

TP5 Algorithmique des images, textes et données

Détection de contours

Le but de ce TP est d'implanter les différents détecteurs de contours vus en cours (naïf, Sobel, Canny).

Dans ce TP on suppose que l'on utilisera la structure `picture` définie dans les TP précédents. On se restreindra dans un premier temps à des images en nuances de gris (`.pgm`).

1. Créer des fonctions `picture *naive_x(picture *image)` et `picture *naive_y(picture *image)` qui renvoient l'image correspondant respectivement au détecteur de contours verticaux et horizontaux.
2. Créer une fonction `picture *naive_edge_detector(picture *image)` permettant d'implanter le détecteur de contours naïf. On fera bien attention à normaliser la valeur des pixels dans `[0, 255]`.
3. En suivant le même raisonnement créer des fonctions `picture *sobel_edge_detector(picture *image)` implantant le filtre de Sobel.
4. Créer une fonction `float **gradient_angle(picture *grad_x, picture *grad_y)` qui prend en paramètre les deux composantes du gradient et renvoie l'angle de celui ci avec la tangente horizontale. On utilisera la fonction `atan2` de la bibliothèque `math.h`
5. Créer une fonction `unsigned char interpolation(float x, float y, picture *image)` permettant d'interpoler bilinéairement la valeur du gradient aux coordonnées `x, y`.
6. Créer une fonction `void non_maxima_suppression(picture *norm, float **angle)`, la fonction prendra en paramètre une image correspondant à la norme du gradient, et un tableau 2D de `float` content l'angle du gradient avec la droite horizontale à chaque coordonnée.

Afin de pouvoir implémenter le seuillage par hystérésis, nous devons lister les pixels étant des contours certains. Étant donné que l'on ne peut pas connaître leur nombre à l'avance, nous allons utiliser une structure de liste chaînée.

7. Créer une structure de liste chaînée avec des fonctions pour ajouter, supprimer des éléments.
8. Créer une fonction `void hysteresis_thresholding(picture *image)` implantant le seuillage par hystérésis. Les valeurs des seuils haut et bas seront définis dans des macros.
9. Créer `picture *canny_edge_detector(picture *image)` implantant le filtre de Canny.
10. Modifier vos fonctions précédentes de sorte à prendre également en charge des images `RGB` – i.e. au format `ppm`.