lambda = 0$$

$$\Leftrightarrow \Pi_m \sum_{q \in Q} \left(\frac{\mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \lambda \Pi_m = 0$$

$$\Leftrightarrow \sum_{q \in Q} \left(\frac{\Pi_m \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \lambda \Pi_m = 0$$

Résoudre le problème

On a donc

$$\sum_{q \in Q} \left(\frac{\Pi_m \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \lambda \Pi_m = 0$$

Cette équation est vraie pour tout m entre 1 et k , on peut donc sommer l'équation pour m allant de 1 à k et continuer à avoir 0 :

$$\sum_{m=1}^k \sum_{q \in Q} \left(\frac{\Pi_m \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \sum_{m=1}^k \lambda \Pi_m = 0$$

$$\Leftrightarrow \sum_{q \in Q} \left(\frac{\sum_{m=1}^k \Pi_m \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \lambda \sum_{m=1}^k \Pi_m = 0$$

Résoudre le problème

$$\sum_{q \in Q} \left(\frac{\sum_{m=1}^k \Pi_m \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \lambda \sum_{m=1}^k \Pi_m = 0$$

$$\Leftrightarrow \sum_{q \in Q} 1 + \lambda = 0$$

$$\Leftrightarrow N + \lambda = 0$$

$$\Leftrightarrow \lambda = -N$$

Résoudre le problème

On reprend l'équation

$$\frac{dF(\Pi_m, \lambda)}{d\Pi_m} = \sum_{q \in Q} \left(\frac{\mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) + \lambda = 0$$

$$\Leftrightarrow \sum_{q \in Q} \left(\frac{\mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) = N$$

$$\Leftrightarrow \sum_{q \in Q} \left(\frac{\Pi_m \mathcal{N}_{\mu_m, \sigma_m}(Q(q))}{\sum_{i=1}^k \Pi_i \mathcal{N}_{\mu_i, \sigma_i}(Q(q))} \right) = \Pi_m N$$

$$\Leftrightarrow \sum_{q \in Q} p(q \in z_m | Q(q)) = \Pi_m N$$

Résoudre le problème

$$\Leftrightarrow \sum_{q \in Q} p(q \in z_m | Q(q)) = \Pi_m N$$

$$\Leftrightarrow \Pi_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q))}{N}$$

Problème résolu ?

On obtient donc les solutions suivantes à notre problème :

$$\mu_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q)) \times Q(q)}{\sum_{q \in Q} p(q \in z_m | Q(q))}$$

$$\sigma_m = \sqrt{\frac{\sum_{q \in Q} p(q \in z_m | Q(q)) \times (Q(q) - \mu_m)^2}{\sum_{q \in Q} p(q \in z_m | Q(q))}}$$

$$\Pi_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q))}{N}$$

Le problème est-il résolu ?

Malheureusement non, car $p(q \in z_m | Q(q))$ dépend, entre autre, de μ_m, σ_m, Π_m , mais aussi de tous les u_i, σ_i, Π_i .

On ne peut pas extraire, des ces équations, une solution exacte immédiatement.

On peut par-contre construire un algorithme...

L'algorithme Expectation-Minimization (EM)

L'algorithme EM (Espérance-Maximisation en français) permet de trouver une valeur approchée des paramètres de notre problème en répétant deux étapes :

1. On **estime l'espérance de la vraisemblance**, c'est-à-dire la probabilité que la valeur d'un pixel ait été attribuée par un des processus que l'on cherche à caractériser :

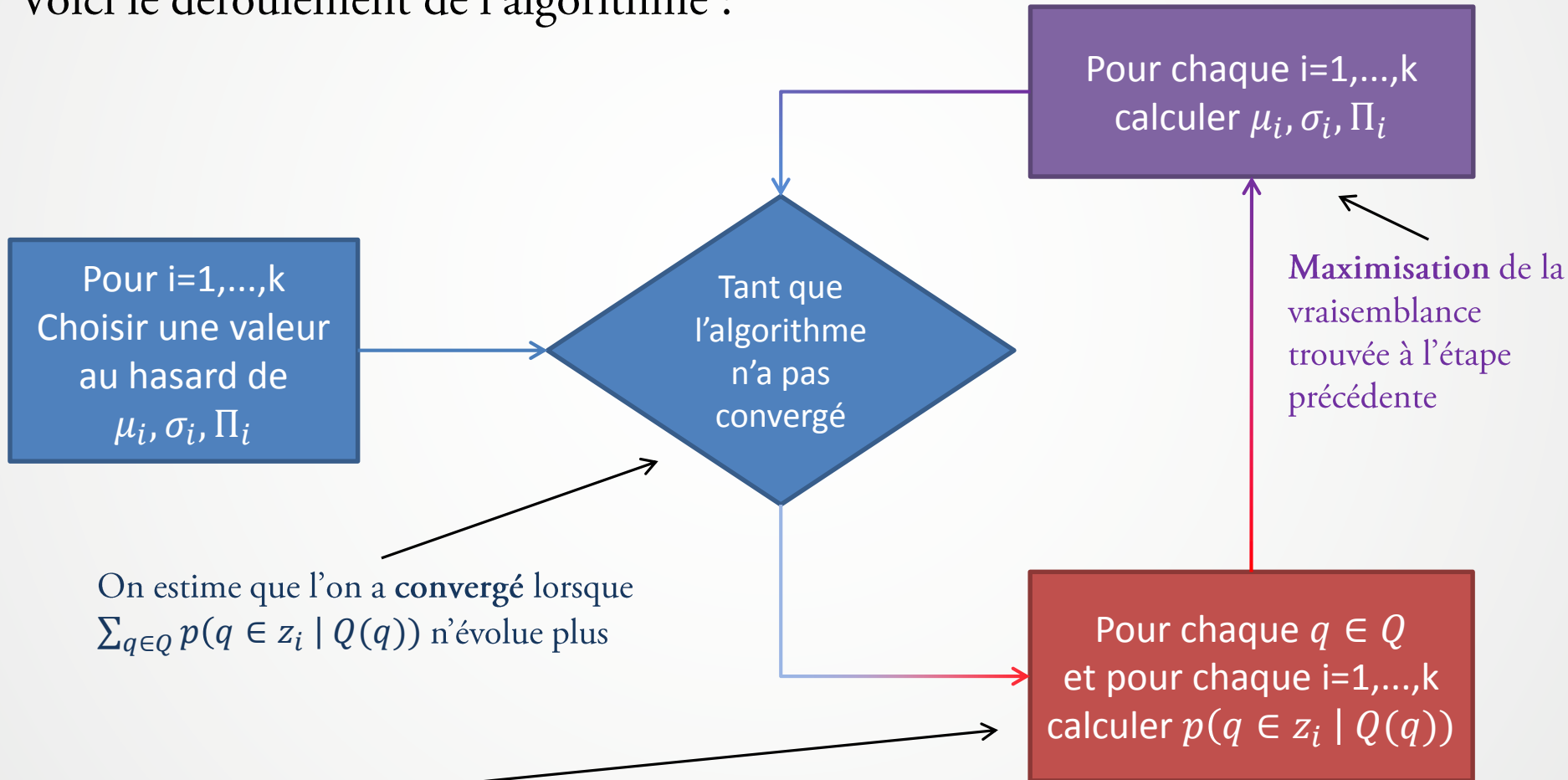
$$p(q \in z_i | Q(q))$$

2. A partir des ces calculs, on **estime les valeurs des paramètres** des processus que l'on cherche à caractériser :

$$\mu_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q)) \times Q(q)}{\sum_{q \in Q} p(q \in z_m | Q(q))} \quad \sigma_m = \sqrt{\frac{\sum_{p \in Q} p(q \in z_m | Q(q)) \times (Q(q) - \mu_m)^2}{\sum_{p \in Q} p(q \in z_m | Q(q))}} \quad \Pi_m = \frac{\sum_{q \in Q} p(q \in z_m | Q(q))}{N}$$

L'algorithme Expectation-Minimization (EM)

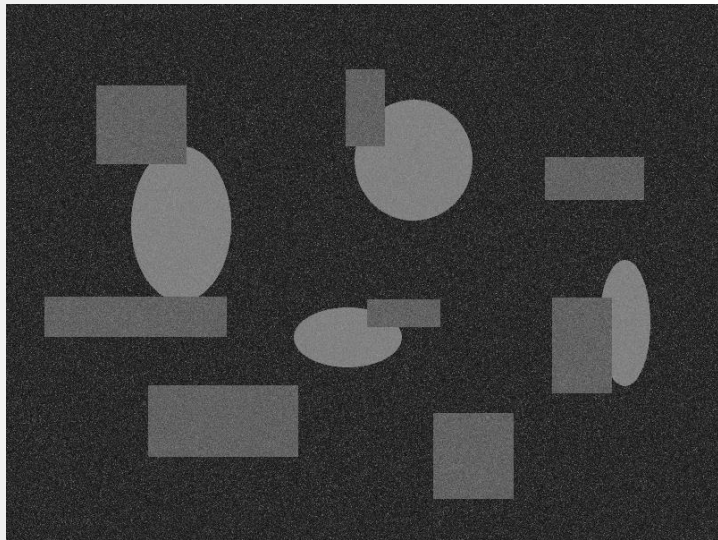
Voici le déroulement de l'algorithme :



On estime que l'on a **convergé** lorsque $\sum_{q \in Q} p(q \in z_i | Q(q))$ n'évolue plus

Calcul de l'**espérance** de la vraisemblance en fonction des observations (Q) et des éléments calculés à la phase précédent

Résultat



L'algorithme trouve

$$\mu_1 = 119,47 \quad \sigma_1 = 19,96 \quad \Pi_1 = 0,80$$

$$\mu_2 = 169,49 \quad \sigma_2 = 06,99 \quad \Pi_2 = 0,12$$

$$\mu_3 = 189,48 \quad \sigma_3 = 04,02 \quad \Pi_3 = 0,08$$

La vérité terrain était

$$\mu_1 = 120 \quad \sigma_1 = 20 \quad \Pi_1 = 0,80$$

$$\mu_2 = 170 \quad \sigma_2 = 07 \quad \Pi_2 = 0,12$$

$$\mu_3 = 190 \quad \sigma_3 = 04 \quad \Pi_3 = 0,08$$

Résultat

On peut attribuer à chaque pixel une valeur selon la gaussienne à laquelle il est le plus probable qu'il appartienne.

Pour chaque pixel q , on cherche

$$\operatorname{argmax}_i p(q \in z_i | Q(q))$$

et on attribue à chaque pixel q la valeur μ_i

Le résultat n'est pas parfait, mais avec un filtre médian, on peut l'améliorer...

Méthodes par histogramme

Conclusion

Conclusion sur les méthodes par histogrammes

Les méthodes par histogrammes ont l'avantage d'être **simples à mettre en œuvre**.

Cependant, elles ne prennent pas en compte une information d'importance : **la disposition spatiale des pixels**.

Leurs **résultats sont donc très limités** car, au final, on finit toujours par effectuer un seuil de l'image en fonction d'un paramètre calculé dynamiquement...