

TP 5 : Segmentation d'image

L'objectif de ce TP est de segmenter le plus précisément possible les pièces de monnaie présentes dans les images "coins.png", "coins_bruite.png", "coins-bg.png" et "coins-bg_bruite.png". Pour évaluer la qualité des segmentations que vous produirez, vous pourrez calculer le taux de superposition (i.e., coefficient Dice, fonction fournie dans le fichier *dice.m*) par rapport à la vérité terrain "GroundTruth.png". Différentes méthodes pourront être employées. À vous de trouver la méthode et les paramètres optimaux pour chacune des images à segmenter.

1. Seuillage

- (a) Afficher l'histogramme des quatre images et interpréter leurs différences (fonction : *imhist*)
- (b) En déduire un seuil pour chaque image et l'appliquer (fonction : *im2bw*)
- (c) *[facultatif]* Trouver pour chaque image le seuil qui maximise le coefficient Dice (faire une boucle *for*)
- (d) Tester la méthode d'Otsu (fonction : *graythresh*)

2. Seuillage adaptatif

- (a) Quel est l'intérêt des méthodes de seuillage adaptatif par rapport aux méthodes de seuillage classique ? Sur quelles images cela permettrait-il d'obtenir de meilleurs résultats ?
- (b) Tester la méthode de seuillage adaptatif fournie dans le fichier *adaptivethresh.m*. Trouver un jeu de paramètres satisfaisant.

3. Ligne de partage des eaux

- (a) Calculer la norme du gradient des images (fonction : *imgradient*)
- (b) Appliquer la méthode de ligne de partage des eaux sur la norme du gradient (fonction : *watershed*)
- (c) Afin de diminuer le nombre de régions segmentées, il est possible au préalable de mettre à zéros les valeurs de la norme du gradient qui sont en deçà d'un certain seuil. Syntaxe : $normGrad(normGrad < monSeuil) = 0$;

4. Contours actifs

- (a) Tester la méthode des contours actifs (fonction : *activecontour*). On prendra dans un premier temps une boîte englobante comme initialisation. Syntaxe : $mask = zeros(size(im))$; $mask(25:end-25, 25:end-25) = 1$;
- (b) Comparer l'influence des options '*Chan-Vese*' et '*edge*'.
- (c) Tester avec une meilleure initialisation. Pour ce faire, on pourra la construire en considérant une version dilatée de la vérité terrain (fonction : *imdilate*)

5. Détection de cercles

- (a) Afin d'exploiter l'*a priori* fort que les objets à détecter soient des disques, quelle méthode proposeriez vous ?
- (b) Tester cette méthode (fonction : *imfindcircles*)

6. Pré- et Post-traitements

- (a) Essayer de proposer pour chaque image la chaîne de traitements qui permet d'obtenir le meilleur coefficient Dice possible. Pour ce faire, on pourra notamment :
 - Filtrer les images initiales (fonctions : *imfilter*, *medfilt2*)
 - Post-traiter les images binaires (fonctions : *imfill*, *bwareaopen*, *imclose*, *imopen*, *imerode*, *imdilate*)
 - Combiner plusieurs méthodes de segmentation
 - Tester d'autres méthodes de segmentation