

ROB313 TP3 : Matrice fondamentale

Zheyi SHEN & Dajing GU

Janvier 2020

1 Introduction

Dans ce TP, on va travailler sur l'estimation de la matrice fondamentale qui encode la géométrie épipolaire reliant deux images prises par une caméra en mouvement. La bibliothèque OpenCV sous Python va être utilisé pour réaliser le traitement d'image.

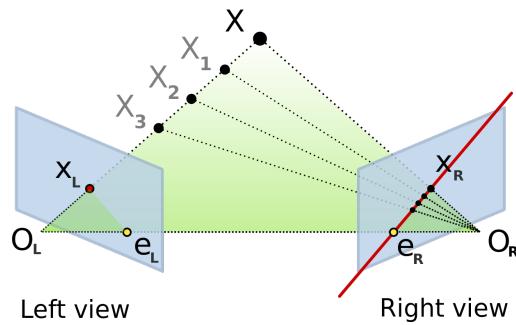


Figure 1.1: Illustration de la géométrie épipolaire

2 Paramètres importants

Dans cette partie, on expérimente la méthode d'estimation de la matrice fondamentale en utilisant la fonction `cv2.findFundamentalMat()`.

Le code fourni réalise cette estimation. Il d'abord trouve des points d'intérêt en utilisant la méthode KAZE (Figure 2.1) et fait une mise en correspondance fondée sur un critère de plus proche voisin. Ensuite, il applique un ratio test pour filtrer des appariements (Figure 2.2). Et puis, la fonction `cv2.findFundamentalMat()` est appliquée pour calculer la matrice fondamentale avec la méthode RANSAC (Figure 2.3).

Ces trois figures sont obtenu avec les paramètres défauts. Sur la figure 2.2, on peut voir que la plupart de appariements est responsable et selon la figure 2.3, les lignes épipolaires convergent vers un point au dehors de l'image.

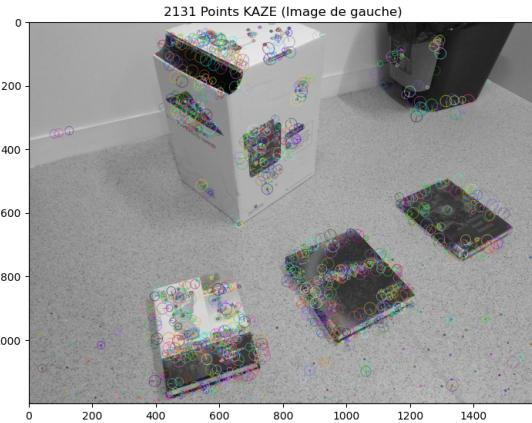


Figure 2.1: Les points d'intérêt pour l'image POP

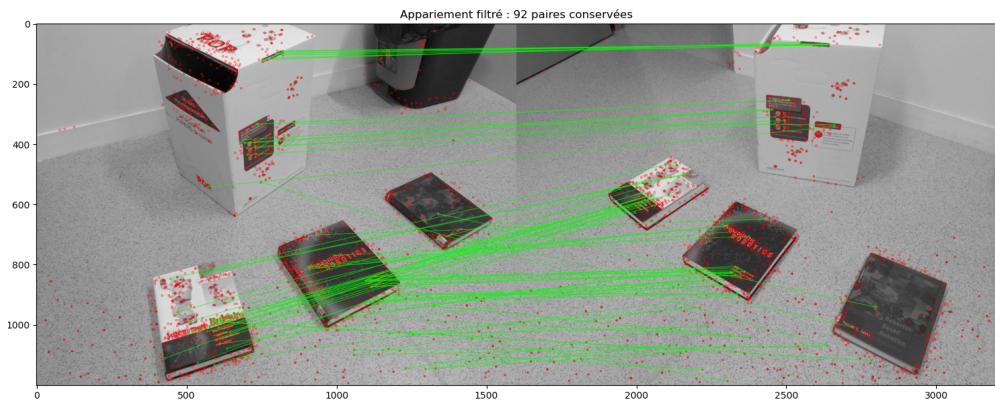


Figure 2.2: Les appariements pour l'image POP

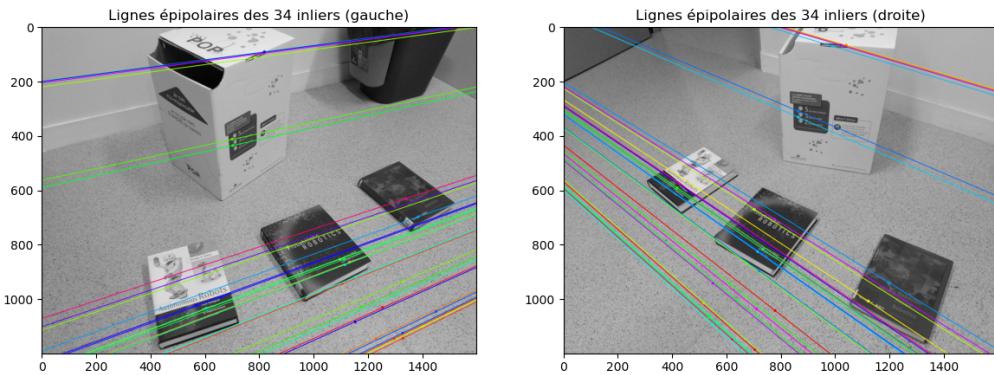


Figure 2.3: Les lignes épipolaires pour l'image POP

Dans le code fouri, les paramètres de détecteur KAZE, le ratio dans le "ratio test" et les paramètres dans la fonction `cv2.findFundamentalMat()` sont intervenants.

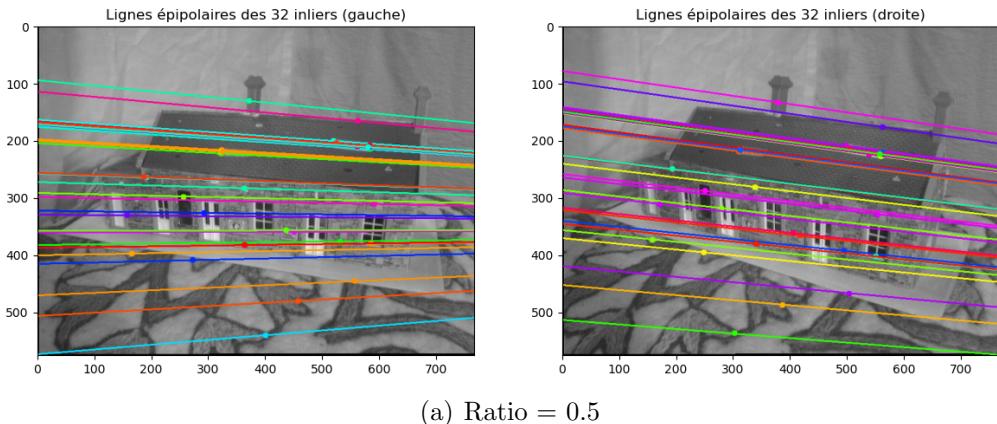
- Dans le détecteur KAZE

- **threshold** : Le seuil de réponse du détecteur pour accepter le point. Quand il est grand, le nombre de points d'intérêt acceptés diminue.

- **nOctaves** : L'évolution maximale en octave de l'image. Si on augmente cette valeur, le détecter est utilisé dans l'échelle plus grande.
- **nOctaveLayers** : Le nombre de sous-niveaux pour chaque niveau d'échelle. si il est élevé, on peut obtenir plus de points d'intérêt.
- **diffusivity** : le type de diffusivité, DIFF_PM_G1, DIFF_PM_G2, DIFF_WEICKERT ou DIFF_CHARBONNIER
- Dans le "ratio test"
 - **ratio** : Un paramètre entre 0 et 1 pour filtrer les appariements. Plus le ratio est grand, plus le nombre de paires de points sera utilisé pour calculer la matrice fondamentale, mais la probabilité d'erreurs d'appariement est élevée.
- Dans la fonction de calcul de matrice fondamentale avec RANSAC
 - **ransacReprojThreshold** : Paramètre utilisé uniquement pour RANSAC. Il s'agit de la distance maximale d'un point à une ligne épipolaire en pixels, au-delà de laquelle le point est considéré comme une valeur aberrante et n'est pas utilisé pour calculer la matrice fondamentale finale.
 - **confidence** : Il spécifie un niveau de confiance (probabilité) souhaitable que la matrice estimée est correcte.

Un des paramètres les plus déterminants dans la qualité du résultat final est le ratio. Si le ratio est grand, il y aura plus de mauvais appariements qui participent au calcul de la matrice fondamentale. Cela va impacter la précision du résultat et rendre la proportion de inlier plus basse. Si le ratio est petit, les appariements restés après le filtrage sont plus correspondants et cela peut augmenter la précision du résultat final. La figure 2.4 représente une comparaison de différentes valeurs de ratio pour l'image ToyHouse. Les autres paramètres restent défauts.

Dans le premier cas, il y a 32 inliers dans les 43 paires de points. Pour la deuxième, il y a 126 inliers parmi 570 appariements de points. La proportion d'inliers est plus grande quand le ratio est petit. Cependant, si il est trop petit, il est probable que le nombre de paires resté est inférieur à 8, ce qui cause l'impossibilité de calculer la matrice fondamentale. De plus, dans ces deux cas, le point convergé par les lignes épipolaires est différent, c'est-à-dire les matrices fondamentales obtenus ne sont pas mêmes.



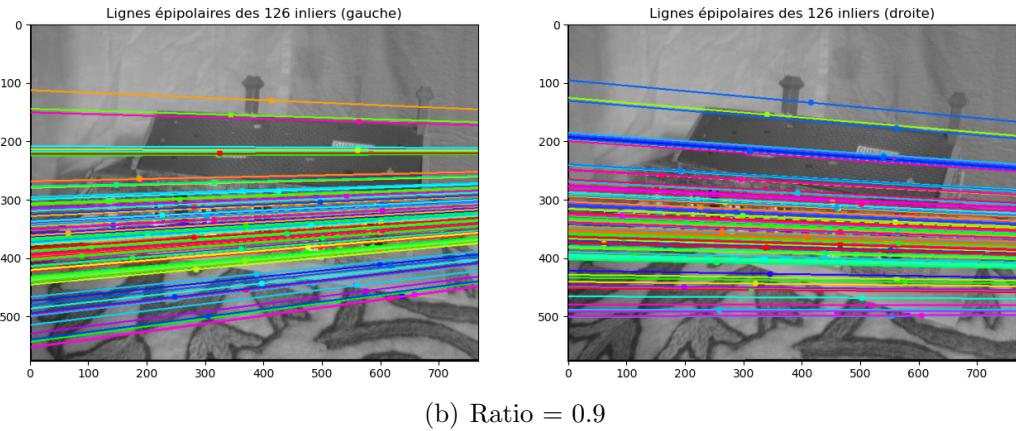


Figure 2.4: Comparaison de résultats pour différents ratios

3 Rectification des images

Rectification des images

Dans la question précédente, on a déjà tracé les lignes épipolaires correspondantes des points d'une image sur l'autre image. Si on étende chaque ligne épipolaire, on constate que chaque ligne épipolaire converge vers le pôle épipolaire donc les lignes obtenues ne sont pas en parallèles.

Les lignes épipolaires non parallèles augmentent la complexité de l'opération en particulier lors du calcul de la parallaxe. On discute maintenant de la façon de convertir des lignes épipolaires non parallèles en lignes parallèles, ce qui est nommée la rectification d'image.

Le phénomène des lignes épipolaires non parallèles provient des plans d'imagerie non coplanaires. En corrigeant le plan d'imagerie pour qu'il soit coplanaire, on peut rendre les paires de lignes épipolaires parallèles.

```

1  imsize = (imgL.shape[1], imgL.shape[0])
2  res, H1, H2 = cv2.stereoRectifyUncalibrated(pts1, pts2, ...
                                                 F, Ransac, imsize)
3  imgL = cv2.warpPerspective(imgL, H1, imsize)
4  imgR = cv2.warpPerspective(imgR, H2, imsize)

```

À l'aide de `cv2.stereoRectifyUncalibrated`, on a rectifié les images en calculant les deux homographies rendues par les deux faisceaux de droites épipolaires. Ensuite avec la fonction `cv2.warpPerspective`, on a appliqué les homographies correspondantes .

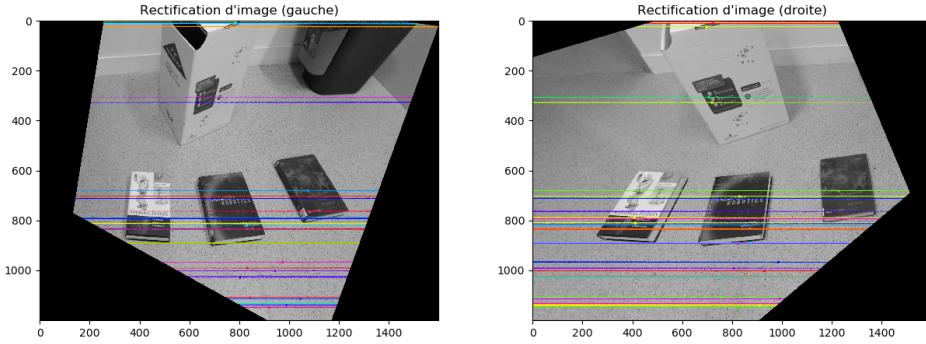


Figure 3.1: Rectification de l'image POP

Le résultat de la rectification de l'image POP est présenté dans la figure 3.1, on peut constater que les lignes epipolaires dans l'image à gauche et celles de l'image droite sont respectivement en parallèles, qui a prouvé l'effet de la rectification.

Difficultés potentielles

En essayant plusieurs paires d'image et on trouve que l'effet de la rectification dépend de la la nature du déplacement et des images.

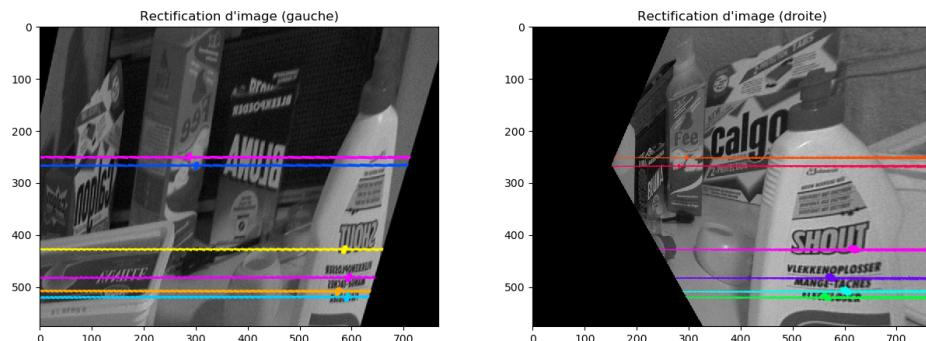


Figure 3.2: Rectification de l'image obj

Selon la figure 3.1 et 3.2, on constate que si le déplacement de la caméra est convergent (orientaion en fonction d'un point) ou si son déplacement est dans le plan focal, la rectification de l'image est bien faite, mais pour une caméra à déplacement radia, la qualité de la rectification n'est pas très satisfaisante, comme présenté dans la figure 3.3.

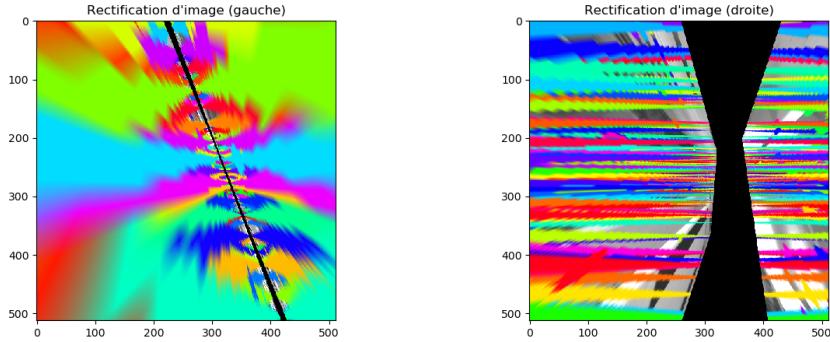


Figure 3.3: Rectification de l'image corridor

À notre avis, l'effet de la rectification des images dépend de la qualité des appariements des points d'intérêt. Donc une des difficultés potentielles se situe sur les appariements. Quand s'il y a trop de déformation sur les images, les appariements des points d'intérêt ne seront pas corrects. On ne peut pas non plus réussir à obtenir les matrices d'homographies correctes, qui nous empêche de finir la rectification des images. Cela peut expliquer la mauvaise qualit/ de la rectification dans la figure 3.3.