

ROB317 - Vision 3d

TP Matrice Fondamentale

Septembre 2024

1 Introduction

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

Nous utiliserons cette matrice fondamentale pour rectifier la paire d'images de façon à se ramener à un cas de stéréovision idéale, dans lequel nous tenterons d'estimer une carte de profondeur.

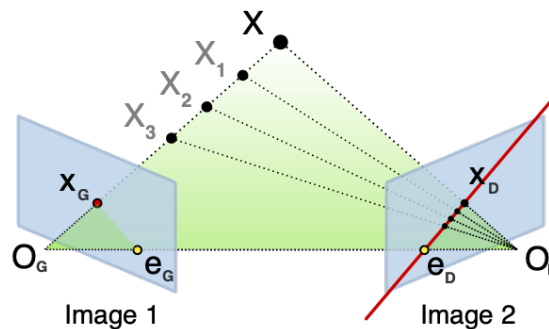


FIGURE 1 – Illustration de la géométrie épipolaire (Wikipedia).

Pour cela, nous utiliserons la bibliothèque de traitement d'images OpenCV sous Python (code testé avec python3 et OpenCV 4.1.0).

Un code Python de base est fourni :

https://perso.ensta-paris.fr/~manzaner/Cours/ROB317/TP_Fondamentale.py

ainsi que des images de test :

https://perso.ensta-paris.fr/~manzaner/Cours/ROB317/Images_Fondamentale.tgz

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 :

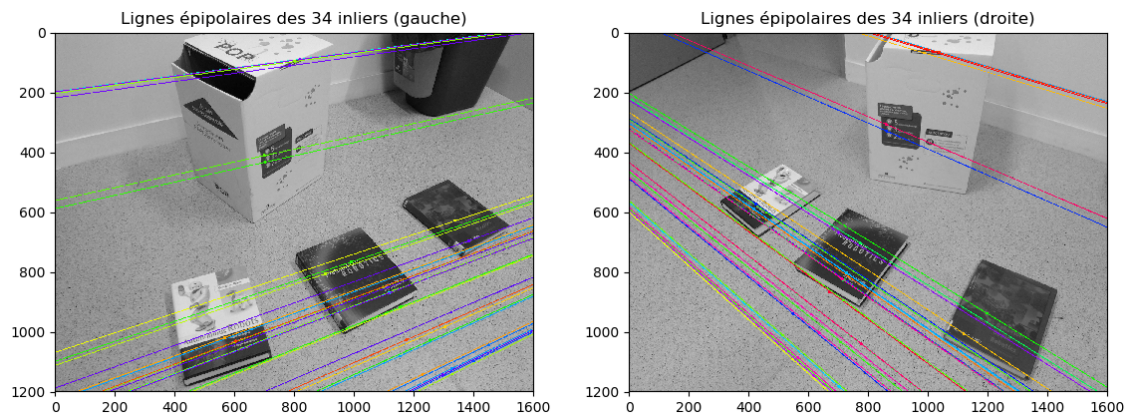


FIGURE 2 – Illustration de la géométrie épipolaire attendue pour les deux images d'exemples.

```
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.

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.

4 Reconstruction 3d

- Essayer de construire une carte de disparité en utilisant une fonction d'appariement (flot optique) d'OpenCV entre les deux images rectifiées. Que devrait-on observer? Comment sélectionner les appariements corrects?
- Dédurre de cette carte une carte de profondeur relative. Quelles informations faut-il ajouter pour obtenir une métrique absolue? Comment peut-on obtenir un nuage de points $\{M_i = (X_i, Y_i, Z_i)\}$ à partir de la carte de profondeur?