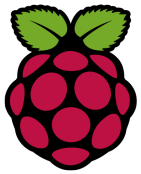
Raspberry Pi OpenCV Trouver les contours



1 Trouver les contours des objets (findContours)

La fonction findContours permet de trouver les contours en remplissant un vecteur de point qui correspond aux points qui sont définies sur le contour extérieur des objets. En fait, plusieurs constantes sont définies dans OpenCV et permettent de trouver les points extérieurs de l'objet mais aussi ceux qui font parties des « trous » internes de l'objet. On peut aussi vouloir créer une hiérarchie d'objet en passant au travers d'une liste des contours extérieurs jusqu'au contours intérieurs.

2 Syntaxe de la fonction findContours

void findContours(InputOutputArray **image**, OutputArrayOfArrays **contours**, OutputArray **hierarchy**, int **mode**, int **method**, Point **offset**=Point())

- **image** – C'est l'image source sur 8 bits et un canal. L'image est traitée en format binaire, ce qui veut dire qu'elle doit avoir subi une binarisation. Vous devrez donc utiliser une fonction parmi les suivantes : **inRange**(), **threshold**(), **Canny**().

- **contours** – Le vecteur de vecteurs de points qui constitue l'ensemble des contours trouvés. Ainsi, contours[0] représente la séquence de points de l'objet « 0 ». contours[1] représente la séquence de points de l'objet « 1 » et ainsi de suite.

- **Hierarchy** – C'est le vecteurs de contours qui permet d'obtenir la hiérarchie des contours d'un objet. Ainsi, pour un contour i, hierarchy[i][0] est le contours extérieurs, hierarchy[i][1] est le deuxième contours intérieurs de l'objet et ainsi de suite. Si les objets n'ont pas de contours intérieurs, ce paramètre n'est pas utilisé.

- **mode** – Le mode de récupération du contour. Une constante parmi les suivantes :

  - **CV_RETR_EXTERNAL** récupère le contour extérieur seulement.

  - **CV_RETR_LIST** récupère tous les contours sans établir de ealation hiérarchique.

  - **CV_RETR_CCOMP** récupère tous les contours et les organise dans une hiérarchie à deux niveaux. Le premier niveau constitue le contour extérieur alors que le deuxième niveau constitue le contour des trous présents dans l'objet.

- **methode** – La méthode d'approximation du contour. Utilisez l'une des constantes suivantes :

  - **CV_CHAIN_APPROX_NONE**
  - **CV_CHAIN_APPROX_SIMPLE**
  - **CV_CHAIN_APPROX_TC89_L1**
  - **CV_CHAIN_APPROX_TC89_KCOS**

- **offset** – Le point de départ qui constitue le premier point du contour. Par défaut, c'est le point de coordonnée (0,0). Peut-être utile lorsqu'on travaille avec des régions d'intérêt.

3 Les étapes de la détection de contours

Voici les étapes habituellement réalisées pour trouver les contours des objets :

- **Étape 1**: Définir un vecteur de vecteur de point. vector<vector<Point>> contours;

- **Étape 2** : Charger l'image en ton de gris (très suggéré) ou la transformer en ton de gris par la suite si vous l'avez chargé initialement en couleur.

- **Étape 3** : Appliquer un « blurring » pour enlever le bruit. Binariser l'image en filtrant uniquement les contours. Il s'agit habituellement d'appliquer la fonction Canny.

- **Étape 4** : Appliquer la fonction « findContours ».
  findContours(ImgCanny, Contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE);

- **Étape 5**: Trouver les caractéristiques des objets.
  Par exemple, le nombre d'objet total trouvé correspond au nombre d'élément dans le vecteur. Ainsi, « contours.size() » donnera le nombre d'objet total dans l'image.

4 programme exemple

Vous trouverez un exemple de programme à compiler sur mon dépôt Github.

https://github.com/PhilippeSimier/openCV/tree/master/DetectionContours

Le programme charge l'image blob.jpg qui contient 20 objets de forme et couleur différentes.

Les objets trouvés dans l'image sont ensuite dessiné un par un à chaque fois que l'on appuie sur une touche du clavier. Les caractéristiques principales de chaque objet sont affichées dans la console (aire et périmètre).

Fiche 43 Raspberry OpenCV Trouver les contours