

TD2 - Traitement d'image - Segmentation par classification

Le but de cette séance est d'étudier la segmentation par classification sur des images couleur et de texture. On utilisera Matlab et les fonctions et images fournies dans l'archive TD2.zip disponible sur le portail.

La séance est découpée en deux séquences :

- Travail dirigé pendant 2h45.
- Evaluation sous forme d'un QCM de 10 questions pendant les 10 dernières minutes (sans documents ni ordinateurs). Les questions pourront porter sur des notions de cours en rapport avec la séance et sur le travail réalisé pendant cette séance.

Contents

- Question 2.1 : nuages de points représentant les pixels
- Question 2.2 : Segmentation par nuées dynamiques
- Question 2.3 : Images de textures
- Question 2.4 Segmentation de l'image de texture
- Question 2.5 : Couleur et texture

Question 2.1 : nuages de points représentant les pixels

Exécuter cette section et interpréter les figures qui sont tracées :

- Que représente le nuage de points 3D ?

C'est la projection des pixels de l'image dans l'espace Rouge-Vert-Bleu

- Que représentent les nuages 2D ?

Ce sont les projections des pixels dans des espaces 2D Rouge-Vert et Rouge-Bleu. Chaque pixel est représenté avec une couleur qui est sa couleur dans l'image d'origine.

- Interpréter le nombre et la forme des nuages de points.

On observe 3 nuages de points correspondant aux 3 classes de pixels dominantes : les pixels des objets verts, les pixels des objets rouges et les pixels du fond blanc. Il y a aussi quelques points entre ces nuages qui correspondent aux points affectés par le bruit ou le flou présent autour des objets

```
clear all;  
close all;
```

```
f=imread('Images/BillesColoreesBruit.bmp');  
figure(1),imshow(f);
```

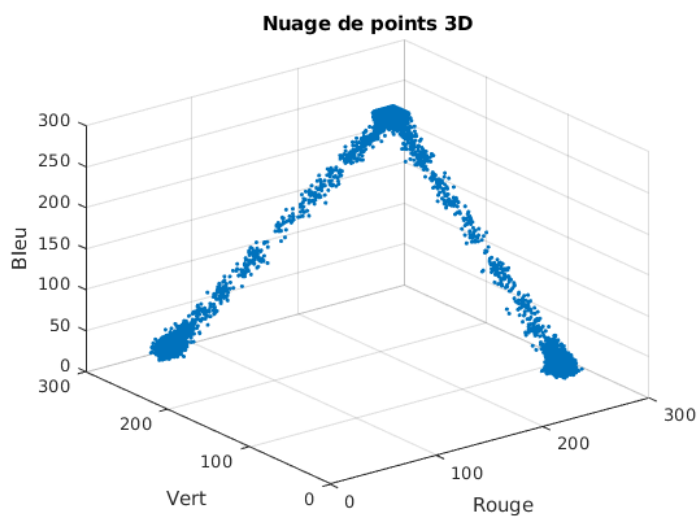
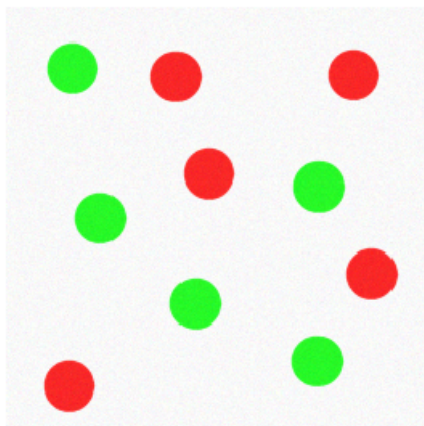
```
% Nuage de points 3D  
fr = f(:, :, 1); fv = f(:, :, 2); fb = f(:, :, 3);  
figure(2), plot3(fr(:),fv(:),fb(:),'.'); grid('on')  
title('Nuage de points 3D');  
xlabel('Rouge'); ylabel('Vert'); zlabel('Bleu');
```

```
% Nuages projetés en 2D
```

```

rv=zeros(256,256,3,'uint8');
rb=zeros(256,256,3,'uint8');
for i=1:size(f,1)
    for j=1:size(f,2)
        rv(256-f(i,j,2),f(i,j,1)+1,:)=f(i,j,:);
        rb(256-f(i,j,3),f(i,j,1)+1,:)=f(i,j,:);
    end
end
figure(3), imshow(rv);title('plan RV');
xlabel('Rouge'); ylabel('Vert');
figure(4), imshow(rb);title('plan RB');
xlabel('Rouge'); ylabel('Bleu');

```





Question 2.2 : Segmentation par nuées dynamiques

On veut segmenter l'image en utilisant la fonction `tse_ikmeans`

- Expliquer ce que doit faire cette fonction par rapport aux nuages de points observés précédemment.

Cette fonction va rechercher les centre des nuages de points les plus denses dans l'espace des attributs.

- Choisir les paramètres et segmenter l'image.

Il faut choisir 3 centres car nous avons 3 classes de pixels

- Vérifier les coordonnées des centres retournés par la fonction.

Voir affichage dans la console.

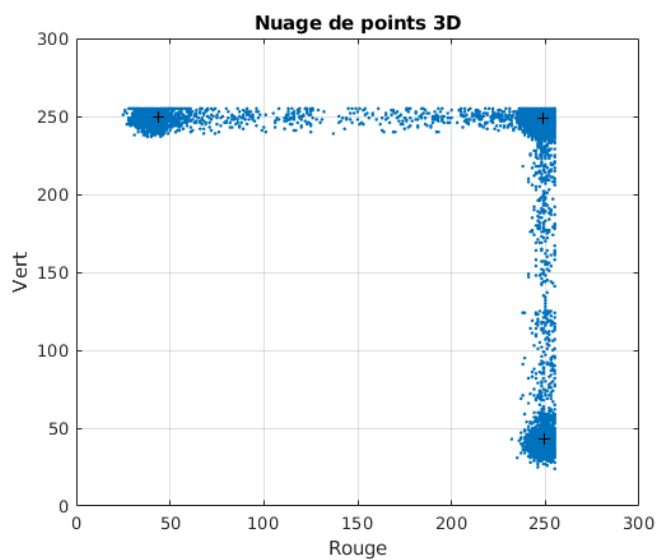
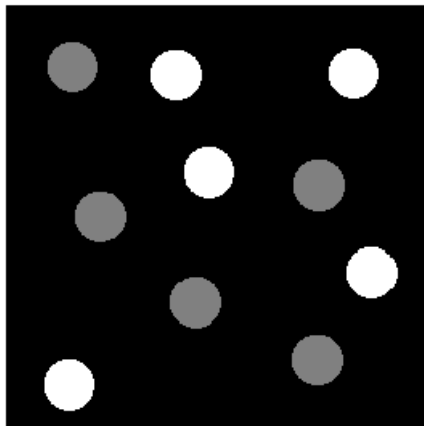
- Quel est l'effet d'un changement du nombre de classes ?

Si on surestime le nombre de classes, on ajoute des classes pour les pixels situés sur les bords des objets, dans les zones floues.

```
[fs,centers]=tse_imkmeans(f,3);
figure(5), imshow(fs,[]);
fprintf('centres :\n');
fprintf('(%d %d %d)\n',round(centers'));
```

```
figure(6)
plot(fr(:),fv(:),'. ',centers(:,1),centers(:,2),'+k'); grid('on')
title('Nuage de points 3D');
xlabel('Rouge'); ylabel('Vert'); zlabel('Bleu');
```

```
centres :
(249 249 249)
(44 249 44)
(249 43 43)
```



Question 2.3 : Images de textures

Exécuter cette section et interpréter les figures qui sont tracées :

- Que représentent les images `fmoy` et `fstd` et les nuages de points ?

`fmoy` représente une image donnant pour chaque pixel la moyenne des pixels situés dans un voisinage de taille `taille_vois`. Cela correspond à un nouvel attribut 'moyenne locale' calculé sur l'image initiale.

De même, `fstd` représente une image donnant pour chaque pixel l'écart type des pixels situés dans un voisinage de taille `taille_vois`. Cela correspond à un nouvel attribut 'écart type local' calculé sur l'image initiale.

- Interpréter le nombre et la forme des nuages de points.

La figure correspond à une projection dans l'espace 'moyenne-écart type'. On observe 2 nuages de points. Le nuage situé en haut correspond aux pixels du fond, il est plus dispersé. Le nuage du bas correspond aux pixels de l'intérieur de l'objet.

- Quel est l'effet du paramètre `taille_vois` ?

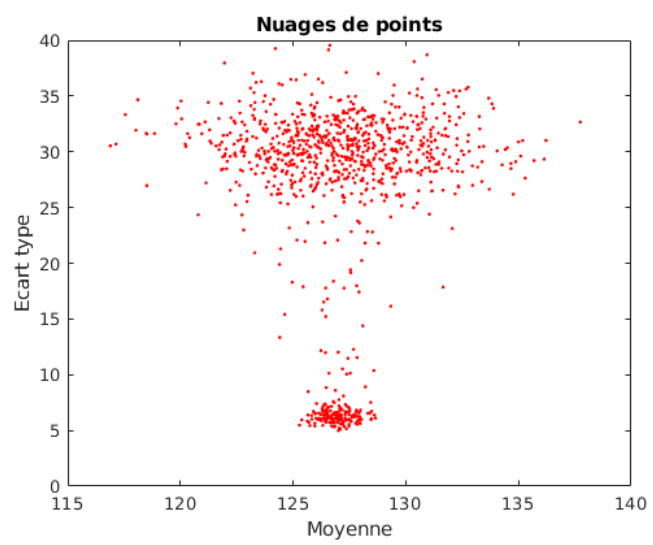
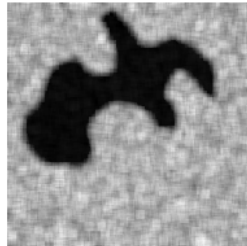
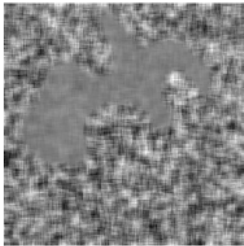
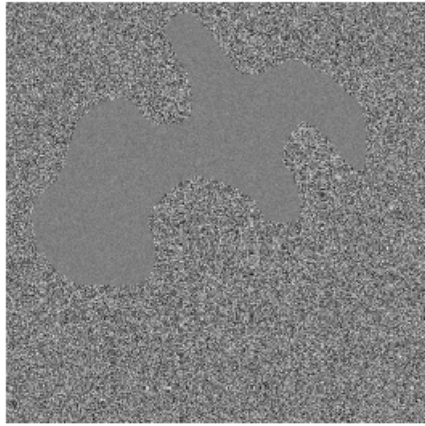
Le paramètre sert à fixer la taille du voisinage. Il doit être de taille suffisante pour bien estimer la moyenne et l'écart type locaux, mais pas trop car on va perdre de la précision pour la segmentation.

```
f=double(imread('Images/Forme2Bruits.bmp'));
figure(7),imshow(f,[]);

% Calcul des attributs
taille_vois=9;
fmoy=imfilter(f,fspecial('average',taille_vois),'symmetric');
fstd=stdfilt(f,ones(taille_vois));

% Affichage des images des attributs
figure(8),subplot(1,2,1);imshow(fmoy,[]);
subplot(1,2,2);imshow(fstd,[]);

% Tracé de l'espace des attributs
sel=ceil(numel(f)*rand(1000,1));
figure(9),plot(fmoy(sel),fstd(sel),'.r');
title('Nuages de points');
xlabel('Moyenne'); ylabel('Ecart type');
```

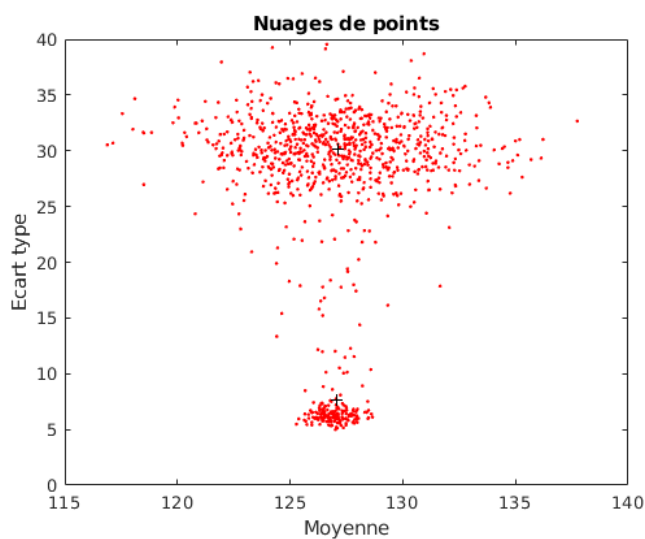


Question 2.4 Segmentation de l'image de texture

Segmenter l'image `forme2bruits.bmp` en utilisant la fonction `tse_imkmeans`. Expliquer le principe et les paramètres choisis.

On réalise une classification dans l'espace 'moyenne-écart type' avec l'algorithme du k-means. Il faut choisir 2 centres car on a deux classes de pixels.

```
f2=cat(3,fmoy,fstd);  
[fs,centers]=tse_imkmeans(f2,2);  
figure(10),imshow(fs,[]);  
fprintf('centres :\n');  
fprintf('(%d %d)\n',round(centers'));  
  
figure(9),hold on,plot(centers(:,1),centers(:,2),'+k'),hold off;  
  
centres :  
(127 30)  
(127 8)
```



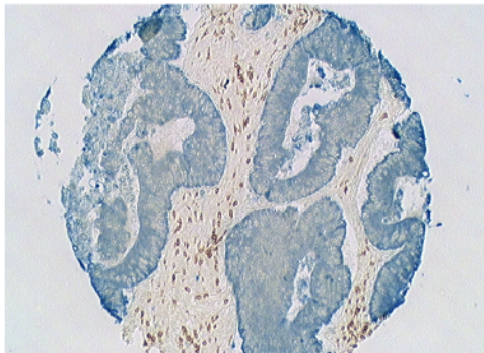
Question 2.5 : Couleur et texture

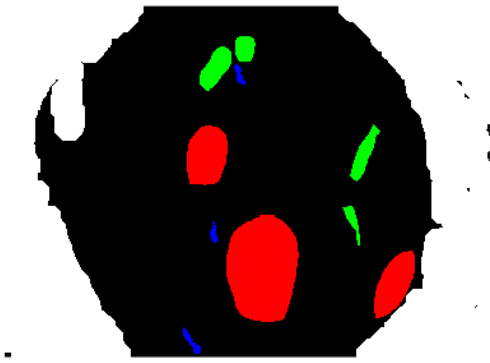
On s'intéresse à la segmentation d'une série d'images de cancer du colon (Colon_TMAxx.bmp). Certaines zones de l'image Colon_TMA01 ont été segmentées à la main par un expert qui a repéré le fond (label 1) et 3 classes de textures (labels 2, 3 et 4). L'image donnant les zones labellisées est Colon_TMA01_Label.bmp . Une image donnant un masque de ces zones est Colon_TMA01_Masque.bmp . Le but est de segmenter toutes les images en accord avec la segmentation de référence (approche supervisée).

Nous proposons de procéder en 2 étapes :

- Une étape d'apprentissage réalisée sur les zones segmentées à la main de l'image de référence (Colon_TMA01). On devra déterminer un ou plusieurs prototypes associés à chaque classe de texture définie par l'expert.
- Une étape de segmentation réalisée sur toute les images qui utilisera les prototypes précédemment déterminés pour classifier les pixels par distance minimale. Cette opération sera réalisée à partir de la fonction tse_imnearestcenter qui est fournie.

```
clear all;  
close all;  
  
f=imread('Images/Colon_TMA01.bmp');  
figure(1),imshow(f);  
  
[label,color]=imread('Images/Colon_TMA01_Label.bmp');  
figure(2),imshow(label,color);
```





1. Calcul des attributs

On affichera les points correspondant aux individus d'apprentissage de chaque classe dans un plan moyenne - écart type pour une composante couleur fixée.

- Expliquer comment choisir la taille du voisinage la plus adaptée.
- Quelle est la meilleure composante couleur ?
- Pourquoi a-t-on intérêt à garder les 3 composantes couleur (espace d'attributs à 6 dimensions) ?

% Calcul de la moyenne et de l'écart type

```
taille_vois=5;
fmoy=imfilter(double(f),fspecial('average',taille_vois),'symmetric');
fstd=stdfilt(f,ones(taille_vois));
```

% Image avec 6 composantes

```
fmoy_std=cat(3,fmoy,fstd);
```

% Label avec 6 composantes

```
label6=cat(3,label,label,label,label,label,label);
```

% Sélection des points de la classe 1

```
index1=find(label6==1);
fc1=fmoy_std(index1);
fc1=reshape(fc1,numel(index1)/6,1,6);
```

% Sélection des points de la classe 2

```
index2=find(label6==2);
fc2=fmoy_std(index2);
fc2=reshape(fc2,numel(index2)/6,1,6);
```

% Sélection des points de la classe 3

```
index3=find(label6==3);
fc3=fmoy_std(index3);
fc3=reshape(fc3,numel(index3)/6,1,6);
```

% Sélection des points de la classe 4

```
index4=find(label6==4);
```

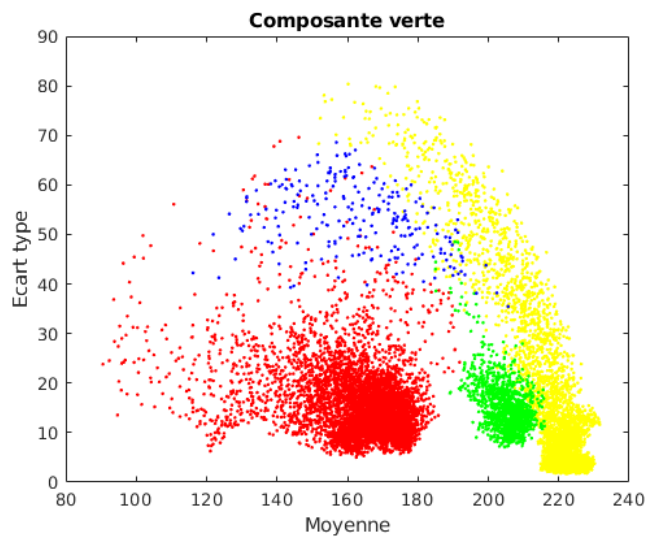
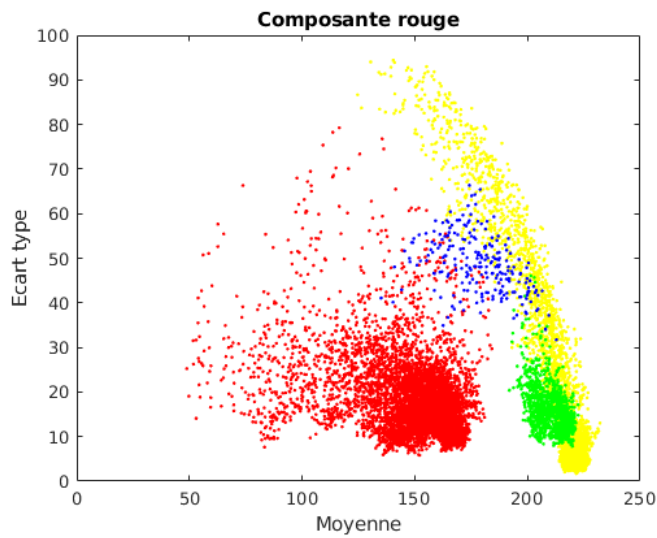
```
fc4=fmoy_std(index4);
fc4=reshape(fc4,numel(index4)/6,1,6);
```

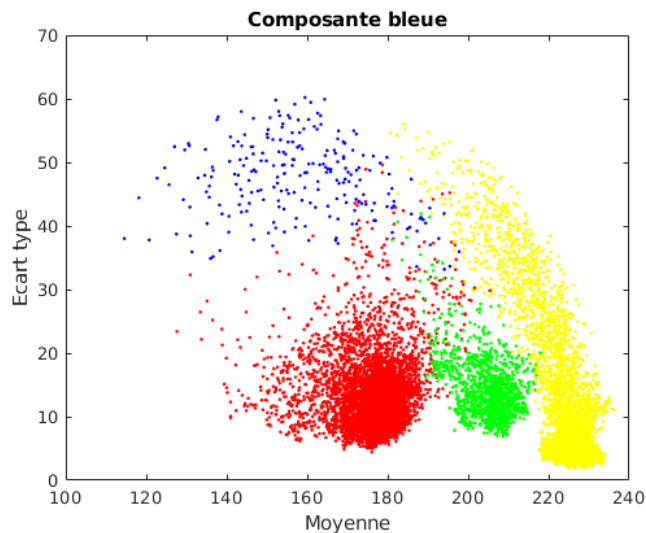
```
% Tracé des points des 4 classes
```

```
figure(3),plot(fc1(:,1,1),fc1(:,1,4),'.y',fc2(:,1,1),fc2(:,1,4),'.r',fc3(:,1,1),fc3(:,1,4),'.b',fc4(:,1,1),fc4(:,1,4),'.g');
title('Composante rouge');
xlabel('Moyenne'); ylabel('Ecart type');
```

```
figure(4),plot(fc1(:,1,2),fc1(:,1,5),'.y',fc2(:,1,2),fc2(:,1,5),'.r',fc3(:,1,2),fc3(:,1,5),'.b',fc4(:,1,2),fc4(:,1,5),'.g');
title('Composante verte');
xlabel('Moyenne'); ylabel('Ecart type');
```

```
figure(5),plot(fc1(:,1,3),fc1(:,1,6),'.y',fc2(:,1,3),fc2(:,1,6),'.r',fc3(:,1,3),fc3(:,1,6),'.b',fc4(:,1,3),fc4(:,1,6),'.g');
title('Composante bleue');
xlabel('Moyenne'); ylabel('Ecart type');
```





2. Calcul des prototypes

Choisir le nombre de prototypes adapté pour chaque classe (justifier).

— Expliquer pourquoi on peut utiliser la fonction `tse_imkmeans` pour calculer les prototypes.

% Classe 1

```
[fce,p1]=tse_imkmeans(fc1,4);
figure(6),plot(fc1(:,1,1),fc1(:,1,4),'.y',p1(:,1),p1(:,4),'.k');
```

% Classe 2

```
[fce,p2]=tse_imkmeans(fc2,2);
figure(7),plot(fc2(:,1,1),fc2(:,1,4),'.r',p2(:,1),p2(:,4),'.k');
```

% Classe 3

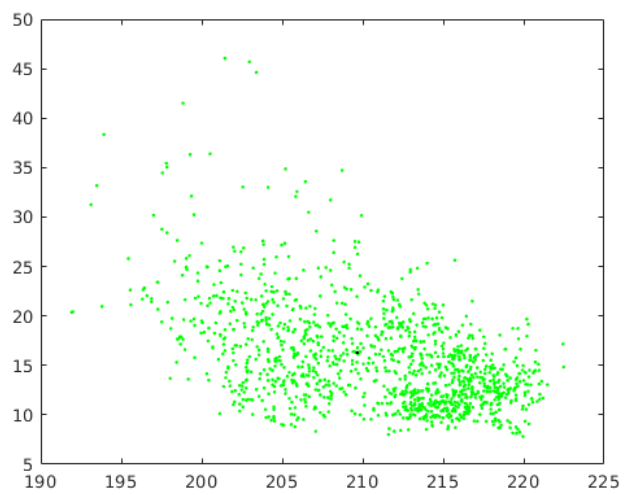
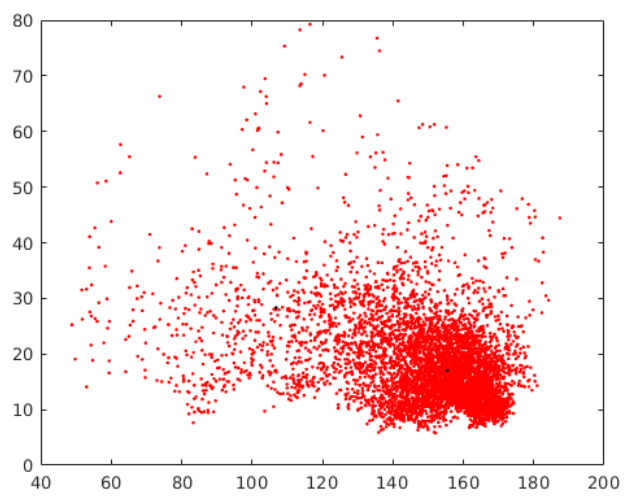
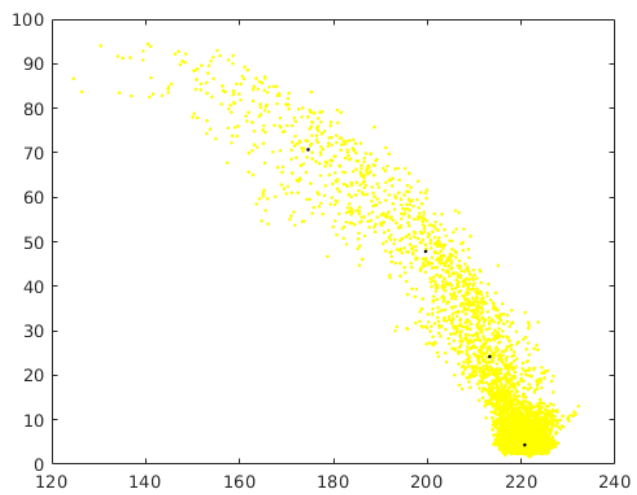
```
[fce,p3]=tse_imkmeans(fc3,1);
figure(8),plot(fc3(:,1,1),fc3(:,1,4),'.g',p3(:,1),p3(:,4),'.k');
```

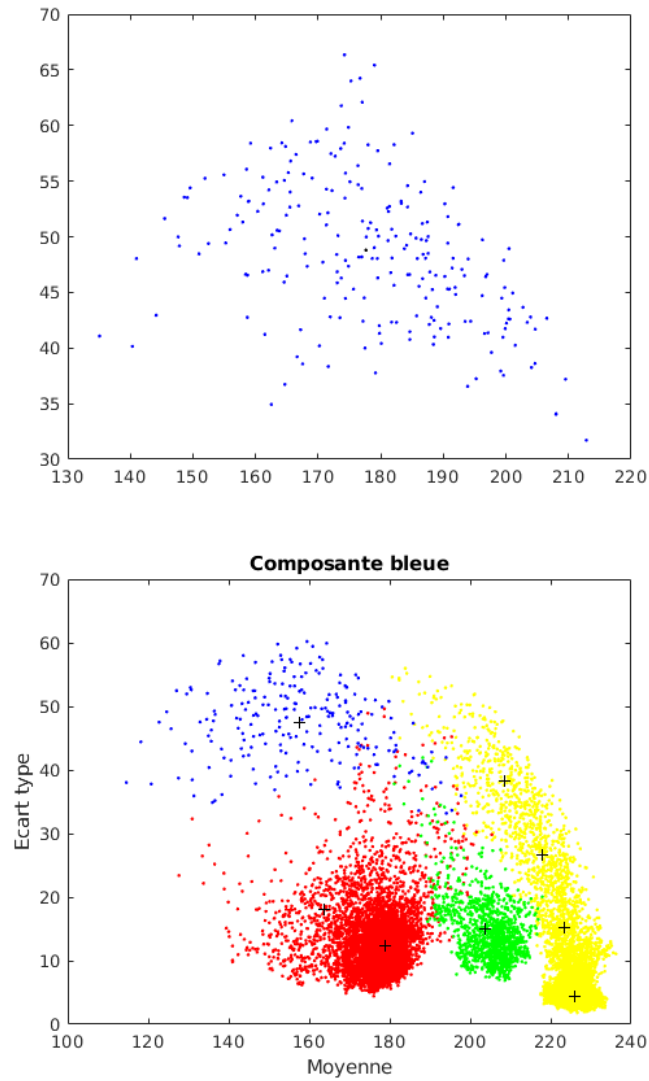
% Classe 4

```
[fce,p4]=tse_imkmeans(fc4,1);
figure(9),plot(fc4(:,1,1),fc4(:,1,4),'.b',p4(:,1),p4(:,4),'.k');
```

```
p=cat(1,p1,p2,p3,p4);
```

```
figure(10),plot(fc1(:,1,3),fc1(:,1,6),'.y',fc2(:,1,3),fc2(:,1,6),'.r',fc3(:,1,3),fc3(:,1,6),'.g',fc4(:,1,3),fc4(:,1,6),'.b');
title('Composante bleue');
xlabel('Moyenne'); ylabel('Ecart type');
```

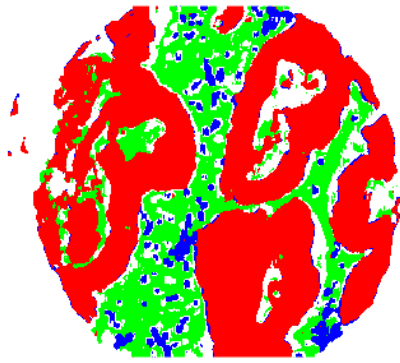




3. Affectation d'un label aux prototypes et segmentation d'une image

Affecter un label à chaque prototype et segmenter l'image de référence à l'aide de la fonction `tse_imnearestcenter`

```
fs=tse_imnearestcenter(fmoy_std,p,[1,1,1,1,2,2,3,4]);
figure(11),imshow(fs,color);
```



4. Segmentation des autres images

Segmenter les autres images à partir de même centres et évaluer qualitativement les résultats.

```
close all;
```

```
f=imread('Images/Colon_TMA03.bmp');
figure(1),imshow(f);
```

```
% Calcul de la moyenne et de l'écart type
```

```
fmoy=imfilter(double(f),fspecial('average',taille_vois),'symmetric');
fstd=stdfilt(f,ones(taille_vois));
```

```
% Image avec 6 composantes
```

```
fmoy_std=cat(3,fmoy,fstd);
```

```
% Segmentation
```

```
fs=tse_imnearestcenter(fmoy_std,p,[1,1,1,1,2,2,3,4]);
figure(2),imshow(fs,color);
```

```
f=imread('Images/Colon_TMA04.bmp');
figure(3),imshow(f);
```

```
% Calcul de la moyenne et de l'écart type
```

```
fmoy=imfilter(double(f),fspecial('average',taille_vois),'symmetric');
fstd=stdfilt(f,ones(taille_vois));
```

```
% Image avec 6 composantes
```

```
fmoy_std=cat(3,fmoy,fstd);
```

```
% Segmentation
```

```
fs=tse_imnearestcenter(fmoy_std,p,[1,1,1,1,2,2,3,4]);
figure(4),imshow(fs,color);
```

```
f=imread('Images/Colon_TMA05.bmp');
figure(5),imshow(f);
```

```
% Calcul de la moyenne et de l'écart type
```

```
fmoy=imfilter(double(f),fspecial('average',taille_vois),'symmetric');  
fstd=stdfilt(f,ones(taille_vois));
```

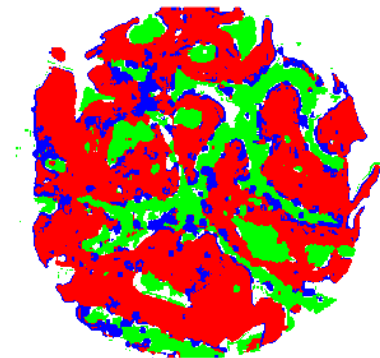
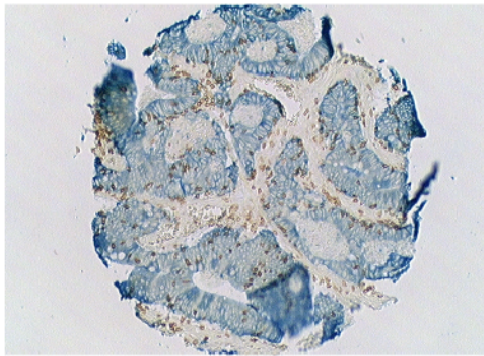
```
% Image avec 6 composantes
```

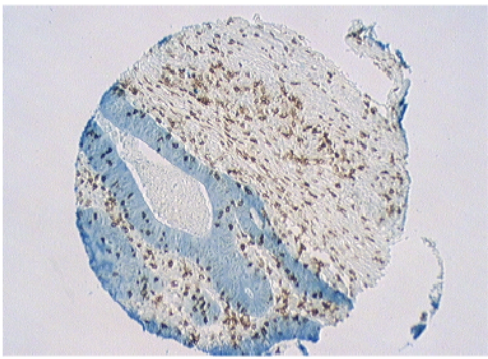
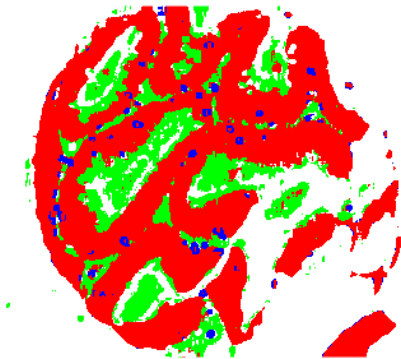
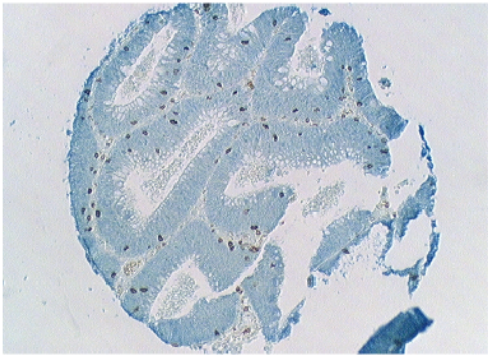
```
fmoy_std=cat(3,fmoy,fstd);
```

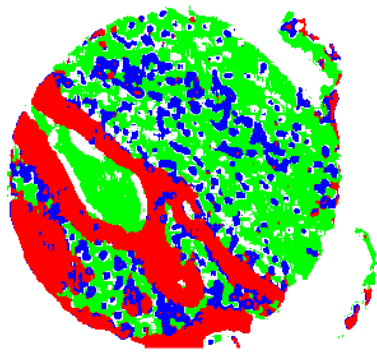
```
% Segmentation
```

```
fs=tse_imnearestcenter(fmoy_std,p,[1,1,1,1,2,2,3,4]);
```

```
figure(6),imshow(fs,color);
```







Published with MATLAB® R2017b