

TD3 – Traitement d'image

1. Objectif pédagogique

Le but de cette séance est d'étudier les principaux opérateurs de détection de contours. On utilisera Matlab et les fonctions et images fournies dans l'archive TD3.zip disponible sur le portail.

2. Travail à réaliser

2.1. Analyse de la fonction *tse_imgrad*

En examinant le fichier .m, analyser la fonction *tse_imgrad*. Préciser notamment :

- ce que fait cette fonction,
- le rôle des paramètres d'entrée et de sortie,
- les filtres et algorithmes utilisés pour les gradients,
- le rôle de la fonction *tse_imdetectmaxgrad* (consulter le fichier .m correspondant).

2.2. Comparaison des différents gradients

A partir de l'image *Image/ZoneCailloux.bmp*, comparer les résultats obtenus à partir du gradient de Sobel et du gradient de gaussienne (faire varier le paramètre sigma). Afficher le module du gradient et les maxima locaux détectés :

- interpréter le résultat obtenu pour les maxima locaux,
- comparer les deux gradients,
- étudier l'influence du paramètre sigma,
- choisir les paramètres adaptés à cette image.

2.3. Analyse de la fonction *tse_imhysthreshold*

Examiner le code de la fonction et préciser :

- ce que fait cette fonction,
- le rôle des paramètres d'entrée et de sortie,
- le rôle de la fonction *find_threshold*.

2.4. Seuillage hystérésis

a) A partir de la même image que précédemment, tester la fonction *tse_imhysthreshold* pour différentes valeurs des paramètres *high* et *low*. En déduire l'intérêt de ce type de seuillage.

b) Afficher les valeurs *high* et *low* calculées automatiquement par la fonction *tse_imhysthreshold*. Comparer avec les valeurs testées précédemment.

c) Proposer des valeurs de seuil adaptées à l'image.

d) Proposer une segmentation finale de l'image en ajoutant des post-traitements permettant de faire des mesures sur les objets (utiliser la fonction *imfill*).

2.5. Fermeture des contours

Les contours obtenus à partir de la détection des maxima locaux du gradient ne sont pas toujours fermés. Il peut donc être intéressant d'appliquer un post-traitement visant à fermer ces contours. C'est le rôle de la fonction *tse_imextendedge*. Nous allons étudier cette fonction en traitant l'image *Image/Cailloux.bmp*.

a) Appliquer les deux fonctions *tse_imgrad* et *tse_imhysthreshold* pour segmenter la nouvelle image. Quels défauts constatez-vous ?

b) Utiliser la fonction *tse_imextendedge* pour corriger ces défauts.

c) Proposer une segmentation finale de l'image en ajoutant des post-traitements permettant de faire des mesures sur les objets (utiliser notamment les fonctions *imfill*, et *bwmorph(f,'spur',n)*).

d) Conclure sur l'intérêt d'un détecteur de contours.

2.6. Image floue et bruitée

Toujours en utilisant une détection de contours (avec fermeture des contours et post-traitements), trouver des paramètres adaptés pour la détection des jets dans l'image *Image\Jet_diesel.bmp*.

2.7. Segmentation par ligne de partage des eaux (LPE)

a) Toujours sur la même image *Jet_diesel*, simplifier le gradient à partir de la transformation *hmin* (fonction *imhmin*) puis calculer la LPE (fonction *watershed*).

b) Faire varier la hauteur *h* et analyser les résultats obtenus. Conclure en comparant au résultat obtenu en 2.6

2.8. LPE avec sélection des bassins

Une autre approche possible pour éviter la sur-segmentation est de sélectionner des marqueurs des bassins à inonder. La procédure à suivre est alors :

- Lisser fortement l'image initiale (prendre un lissage gaussien avec un fort écart type)
- Extraire les maxima régionaux étendus de l'image à l'aide de la fonction *imextendedmax*.
- Construire des marqueurs pour l'extérieur des objets à partir d'une LPE sur l'image initiale à partir des marqueurs précédents.
- Fusionner les 2 ensembles de marqueurs.
- Calculer la LPE à partir du module du gradient et de l'ensemble des marqueurs fusionnés (appliquer la fonction *imimposemin* à l'image gradient pour imposer des minima sur les marqueurs).

a) Segmenter l'image *Jet_diesel* en appliquant la procédure ci-dessus.

b) Comparer aux résultats obtenus en 2.6 et 2.7.