ROB313 - Vision Robotique
TP Matrice Fondamentale

Décembre 2020

1 Introduction

Dans ce TP, nous allons travailler sur l'estimation de la matrice fondamentale qui encode la géométrie épipolaire reliant deux images prises par une caméra en mouvement.

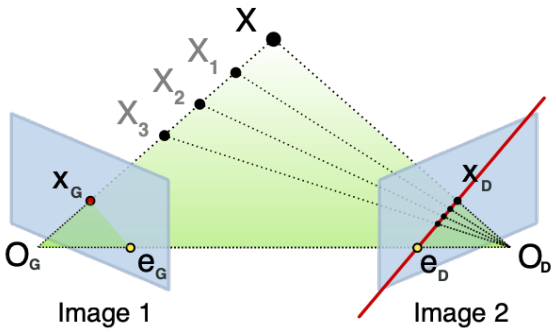

FIGURE 1 – Illustration de la géométrie épipolaire (Wikipedia).

Pour cela, nous utiliserons la bibliothèque de traitement d'images OpenCV sous Python (code testé avec python3 et OpenCV 4.1.0). Le code nécessaire pour le TP et des images de test sont disponibles sur la page du cours.

2 Paramètres importants

Dans cette première partie, on expérimentera simplement la méthode d'estimation de la matrice fondamentale en utilisant la fonction d'OpenCV avec RANSAC. La figure 2 illustre les lignes épipolaires calculées à partir de la matrice fondamentale F correcte. Vous pouvez notamment constater que les lignes dans une image convergent vers le centre de la caméra qui a pris l'autre image.

Le code fourni réalise cette estimation avec des points d'intérêt KAZE utilisant les paramètres par défaut et une mise en correspondance fondée sur un critère de plus proche voisin entre les paires de descripteurs, plus une condition de ratio maximum entre cette distance et celle au second plus proche voisin ("ratio test"). A partir de ces correspondances, il calcule la matrice fondamentale avec la méthode RANSAC :

```
FRansac, mask = cv2.findFundamentalMat(pts1,pts2,cv2.FM_RANSAC)
```

— Observer, pour plusieurs paires d'image, les lignes épipolaires calculées ainsi que les résultats intermédiaires (points détectés, appariements sélectionnés).
— Identifier les paramètres qui semblent les plus déterminants dans la qualité du résultat final.

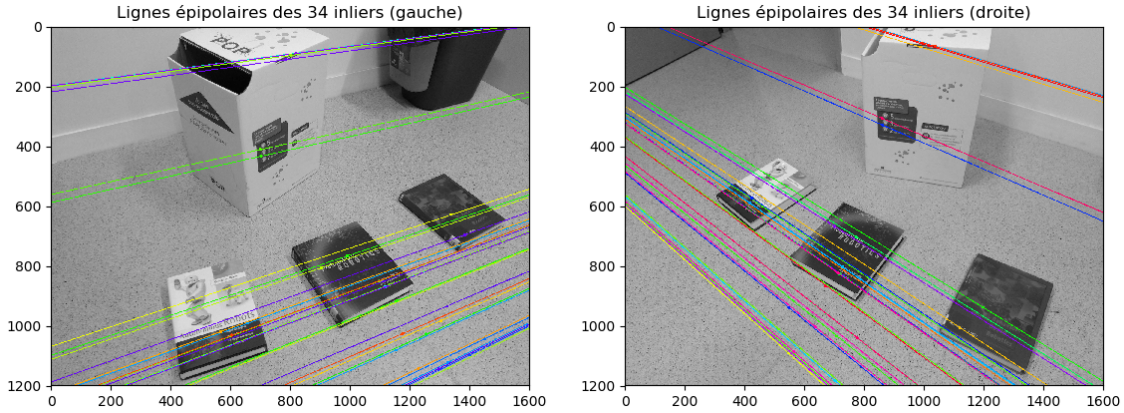

FIGURE 2 – Illustration de la géométrie épipolaire attendue pour les deux images d'exemples.

3 Rectification des images

— Rectifier les images en calculant d'abord les deux homographies rendant les deux faisceaux de droites épipolaires parallèles, grâce à la fonction OpenCV `cv2.stereoRectifyUncalibrated`, puis en appliquant les homographies correspondantes avec la fonction `cv2.warpPerspective`.

— Identifier les difficultés potentielles selon la nature du déplacement et des images.