

TP Reconstruction 3D

Hermine CHATOUX

14 janvier 2025

L'objectif de ce TP est de vous présenter la reconstruction 3D à travers toutes ses étapes : de l'acquisition aux résultats, du calibrage à la carte de profondeur. Il y aura donc une partie acquisition, puis une partie traitement d'images.

Critère attendu :

Un compte rendu par binôme au format PDF contenant :

- certaines de vos images acquises,
- le code commenté,
- vos images résultats,
- et votre analyse des résultats obtenus.

Je ne souhaite pas un résultat parfait, mais votre compréhension de la méthode et les limites que vous avez pu rencontrer et identifier durant le TP.

Les critères de notation seront l'effort réalisé pour créer vos images, la compréhension du code fourni et l'analyse de vos résultats et de leurs limites.

Fonctionnement du fichier .py fourni :

L'ensemble des fonctions à appeler sont dans le *main*. Vous pouvez sélectionner les fonctions à lancer et sauvegarder si nécessaire les images/matrices à conserver pour la suite du TP.

1 Calibrage de la caméra

1.1 Acquisition

- Choisir votre paire de webcam. Vous pouvez effectuer des captures d'écran grâce à *Camera* sous Windows ou *Cheese* sous linux.
- Prendre une quinzaines de photos d'un damier fourni grâce au système d'acquisition choisi comme présenté sur la figure 1. Il est préférable de caler la mire ou la caméra pour éviter le flou de bouger durant l'acquisition.
- Enregistrer les images acquises.



(a) Image 1



(b) Image 2



(c) Image 3

FIGURE 1 – Exemples d'acquisition pour le calibrage de la caméra.

1.2 Estimation de la matrice de la caméra

- Ouvrir le fichier .py fourni. Nous nous intéressons à la fonction *CameraCalibration()*.
- Adapter et commenter le code pour calculer la matrice de calibrage de votre appareil photo.

1.3 Analyse des résultats

Une partie du code permet d'enlever la distorsion induite par l'appareil photo. La figure 2 présente des résultats. La première ligne présentent ceux obtenus avec les images openCV. Nous observons l'effet fish-eye qui grossit le zone centrale de l'image. La deuxième ligne présente les résultats avec mon téléphone. L'effet fish-eye est moindre mais la rectification entraîne une déformation que nous n'attendions pas.

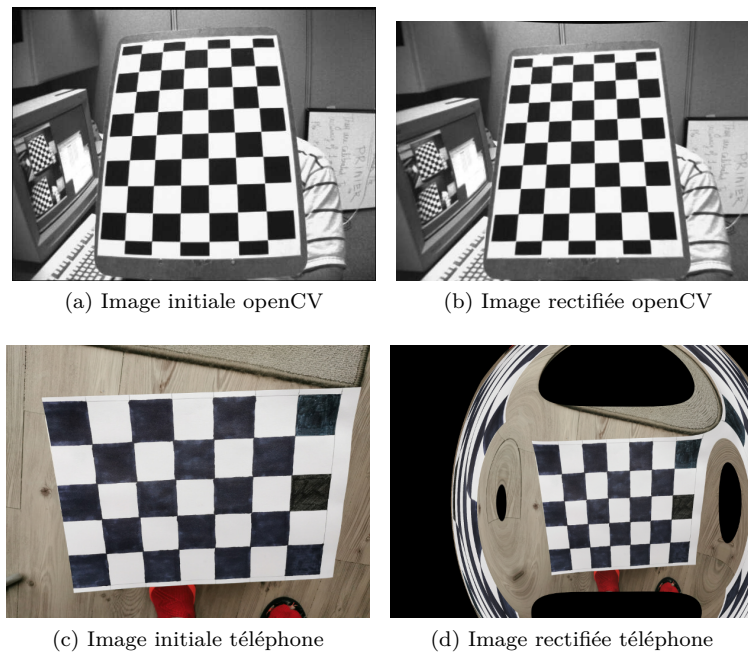


FIGURE 2 – Exemples de rectification de la distorsion de l'image.

- Quels phénomènes peuvent engendrer la déformation de la figure 2d ?
- Commenter les résultats obtenus avec vos images. Illustrer l'étape de rectification sur vos images résultats.
- Comparer les matrices de calibrage obtenues entre les deux webcams.

2 Calibrage de la stéréovision

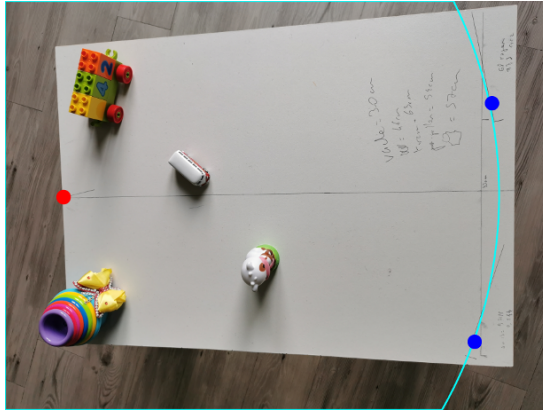
2.1 Acquisition

La figure 3 présente deux exemples de scènes acquises. Dans la configuration de la figure 3a, les deux centres des caméras (points bleus) sont situés sur un cercle de centre le point rouge. Le rayon est d'environ 68 cm et l'angle entre les deux prises de vue d'environ 30° . La figure 3b présente une configuration de caméras parallèles avec une simple translation de 4.5 cm.

- Monter une scène avec différents objets à différentes profondeurs.
- Sélectionner deux positions de caméras dont vous pouvez calculer la transformation spatiale entre les deux (rotation et/ou translation).

2.2 Estimation de la matrice fondamentale

Rappel Matrice de rotation autour d'un axe du repère :



(a) Rotation et translation



(b) Translation

FIGURE 3 – Exemples de mise en place pour l'acquisition

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} \quad R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Calculer/Mesurer la matrice de rotation R et le vecteur de translation t théorique d'après votre montage d'acquisition.
- Que fait la fonction `recoverPose` du fichier `.py` fourni ?
- Rappeler la relation entre la matrice de calibrage, la matrice essentielle et la matrice fondamentale. A quels éléments avons nous accès ?
- Adapter et commenter la fonction `StereoCalibrate(img1, img2, mapx1, mapy1, mapx2, mapy2)` du fichier `.py` fourni.

2.3 Analyse des résultats

- Commenter les résultats de matrice de rotation et translation obtenus avec la matrice essentielle et ceux que vous avez estimés. Correspondent-ils à ceux mesurés ?
- Tous les appariements semblent-ils cohérent ? Comment expliquer les anomalies ?

3 Géométrie épipolaire

3.1 Reconstruction des lignes épipolaires

- Adapter et commenter la fonction `EpipolarGeometry(pts1, pts2, F, maskF, img1, img2)` du fichier `.py` fourni.

3.2 Analyse des résultats

- Vérifier que les points d'intérêt de l'image i se retrouve sur les lignes affichées dans l'image j .
- Quel est l'intérêt de la rectification des images avant reconstruction ? Illustrer le phénomène sur vos images résultats.
- Quelles sont les étapes suivantes pour extraire la profondeur à votre avis ?

4 Reconstruction 3D

- Lancer le code `DepthMapfromStereoImages(imgL, imgR)` issu du fichier `.py` fourni.
- Commenter l'effet des paramètres que vous pouvez faire varier un par un.
- Adapter le à vos images. Attention il va falloir optimiser les paramètres. Bonne chance !