

## Traitement d'image

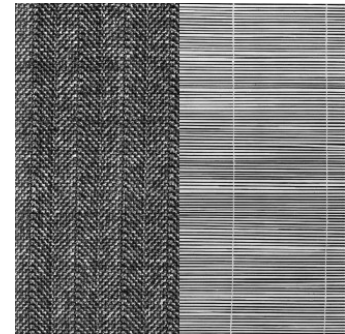
### TP N° 2 : Segmentation par attributs de texture

#### I. Présentation du problème

L'objet du TP est une étude d'un attribut de texture pour la segmentation de l'image constituée de deux régions de la figure 1( *texture3.tif*).

Pour réaliser la segmentation, nous allons utiliser l'attribut de Law L5S5 défini par le masque suivant (c.f. cours de traitement d'image) :

**L5S5** = [ -1      0      2      0      -1 ;  
          -4      0      8      0      -4 ;  
          -6      0      12      0      -6 ;  
          -4      0      8      0      -4 ;  
          -1      0      2      0      -1 ] ;



**Figure 1**

Rappel : Le calcul des attributs de Law s'effectue en deux étapes :

**Etape 1.** Dans un premier temps, l'image est convoluée par le masque chargé de ne conserver que les composantes fréquentielles représentatives (ici le masque L5S5 précédemment défini).

**Etape 2.** L'attribut est ensuite déterminé en calculant la variance locale sur l'image en utilisant un masque de grande taille (11x11 ou 15x15).

La variance locale est donnée par :  $\sigma^2(i, j) = \frac{1}{(2N+1)^2} \sum_{k=-N}^N \sum_{l=-N}^N (I(i+k, j+l) - \bar{I})^2$  où  $\bar{I}$  est la moyenne

locale définie par :  $\bar{I}(i, j) = \frac{1}{(2N+1)^2} \sum_{k=-N}^N \sum_{l=-N}^N (I(i+k, j+l))$

Ce traitement sera réalisé par la fonction **variance\_locale** disponible sur le moodle.

#### II. Manipulation

Importer tout d'abord les modules habituels :

```
>>import numpy as np
>>import cv2 as cv
>>from matplotlib import pyplot as plt
```

Charger ensuite l'image *texture3.tif* (disponible sur la plateforme moodle) :

```
>>im = cv.imread('texture3.tif', cv.IMREAD_GRAYSCALE)
```

##### 1. Étude du filtre L5S5.

Dans un premier temps nous allons étudier les effets du filtrage seul. Réaliser la convolution de l'image *texture3.tif* par le filtre L5S5. Pour construire le filtre L5S5 vous pouvez utiliser l'instruction suivante :

```
>> msk_lap = np.array([[-1,0,2,0,-1],[-4,0,8,0,-4],[-6,0,12,0,-6],
[-4,0,8,0,-4],[-1,0,2,0,-1]])
```

On rappelle que l'opération de convolution est réalisée à l'aide de la fonction **cv.filter2D** (c.f. tp1 et tp2).

Afficher l'image. Qu'observe-t-on ? Conclusion.

## 2. Étude de l'attribut

Le but de l'attribut est d'effectuer un calcul sur l'image qui sera différent pour les deux textures ce qui permettra de les séparer par segmentation.

Pour étudier cet effet, nous allons maintenant calculer les images attributs à partir des deux images *texture1.tif* et *texture2.tif* qui représentent les deux textures séparément. On rappelle que le calcul de l'attribut est constitué de la succession du filtrage par **L5S5** et du calcul de la variance locale par la fonction **variance\_locale(imf,mask)** (fournie). La valeur de **msk** correspond à la taille en pixel du masque (pour un masque de 11x11 il faut donc entrer la valeur 11).

a/ Calculer les valeurs moyennes des deux images attributs. Conclusion.

b/ Afficher les histogrammes des deux images attributs. On utilise pour cela la fonction **hist()** du module pyplot.

Exemple pour afficher l'histogramme de **im** :

```
>> plt.hist(im.ravel(), 100)
>> plt.show()
```

Affiche l'histogramme de l'image **img** sur 100 échantillons. (N.B. : La méthode **ravel()** permet de réorganiser les valeurs de **img** en un vecteur.)

Conclusion ?

## 3. Segmentation de l'image

Nous allons dans cette partie réaliser la segmentation de l'image *texture3.tif*. Pour cela il est nécessaire de calculer au préalable l'image attributs à l'aide des deux étapes convolution et variance locale comme précédemment.

a/ En utilisant l'implantation fournie dans OpenCV de l'algorithme *Otsu*, calculer l'image segmentée de *texture3.tif*. Cette fonction s'applique simplement sur une image **img** de la façon suivante :

```
imb = cv.threshold(img, 0, 255, cv.THRESH_BINARY | cv.THRESH_OTSU)
```

Le résultat est l'image **imb** ne contenant que des 1 et des 2 correspondant aux deux classes détectées.

b/ Segmentation supervisée. Pour réaliser la segmentation supervisée, nous allons utiliser les deux images *texture1.tif* et *texture2.tif* qui représentent les deux textures séparées. Calculer les deux images attributs de *texture1.tif* et *texture2.tif* (question 2). Les deux histogrammes des attributs des deux textures seront modélisés par des fonctions Gaussiennes dont l'expression analytique est donnée par :

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad \text{où } m \text{ et } \sigma \text{ sont respectivement la moyenne et l'écart type de la Gaussienne.}$$

Le seuil optimal est donné par l'intersection des deux lois Gaussiennes (c.f. figure) correspondant aux histogrammes des deux images attributs obtenues sur les deux textures.

Calculer analytiquement ce seuil. Appliquer le seuillage sur l'image des attributs de *texture3.tif*

**N.B. :** Pour calculer la valeur moyenne et l'écart type de l'attribut vous pouvez utiliser respectivement les fonctions :

```
m = att.mean()
ec = np.sqrt(att.var())
```

Pour calculer l'image binaire par seuillage, vous pouvez utiliser la fonction suivante :

```
imb = (img > seuil)
```

Le résultat sera une image contenant des 1 pour les pixels dont la valeur est inférieure à **seuil** et 2 pour les pixels dont la valeur est supérieure.

