

Descripteurs locaux invariants pour le suivi d'objet 2D de forme libre

Partie 1 : Détection de points d'intérêt et extractions des descripteurs des images de référence

L'objectif de ce travail est de réaliser une application de tracking dédiée aux systèmes de vision temps réel. L'approche implémentée se base sur la détection et l'extraction des points d'intérêt dans l'image. Ensuite, des descripteurs sont calculés afin de caractériser ces points et de les appairer avec leurs homologues dans une base d'images de référence.

Le programme ouvre le flux de la vidéo et capture les images instantanément, ce sont les images requêtes. Une base d'images de référence est constituée d'images sauvegardées sur le disque et qui représentent les images d'apprentissage. Le même traitement est appliqué aux images requêtes et aux images de références pour détecter et calculer les descripteurs. Dans la première étape qui est réalisée hors ligne, il est indispensable de charger en mémoire tampon les images d'apprentissage et, ensuite, détecter et calculer les points saillants et leurs attributs locaux invariants. Pour réaliser ces opérations, les traitements suivants doivent être effectués :

- 1.1. Définir une fonction qui charge toutes les images de référence. Cette fonction permet la lecture et la sauvegarde temporaire de ces images dans un buffer en mémoire.
- 1.2. Calculer les points d'intérêt de toutes les images de référence ainsi que leurs descripteurs, en utilisant l'approche **ORB**.

Partie 2 : Appariement des descripteurs

Dans cette partie, il est nécessaire de réaliser le même traitement de la partie 1 (détection de points d'intérêt et calcul de descripteurs) pour les images requêtes afin de permettre leurs appariements avec les images de références. Dans la boucle principale d'acquisition des images du flux vidéo, réaliser les traitements suivants :

- 2.1. Détecter les points d'intérêt de l'image capturée et calculer les descripteurs locaux associés à ces points en utilisant ORB.
- 2.2. Définir une fonction de calcul d'appariement qui vérifie si la mise en correspondance entre les descripteurs est pertinente en termes de distance. Dans cette fonction, l'algorithme du Plus Proche Voisin **KNN** est utilisé. KNN cherche pour chaque descripteur de point d'intérêt, le vecteur descripteur le plus proche.
- 2.3. Définir une fonction de validation d'appariement de descripteurs qui appelle la fonction d'appariement définie précédemment (définie dans 2.2). Si le nombre d'appariements retourné par la fonction d'appariement (définie dans 2.2) est supérieur à 4, alors calculer l'homographie qui relie l'ensemble des points d'intérêt de l'image requête et celle de référence. Enfin, si la distance entre deux points des deux images est inférieure à 3px alors ces deux points sont considérés comme des vrais positifs (inlier) et ils sont retenus. Cette fonction de validation d'appariement retourne le nombre de vrais positifs.

Partie 3 : Classification d'image et rendu visuel

Dans cette partie, l'algorithme classe l'image requête en utilisant le nombre de vrais positifs retournés par la fonction de validation d'appariement de descripteurs (définie dans 2.3).

- 3.1. Écrire les instructions permettant de trouver si l'image requête est reconnue et retourner son identifiant.
- 3.2. En fonction de l'identifiant de l'image, afficher l'image de référence correspondante sur l'image requête (figure 1). Cet affichage doit être maintenu tant que l'image de référence est identifiée.
- 3.3. Définir une fonction qui affiche les mises en correspondances, sous forme de ligne, entre les points d'intérêt des deux images. Cet affichage est effectué dans une nouvelle fenêtre graphique (figure 2).
- 3.4. Ajouter un curseur de défilement pour contrôler l'avancement du flux vidéo. À la fin de la vidéo, le flux se repositionne au début de la séquence vidéo et ne démarre que lorsqu'une touche clavier est appuyée.



Figure 1 - Identification d'image.



Figure 2 - Appariements des points entre les images.