

OpenCV & Python

Justine Senellart

Fonctions générales

cv2.imread() : Charge une image depuis un fichier
cv2.imwrite() : Enregistre une image
cv2.imshow() : Affiche une image dans une fenêtre
img[y :y+h, x :x+w] : Sélectionne une région d'intérêt

Filtres

cv2.filter2D() : Filtre linéaire non séparable
cv2.sepFilter2D() : Filtre linéaire séparable
cv2.boxFilter(), *cv2.GaussianBlur()*, *cv2.medianBlur()*,
cv2.bilateralFilter() : Lisse une image avec un filtre linéaire ou non linéaire.

Histogrammes

cv2.calcHist() : Calcule l'histogramme d'une image
cv2.calcBackProject() : Projette l'histogramme
cv2.equalizeHist() : Normalise le contraste et la luminosité d'une image
cv2.compareHist() : Compare deux histogrammes

Transformation géométrique

cv2.resize() : Redimensionne l'image
cv2.getRectSubPix() : Extrait un patch
cv2.warpAffine() : Déforme une image affinement
cv2.warpPerspective() : Déforme une image perspective-ment
cv2.remap() : Déformation d'image générique

Diverses transformations d'images

cv2.cvtColor() : Convertit une image d'un espace de couleur à l'autre
cv2.threshold() : Convertit une image niveau de gris en image binaire
cv2.adaptiveThreshold() : Utilise un seuil fixe ou variable
cv2.floodFill() : Trouve un composant connecté en utilisant un algorithme de croissance de région
cv2.integral() : Calcule l'intégral d'une image
cv2.distanceTransform() : Construit une carte de distance ou un diagramme discret de Voronoi pour une image binaire
cv2.watershed(), *cv2.grabCut()* : Algorithmes de segmentation d'image à base de marqueurs

Gradient

cv2.Laplacian() : Calcule le Laplacien d'une image
cv2.Sobel() : Calcule la première, deuxième ou troisième dérivée de l'image en utilisant l'opérateur de Sobel

Opérations arithmétiques

cv2.bitwise_and(), *cv2.bitwise_or()*, *cv2.bitwise_xor()* : Opérations booléennes sur deux images
cv2.add() : Addition de deux images
cv2.addWeighted() : Mélange de deux images
cv2.subtract() : Soustraction de deux images

Transformation morphologique

cv2.getStructuringElement() : Retourne un élément de la taille et la forme spécifiées pour des opérations morphologiques
cv2.morphologyEx() : Effectue une transformation morphologique avancée (ouverture, fermeture ...)
cv2.erode() : Opération d'érosion sur l'image spécifiée en utilisant l'élément structurant spécifié
cv2.dilate() : Opération de dilatation sur l'image spécifiée en utilisant l'élément structurant spécifié

Contours

cv2.findContours() : Trouve les contours dans une image binaire
cv2.moments() : Calcule les moments jusqu'au troisième ordre d'un polygone
cv2.contourArea() : Calcule la surface des contours
cv2.boundingRect() : Calcule le rectangle droit d'un ensemble de points
cv2.minEnclosingCircle() : Calcule le plus petit cercle entourant un ensemble de points
cv2.minAreaRect() : Trouve le plus petit rectangle pivoté entourant un ensemble de points
cv2.fitEllipse() : Adapte une ellipse autour d'un ensemble de points
cv2.fitLine() : Adapte une ligne à un ensemble de points
cv2.approxPolyDP() : Approxime un polygone avec la précision donnée

Détection de features

cv2.Canny() : Détection des contours avec l'algorithme Canny86
cv2.cornerHarris() : Détection des coins en utilisant l'algorithme de Harris
cv2.HoughCircles() : Trouve les cercles présents dans une image niveau de gris
cv2.HoughLines() : Trouve les lignes présentes dans une image binaire

Interface homme-machine

cv2.namedWindow() : Crée une fenêtre nommée
cv2.destroyWindow() : Ferme la fenêtre spécifiée
cv2.destroyAllWindows() : Ferme toutes les fenêtres
cv2.moveWindow() : Déplace une fenêtre à la position spécifiée
cv2.resizeWindow() : Modifie la taille d'une fenêtre
cv2.waitKey() : Attend l'appui sur une touche du clavier