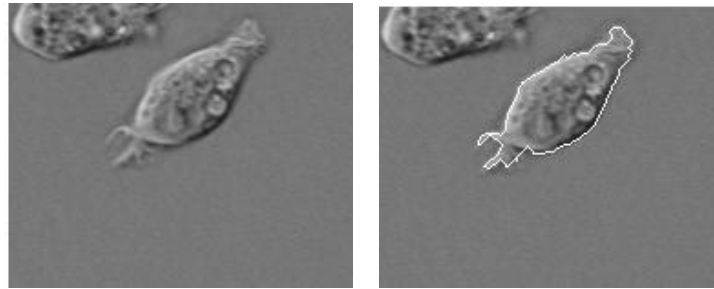


# Partiel - Algorithmes de l'imagerie

Deux heures - Documents autorisés

## Exercice 1 – Segmentation d'une image de cellule

On souhaite segmenter la cellule centrale de l'image de la figure 1-a, et obtenir un contour représentatif de cette cellule, comme illustré sur l'image de la figure 1-b.

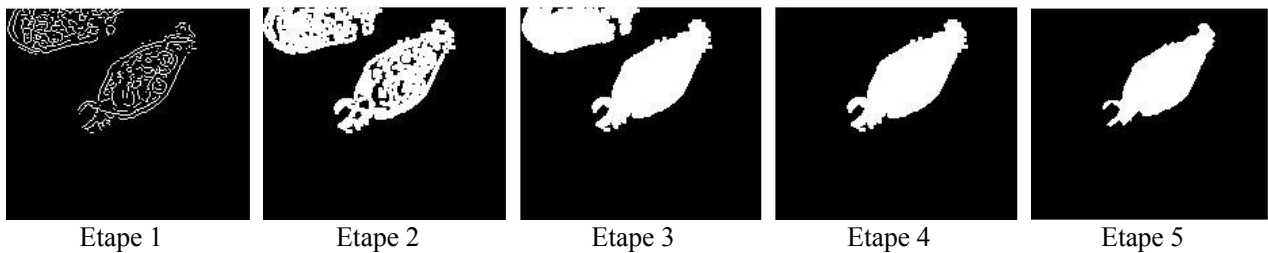


a – image originale

b- image finale

Figure 1 – une image de cellules

Pour ce faire, on se propose de passer par différentes étapes, dont les résultats sont donnés dans la figure 2.



Etape 1

Etape 2

Etape 3

Etape 4

Etape 5

Figure 2 – Etape de la segmentation de la cellule centrale

Pour chacune de ces étapes, incluant le passage de l'étape 5 au résultat final, proposer une solution algorithmique permettant d'aboutir au résultat désiré. Vous argumenterez votre réponse en fonction de la dynamique de l'image, de la 'géométrie' de l'objet à segmenter et de l'observation des images de la figure 2.

## Exercice 2 – Recherche d'objets circulaires dans une image

On se propose de calculer pour chacun des objets de l'image suivante un indice de circularité dans  $[0,1]$ , indiquant à quel point l'objet a une forme proche d'un cercle (1 s'il s'agit d'un cercle parfait)

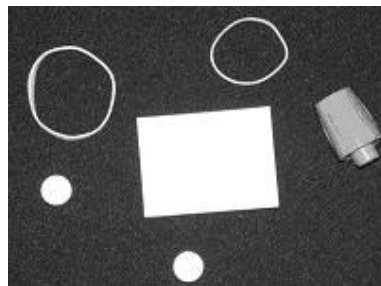


Figure 1 : image initiale

- 1- Dans un premier temps, on applique un algorithme de seuillage, permettant d'aboutir au résultat de la figure 2-a. Donner une méthode permettant de réaliser ce traitement, en justifiant par rapport à la nature et la qualité de l'image initiale

- 2- Les objets doivent maintenant être isolés (fig. 2-b) et étiquetés (fig. 2-c, chaque objet a un niveau de gris unique) avant d'être mesurés. Proposer une démarche pour aboutir à cet étiquetage (remarquer en particulier que certains pixels issus du seuillage et situés près de l'objet vert ont été supprimés).
- 3- A partir de cette dernière image, proposer une procédure algorithmique permettant de calculer, pour chacun des objets étiquetés, un indice de circularité.

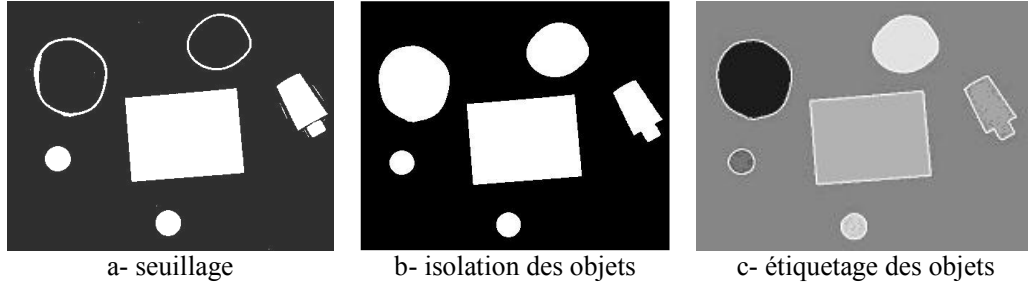


Figure 2 : étapes de l'algorithme

### Exercice 3 – Analyse d'un algorithme

Commenter de façon pertinente le programme suivant. Que réalise-t-il ? Quelles améliorations pouvez-vous lui apporter pour qu'il s'applique à un plus grand nombre de problèmes ?

**Programme** keskifaidonc

En entrée : I image, de dimensions wx\*wy, (x,y) un pixel, coulcr, coulc des niveaux de gris  
 En sortie : v ???

Variables : xc, yc, xck, coulc entiers  
 tys, tyi, txs, txi booleens  
 pilePixels pile de pixels

**Début**

```

Si (coulcr=I[x,y]) Alors
  Retourner 0 ;
Fin Si
coulc ← I[x,y];
v ← 0;

Vider(pilePixels);
Empiler(pilePixels,x,y);

Tant que (PileNonVide(pilePixels)) Faire
  Depiler (pilePixels, xc,yc);
  I[xc,yc] ← coulc;
  tys ← (yc=(wy-1));
  tyi ←(yc=0);
  txs ← (xc=(wx-1));
  txi ← (xc=0);
  Si (!tys Et abs(I[xc,yc+1]-coulc)<coulcr) Alors
    Empiler (pilePixels, xc,yc+1);
    v ← v+1;
  Fin si
  Si (!tyi Et abs(I[xc,yc-1]-coulc)<coulcr) Alors
    Empiler (pilePixels, xc,yc-1);
    v ← v+1;
  Fin si
  Si (!txs Et abs(I[xc+1,yc]-coulc)<coulcr) Alors
    Empiler (pilePixels, xc+1,yc);
    v ← v+1;
  Fin si
  Si (!txi Et abs(I[xc-1,yc]-coulc)<coulcr) Alors
    Empiler (pilePixels, xc-1,yc);
    v ← v+1;
  Fin si
Fin Tant Que
Retourner v ;

```

**Fin**

Les fonctions Empiler(pilePixels,x,y) (resp. Depiler (pilePixels, xc,yc)) empile x,y (resp. dépile xc,yc) dans (resp. de) pilePixels.