

## **TP2 : Histogramme**

### **Segmentation d'une image**

#### **Etalement**

#### **Egalisation**

#### ***Preliminaires***

N'oubliez pas de créer le répertoire /TP2 à côté du répertoire /TP1 du premier TP. Une nouvelle image a été rajoutée (liftingbody.jpg). La mettre dans le répertoire "Image\_jpg". Lancer le programme Matlab® et positionner le répertoire courant sur l'emplacement du répertoire /TP2.

#### ***Objectifs du TP***

Ce TP concerne la mise en place de quelques traitements d'image liés à l'histogramme d'une image en niveaux de gris ou en couleur.

Tout d'abord, on s'intéressera à l'utilisation de la manipulation d'histogramme pour réaliser une segmentation d'image.

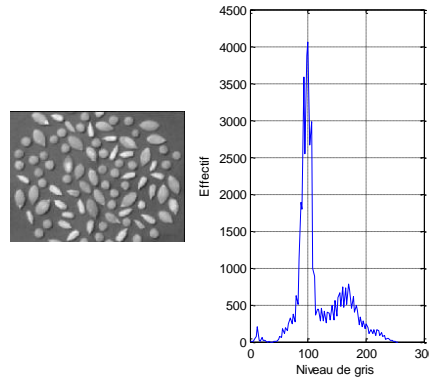
Ensuite un étalement d'histogramme sera réalisé sur une image à niveaux de gris, suivie par une méthode d'égalisation d'histogramme.

Après, nous nous intéresserons à la mise en place de ces deux méthodes sur des images en couleurs. Une première implémentation sera réalisée de façon naïve en traitant séparément les trois histogrammes rouge, vert et bleu.

Enfin, après avoir constaté l'échec de cette approche, l'étalement et l'égalisation seront réalisés après un changement de base approprié en prenant en compte la luminance de l'image tout en préservant la teinte. Une fois modifiée, l'image sera affichée dans son espace RGB original.

#### ***Segmentation d'une image par seuillage des niveaux de gris***

La segmentation d'une image par le seuillage des niveaux de gris est la technique de segmentation la plus simple. Elle s'applique à des images dont l'histogramme en niveaux de gris comporte plusieurs modes bien distinctes, comme par exemple pour l'image ci-dessous. L'objectif est alors de déterminer le placement des seuils afin de faire la segmentation en un nombre donné de classes de niveau de gris. Dans l'exemple ci-dessous, on pourrait choisir deux ou trois classes. Cela demanderait de trouver respectivement un, ou deux seuils.



- 1°) Editer le programme “TP2\_Segment.m”. Le programme ne peut pas s’exécuter tel quel car il vous faut préalablement compléter la fonction “segmentation.m”. Editer cette fonction, lire le code et expliquer le fonctionnement de l’algorithme de segmentation qui est codé entre les lignes 20 et 75.
- 2°) Compléter le code pour former l’image segmentée. Suivant le choix de l’utilisateur (variable “mode”), la valeur du niveau de gris pour une région donnée pourra être soit la valeur milieu de l’intervalle bornée par les seuils, soit la valeur moyenne des niveaux de gris qui sont entre deux seuils.
- 3°) Vérifier votre code sur des images “simples” (par exemple “graines.jpg” ou “liftingbody.jpg”). Faire varier le nombre de classes. Montrer et discuter les résultats obtenus.
- 4°) Compléter le code du programme “TP2\_Segment.m” pour que les pixels de la classe la plus fréquente soient affichés en rouge. Utiliser pour ce faire, la manipulation par table de couleurs (instruction “colormap”). Expliquer l’implémentation réalisée.
- 5°) Faire d’autres tests de segmentation sur des images plus complexes. Conclure sur les avantages et les inconvénients de la méthode de segmentation.

## **Modification d’histogramme sur une image en noir et blanc**

### **A°) Étalement d’un histogramme de niveaux de gris**

Dans l’ensemble du TP, on pourra simplement se servir de la fonction matlab “imhist” qui calculera directement l’histogramme d’une image (Attention, une image encodée en uint8 devra avoir des valeurs comprises entre 0 et 255 alors qu’une image encodée sur des “double” doit avoir des valeurs comprises entre 0 et 1. Le TP est réalisé de façon à garder une structure d’image encodée sur des entiers).

Par rapport, à l’équation d’étalement d’histogramme vue en cours, on introduit ici deux nouveaux paramètres  $\alpha$  et  $\beta$  permettant de contrôler les niveaux de gris minimum et maximum dans l’équation d’étalement :

$$n' = 255 \cdot \frac{n - n_{min}}{n_{max} - n_{min}},$$

dans le cadre de niveaux codés entre 0 et 255, avec  $n_{max} = \frac{I_{max}}{\alpha}$  et  $n_{min} = I_{min} \cdot \beta$ . Ces paramètres  $\alpha$  et  $\beta$  pondèrent les valeurs maximale et minimale pour éviter les artéfacts de

valeurs isolées. On peut ainsi faire décroître artificiellement le maximum (respectivement de faire accroître le minimum).

A.1°) Expliquer brièvement ce que renvoie l'appel "`[h,x] = imhist(pic, 256)`"

A.2°) Suivre le script "`TP_Histo.m`". L'étape 1 consiste à comprendre et compléter la fonction "`histo_etallement`" qui va accroître la dynamique de l'image par étallement.

Visualiser le résultat obtenu. Quelles remarques pouvez-vous faire sur la forme de l'histogramme ainsi déformé ?

A.3°) Faire varier les paramètres "`factor max`" et "`factor min`". Expliquez leur rôle, les avantages ainsi que les inconvénients visualisés lorsque "`factor max`" est  $>1$  par exemple. Comment évolue l'histogramme avec ce paramètre ? Que peut-on dire sur la dernière valeur de l'histogramme suivant la valeur de ce paramètre ?

A.4°) Pour l'image "`venise`", peut-on visualiser à la fois le ciel plus clair dans toutes ses nuances, tout en éclaircissant correctement le reste du paysage ? Quelle limitation inhérente à cette technique empêche d'avoir un résultat correct dans toutes les parties de l'image ?

### **B°) Egalisation d'histogramme de niveaux de gris**

La seconde étape du script va consister à réaliser l'égalisation de l'histogramme.

B.1°) Comprendre comment est programmée la fonction "`histo egalisation`".

B.2°) Observer le résultat obtenu, que peut-on dire concernant le contraste de l'image ? Observez la forme de l'histogramme.

B.3°) Y a-t-il toujours conflit entre la visualisation détaillée du ciel et des parties sous-exposées sur la même image ("`venise`")? Expliquez pourquoi.

### ***Modification d'histogramme sur une image en couleurs, approche naïve***

La partie suivante va désormais traiter du cas des images en couleur. Une approche naïve du traitement de celle-ci consisterait à traiter séparément chaque composante (R,G,B) avec les fonctions traitant du cas scalaire, en niveaux de gris. C'est celle-ci que l'on va tout d'abord expérimenter afin de visualiser ces limites.

### **C°) Etalement d'histogramme de couleurs**

On va cette fois appliquer trois fois l'étalement d'histogramme sur chaque composante couleur de l'image

C.1°) Compléter le programme "`TP_Histo`" en partie 3 de sorte à appliquer l'étalement sur chacune des composantes

C.2°) Observez le résultat obtenu pour différentes valeurs des paramètres "`factor`". Quelles remarques pouvez-vous faire sur la teinte ? Expliquer ce phénomène, commenter les déformations de l'histogramme.

### **D°) Egalisation d'histogramme de couleurs**

De même que pour l'étalement, on applique désormais l'égalisation sur les trois composantes de façon séparée.

D.1°) Compléter la partie 4 du "TP Histo". Appliquez trois fois l'égalisation d'histogramme sur chaque composante.

D.2°) Quels constats pouvez-vous faire sur les teintes ?

### **Changement de base "couleur"**

Les problèmes précédents de teintes proviennent de la mise en place naïve de méthodes scalaires sur dans un espace vectoriel de dimension trois. Pour éviter de faire varier les teintes lors de l'augmentation du contraste de l'image, nous allons effectuer un changement de base pour nous placer dans l'espace YCbCr.

La matrice associée à la transformation est donnée par :

$$M = \begin{pmatrix} 0.299 & 0.587 & 0.144 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix}$$

telle que :  $\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = M \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$ .

E.1°) Ce nouvel espace est appelé Luminance-Chrominance, expliquez pourquoi. Quelles couleurs sont en opposition ? Quelle couleur contribue la plus/la moins à la luminance ? Pourquoi ce choix ? Entre quelles valeurs varie la luminance ?, les chrominances ? Conclure sur ce type d'encodage en YCbCr sous forme d'entier ou de double.

E.2°) Donner le changement de base inverse entre YCrCb et RGB.

E.3°) Compléter les fonctions de changement de base "convert\_RGB\_to\_YCbCr" et "convert\_YCbCr\_to\_RGB".

### **Etalement d'histogramme de couleurs (YCbCr)**

L'opération d'étirement va maintenant se réaliser dans l'espace Luminance-Chrominance. On va réaliser cette opération uniquement sur la luminance afin d'éclaircir l'image sans modifier les teintes liées aux chrominances.

F.1°) Complétez le script "TP Histo" en partie 5 afin d'appliquer la conversion en YCrCb puis l'étalement sur la coordonnée Y. Enfin, rétablissez l'image en RGB afin de pouvoir la visualiser.

F.2°) Faire varier les paramètres "factor min" et "factor max". Quelles améliorations a apporté le travail dans l'espace de couleur YCbCr ?

### **Egalisation d'histogramme de couleurs (YCbCr)**

On réalise l'opération d'égalisation d'histogramme dans l'espace (Y, Cb, Cr) également pour la composante de luminance.

G.1°) Complétez la partie 6 du script "TP\_Histo" afin d'appliquer l'égalisation d'histogramme sur la composante Y, puis afficher l'image corrigé dans l'espace RGB.

G.2°) Comparez les teintes de l'image obtenue avec le cas naïf appliqué sur R, G et B

### **Comparaison**

La dernière partie vous permet de réaliser rapidement des comparaisons pour différentes images entre la méthode appliquée en RGB, et celle appliquée dans l'espace Luminance-Chrominance.