

TP5 : Segmentation d'images à l'aide de l'algorithme de Canny et de la détection de contours

Dans ce TP, vous devez traiter une image (« forma.png ») en trois étapes en utilisant OpenCV et matplotlib :

1. Détection des contours avec Canny

- Charger l'image et la convertir en niveaux de gris.
- Appliquer l'algorithme Canny en choisissant des seuils bas (ex. 10) et haut (ex. 200) pour obtenir une image binaire des contours.
- Afficher l'image originale en niveaux de gris et l'image résultant de la détection des contours.

(en règle général il est souhaitable d'appliquer un filtre flou afin d'aider l'algorithme canny à mieux détecter les bords. Il n'est pas nécessaire vu le caractère de l'image simple qui est fournie)

2. Extraction et affichage des contours

- À partir de l'image binaire, utiliser la fonction `cv2.findContours` pour extraire la liste des contours (avec le mode `cv2.RETR_EXTERNAL` et la méthode `cv2.CHAIN_APPROX_SIMPLE`).
- Dessiner ces contours sur une copie de l'image d'origine (par exemple en traçant les contours en vert).
- Afficher trois images côte à côte : l'image originale (convertie en RGB pour matplotlib), l'image des contours (binaire) et l'image avec les contours extraits.
- Afficher le nombre total de contours détectés.

(La fonction `cv2.findContours` implémente principalement l'algorithme de Suzuki Abe et Suzuki, souvent appelé l'algorithme de "border following" (suivi de contours). Cet algorithme recherche les contours dans une image binaire en détectant et en suivant les frontières (ou bords) des objets présents dans l'image.)

3. Segmentation finale et numérotation des contours

- Créer une nouvelle image (initialisée à zéro) de la même taille que l'image d'origine.
- Pour chaque contour sélectionné, procéder comme suit :
 - Générer une couleur aléatoire et remplir le contour avec cette couleur sur la nouvelle image.
- Utiliser la fonction `cv2.boundingRect` pour obtenir le rectangle englobant le contour.
- Calculer le centre de ce rectangle en utilisant la formule :
$$cX = x + w // 2$$
$$cY = y + h // 2$$

où (x, y) correspond aux coordonnées du coin supérieur gauche et w et h aux dimensions du rectangle.
- Placer un numéro (commençant par 1) à l'intérieur ou à côté de l'objet, en utilisant ce centre pour positionner le texte.

- Afficher côte à côte l'image originale (en RGB) et l'image de segmentation finale.