Segmentation d'une images.

1• Objectif de ce travail.

Il s'agit de réaliser une fonction de séparation/fusion. Comme les structures de matlab ne sont pas appropriées pour ces opérations, c'est une méthode de séparation-fusion simplifiée qui vous est proposée ici.
La séparation/fusion consiste à séparer l'images en régions homogènes par une segmentation systématique, puis à regrouper ces régions par une méthode proche de la croissance de région. Les calculs impliqués étant particulièrement longs, nous vous conseillons de prendre bien soin de ne pas faire de calculs inutiles.

2• Séparation.

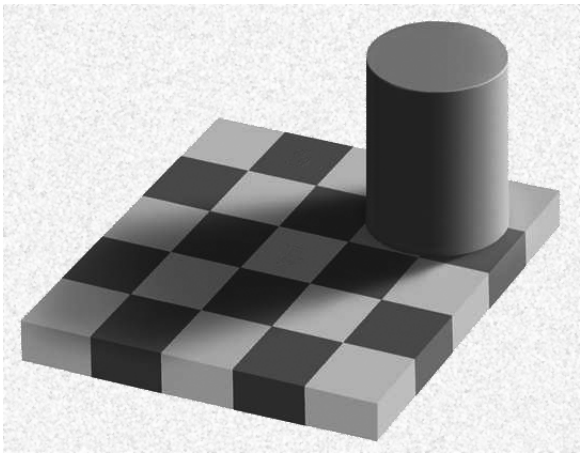

Figure 1 : Image de test.

Vous disposez d'une image de test portant le nom de "IllusionNG.tif". La méthode de séparation consiste à estimer la variance de cette image et à la séparer en 4 parties à peu près égales si la variance dépasse un seuil que vous définirez a priori (utilisez une variable de façon à pouvoir le changer facilement).

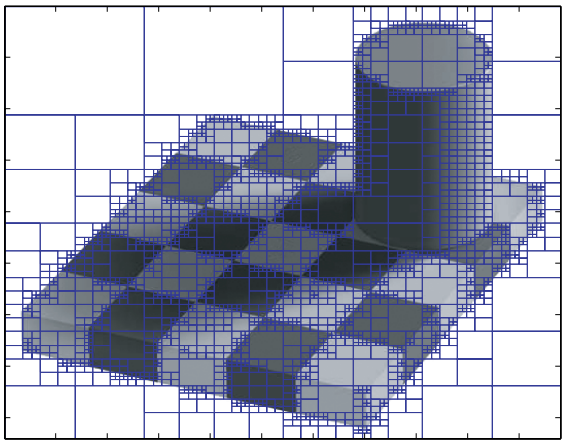

Figure 2 : image de la Figure 1 segmentée.

Cette procédure doit être itérée récursivement pour chaque sous-région, jusqu'à ce qu'il n'y ait plus que des régions dont la variance soit en dessous du seuil fixé. Evidemment,

il faut fixer aussi un seuil de taille minimale pour les régions (par exemple 2x2). Faites attention à ce que la taille des régions n'est pas forcément divisible par deux. Il faut donc vous arranger pour que votre algorithme fonctionne même avec des tailles impaires. *Application numérique : vous pouvez commencer avec un seuil de 100 pour la variance.* Vous devez obtenir une image résultante sous une forme explicite représentée sur la Figure 2.

De façon à vous permettre de démarrer rapidement, on vous fournit ici le début du code :

```
MonImage = double(imread('Illusion.tif')) ;
[Nlin, Ncol] = size(MonImage) ;
figure(1) ; image(uint8(MonImage)) ; colormap(gray(256)) ;
axis image ; drawnow ;
NombreMaxDeRegions = floor( (Nlin/2) * (Ncol/2) ) ;
Region = zeros(NombreMaxDeRegions,4) ;
Region(1,:) = [1, Nlin , 1, Ncol] ; % initialisation de la premiere region
... à complèter
```

Chaque élément du tableau Region contient les limites du pavé délimitant une région. On initialise la première région avec l'image complète. Si la variance de cette région est supérieure au seuil choisi, elle est séparée en 4 (on rajoute donc 3 régions) et l'une des régions créées remplace la région segmentée.
Pour vous aider dans l'implantation de cet algorithme, nous vous fournissons le code d'une fonction `MomentsImage` qui calcule les moyennes et variances d'une région de l'image. Son appel est :

```
[moyenne, variance] = MomentsImage(MonImage,Pave) ;
```

ou encore

```
[moyenne, variance, nombre_de_points] = MomentsImage(MonImage,Pave) ;
```

où Pave est un vecteurs à 4 composantes `[lin_min, lin_max, col_min, col_max]`, `moyenne` la moyenne de la région, `variance` la variance de la région et `nombre_de_points` le nombre de points de la région délimitée par le pavé.
Vous sauverez ce code dans un fichier nommé `MomentsImage.m`.

```
function [moyenne, variance, nombre_de_points] = MomentsImage(MonImage, Pave)

[Nlin, Ncol] = size(MonImage) ;

Pave = max(Pave,1) ;
Pave(2) = min(Pave(2),Nlin) ;
Pave(4) = min(Pave(4),Ncol) ;
Vecteur = MonImage(Pave(1):Pave(2),Pave(3):Pave(4)) ;
Vecteur = Vecteur(:) ;

moyenne = mean(Vecteur) ;
variance = var(Vecteur) ;

if nargout == 3
 nombre_de_points = length(Vecteur) ;
end
```

3• AGRÉGATION DE RÉGIONS APRÈS SÉPARATION.

L'agrégation complète est un peu difficile. C'est pourquoi nous vous proposons d'agréger seulement les régions connexes à une région sélectionnée.
Vous aurez besoin de la fonction suivante (que vous devez recopier dans un fichier a part nomme Voisin.m). Elle renvoie 1 si deux pavés sont voisins, et 0 sinon.

```
function [voisin] = Voisin(Rectangle1, Rectangle2)

Point1(1,1:2) = [Rectangle1(1), Rectangle1(3)] ;
Point1(2,1:2) = [Rectangle1(1), Rectangle1(4)] ;
Point1(3,1:2) = [Rectangle1(2), Rectangle1(3)] ;
Point1(4,1:2) = [Rectangle1(2), Rectangle1(4)] ;

Point2(1,1:2) = [Rectangle2(1), Rectangle2(3)] ;
Point2(2,1:2) = [Rectangle2(1), Rectangle2(4)] ;
Point2(3,1:2) = [Rectangle2(2), Rectangle2(3)] ;
Point2(4,1:2) = [Rectangle2(2), Rectangle2(4)] ;

voisin=0 ;
for i=1:4
    for j=1:4
        if ( abs( Point1(i,1) - Point2(j,1) ) < 2 ) && ( abs( Point1(i,2) -
Point2(j,2) ) < 2 )
            voisin = 1 ;
        end
    end
end
return ;
```

La méthode que vous devez implanter consiste à créer un regroupement de régions autour d'une région selectionnée. Elle peut être décomposée en quatre étapes :

• Sélection d'un zone

Le morceau de code suivant vous permet de trouver la zone issue de votre décomposition en `NombreDeRegions` régions la plus proche du point que vous avez cliqué.

```
figure(1) ;
[col,lin] = ginput(1) ;

numero_de_la_region_selectionnee = 1 ;
for numero = 1:NombreDeRegions
    lin_min = Region(numero,1) ;
    lin_max = Region(numero,2) ;
    col_min = Region(numero,3) ;
    col_max = Region(numero,4) ;
    if ( lin <= lin_max ) && ( lin >= lin_min ) && ( col <= col_max ) && (
col >= col_min )
        numero_de_la_region_selectionnee = numero ,
    end
end
```

• Création de deux tableaux :

```
GroupeDeRegion = zeros(NombreDeRegions,1) ;
RegionAttribuee = zeros(NombreDeRegions,1) ;
```

• Augmentation du groupe de région de toute région connexe à une des régions déjà agrégées minimisant un critère quadratique de distance. Le critère que vous utiliserez est la distance de Mahalanobis :
soit m_R, v_R, N_R respectivement la moyenne, la variance et le nombre de points de la région à tester.
soit m_G, v_G, N_G respectivement la moyenne, la variance et le nombre de points du groupe de région.

La distance de Mahalanobis entre la région et le groupe s'écrit : $\frac{(m_R - m_G)^2}{v_R + v_G}$

Attention dans certaines zones la variance peut être nulle, ce qui entraine une divergence du calcul de la distance. Si $(v_R + v_G) < 1.10^{-16}$ remplacez cette valeur par 1.10^{-16}. Les valeurs de seuil classiques pour la distance de Mahalanobis vont de 0.1 à 10.0. Testez différentes valeurs.

• Mise à jour des moyennes et des variances lorsqu'une région est intégrée au groupe de régions connexes :
Le nouveau nombre de points s'obtient par : $N = N_R + N_G$

La nouvelle moyenne s'obtient par : $m = \frac{N_R m_R + N_G m_G}{N}$

La nouvelle variance s'obtient par :

$$v = \frac{N_R(m_R)^2 + N_G(m_G)^2 - N(m)^2 + (N_R - 1)v_R + (N_G - 1)v_G}{N - 1}$$

Pour vous entraîner à faire des calculs sur les images, vous pouvez vérifier cette formule.

Une façon très simple de visualiser vos résultats est de créer une image couleur qui est une copie de votre image à niveaux de gris et de changer un des plans couleur de la zone de cette image correspondant à la nouvelle zone agrégée. Par exemple :

```
ImageCouleur = zeros(Nlin,Ncol,3) ;
ImageCouleur(:,:,3) = MonImage(:,:) ;
ImageCouleur(:,:,2) = MonImage(:,:) ;
ImageCouleur(:,:,1) = MonImage(:,:) ;
```

et pour afficher :

```
% modification du plan bleu
ImageCouleur(lin_min:lin_max,col_min:col_max,3) = 255 ;
% modification du plan rouge
ImageCouleur(lin_min:lin_max,col_min:col_max,1) = 0 ;
```

Faites des essais avec différentes images et différents seuils, et essayez de tirer des conclusions de vos différents essais.