

# Estimation de Matrice Fondamentale - RANSAC

David FILLIAT - Antoine MANZANERA  
Cours ROB313 - ENSTA Paris

10 décembre 2019

## 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 en utilisant la méthode RANSAC.

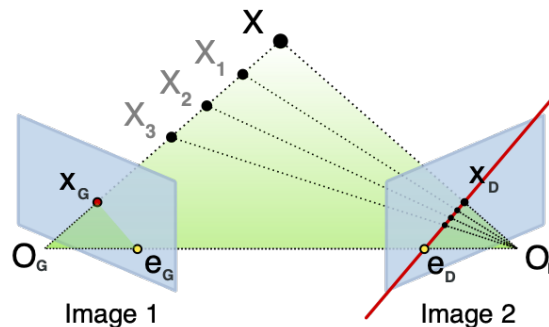


FIGURE 1 – Illustration de la géométrie épipolaire (Wikipedia).

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 est disponible sur la page du cours.

Vous devez rédiger un rapport que vous enverrez au format PDF. Utilisez le format `Nom_Rapport_Fondamentale.pdf` pour nommer votre rapport.

## 2 Question 1 - Influence des détections de points d'intérêt

Dans cette première partie, nous allons utiliser la méthode d'estimation de la **matrice fondamentale** avec **RANSAC** de OpenCV. 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 seuil sur la distance au plus proche voisin, et 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)
```

Vous verrez que le **résultat avec le code fourni** ne correspond pas au résultat attendu car les **mis en correspondance** de points d'intérêt ne sont pas suffisamment de bonne qualité. **Faites varier les différents seuils** de mise en correspondance proposés et les paramètres du détecteur de points d'intérêt jusqu'à obtenir

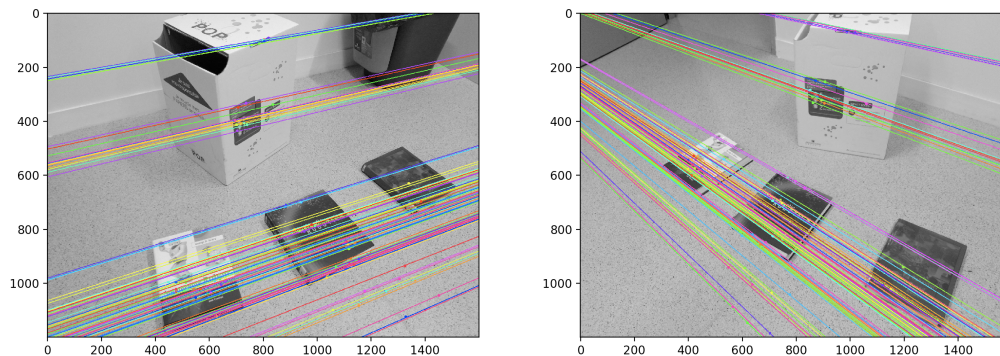


FIGURE 2 – Illustration de la géométrie épipolaire attendue pour les deux images d'exemples.

une **matrice fondamentale correcte**. Quels sont les critères qui vous semblent les plus importants pour obtenir une bonne solution ?

Vous pouvez dans cette question afficher les points détectés sur la première image, ainsi que les appariements réalisés entre les deux images en décommentant les groupes de lignes correspondants dans le code.

### 3 Question 2 - Implémentation de la méthode RANSAC

Dans cette deuxième partie, ils vous est demandé d'implémenter vous-même la méthode RANSAC. Pour cela, vous utiliserez uniquement le calcul de la matrice fondamentale à partir de l'algorithme de 7 points :

```
FRansac, mask = cv2.findFundamentalMat(pts1,pts2,cv2.FM_7POINT)
```

Pour rappel, voici l'algorithme proposé pour la méthode RANSAC :

- Répéter N fois
  - Sélectionner aléatoirement 7 points
  - Calculer la matrice F correspondante
  - Calculer les lignes épipolaires dans l'image 2 à partir des points de l'image 1 et de la matrice F
  - Compter et enregistrer le nombre de points dans l'image 2 qui sont à une distance de leur ligne épipolaire  $< P$  pixels (inliers)
  - (optionnel) recalculer F à partir de inliers
  - (optionnel) recalculer les inliers
  - mémoriser F si le nombre d'inliers est supérieur au meilleur nombre trouvé jusqu'ici

Réglez ensuite les différents seuils et paramètres pour avoir des résultats les plus proches possibles de la fonction utilisée dans la première question. Pour le nombre d'itérations, on pourra utiliser la table de la figure 3

Sample size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

FIGURE 3 – Nombres d’itérations de RANSAC nécessaires en fonction du nombre de paramètres du modèle et du taux d’outliers.