

Outils d'imagerie pour la robotique

TP détection d'objets

Résumé

Le but de cette séance est de détecter les objets en mouvement dans une vidéo

1 Lecture d'un vidéo avec OpenCV

En vous servant de la documentation de la classe `VideoCapture` d'OpenCV, écrire un programme permettant de lire une vidéo dont le nom est passé en paramètre.

Récupérer ensuite successivement les images de la vidéo et afficher les dans une même fenêtre à l'écran. Tester sur les séquences `seq4.avi` et `tabletennis.avi`

2 Détection du mouvement

2.1 Différence d'images successives

Détecter et afficher les pixels en mouvement en calculant la différence entre les images successives. Des fonctions pour la différence et le seuillage sont à votre disposition dans OpenCV. Faire varier les paramètres pour voir l'influence sur les résultats.

2.2 Soustraction de fond

Comparer les résultats obtenus avec ceux obtenus en utilisant directement les méthodes d'analyse du mouvement et suivi d'objets (Motion Analysis and Object Tracking) décrites dans la documentation OpenCV. Observer en particulier l'influence d'une représentation du fond par des mélanges de gaussiennes.

3 Extraction des objets

En utilisant les fonctions d'érosion et de dilatation disponibles dans OpenCV, observer la différence entre l'ouverture (érosion puis dilatation) et la fermeture (dilatation puis érosion) morphologique. Quelles sont les conséquences de chacune de ces opérations? Pourquoi?

Pour finir, extraire les composantes connexes en utilisant la fonction `findContours`.

4 Champ de déplacement

Tester une méthode de calcul de champ de déplacement dense (avec `calcOpticalFlowFarneback`). Visualiser les résultats avec la fonction suivante :

```

void drawOptFlowMap(const Mat& flow, Mat& cflowmap, int step,
    const Scalar& color)
{
    for(int y = 0; y < cflowmap.rows; y += step)
        for(int x = 0; x < cflowmap.cols; x += step) {
            const Point2f& fxy = flow.at<Point2f>(y, x);
            line(cflowmap, Point(x,y), Point(cvRound(x+fxy.x), cvRound(y+fxy.y)), color);
            circle(cflowmap, Point(x,y), 2, color, -1);
        }
}

```

5 Suivi de points par KLT

Observer et tester le programme KLT. Que fait-il ?