

Traitement d'images (2017-2018)

Travaux pratiques - séance 1

- A la fin du TP, envoyez une archive avec ce que vous avez réussi à faire (sources + fichier readme.txt très concis) à emanuel.aldea@u-psud.fr, avec le sujet [TI5A]“nom”-“prenom”-travail-TP1 (par exemple [TI5A]Dupont-Pierre-travail-TP1). Pas de fichiers image dans l'archive.
- Avant dimanche 1 oct. minuit, envoyez à la même adresse la version finale des sources + un compte-rendu où vous détaillez bien les problèmes rencontrés, en justifiant vos solutions. Le sujet du message sera [TI5A]nom-prenom-final1-CR. De même, évitez de joindre les images de test que j'ai mises à votre disposition au début.

Exercice 1 — Détection de fissures

L'objectif du TP est d'implémenter une solution pour la détection de fissures dans les images de routes ou de murs.

1. Implémentez un programme qui applique un seuil donné en paramètre sur les niveaux de gris de l'image, les pixels inférieurs au seuil étant considérés comme appartenant à la fissure. L'utilisation du programme est `./seuilGris seuil image_entree image_binaire_sortie`.
2. Implémentez un programme qui applique un filtre de Sobel horizontal et vertical et qui calcule la magnitude du gradient. L'image de gradient est transformée dans une image binaire (valeurs 0 et 255 uniquement) grâce à un seuil que vous choisissez vous mêmes (à justifier dans le CR). L'utilisation du programme est `./sobel image_entree image_binaire_sortie`
3. Implémentez un programme qui calcule la qualité de votre détection à partir d'une image binaire de résultat et d'une vérité terrain. Pour cela on calcule la précision p , la proportion de pixels détectés qui appartiennent réellement à la fissure :

$$p = \frac{TP}{TP + FP}$$

et le recall, la proportion de pixels de fissure qui apparaissent dans votre résultat :

$$r = \frac{TP}{TP + FN}$$

L'utilisation du programme est :

`./evaluation image_binaire_verite_terrain image_binaire_sortie`

Ce programme doit afficher dans la console les deux valeurs p et r , une par ligne, et sans d'autres choses.

Ici TP (true positives) représente le nombre de pixels détectés qui sont bien des pixels de fissure, FP (false positives) représente le nombre de pixels détectés qui ne sont pas des pixels de fissure, et FN (false negatives) représente le nombre de pixels de fissure qui ne se retrouvent pas parmi les pixels détectés.

4. Implémentez un programme qui essaye d'obtenir les meilleurs résultats de détection possibles en utilisant une combinaison de techniques de filtrage. Quelques solutions possibles (liste non exhaustive) :
 - un filtrage passe bas pour enlever les très hautes fréquences sans atténuer les fissures
 - utilisation du Laplacien

- le calcul de contours orientés et la propriété suivante d'une fissure : gradient fort dans une direction et faible dans la direction perpendiculaire
- calcul de la composante basse fréquence de l'image ou d'un filtrage médian, et soustraction de l'image initiale