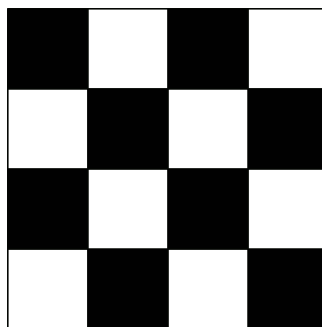


Année universitaire
2023 - 2024

Majeure IMI — Partie 3 — 5ETI
Acquisition Calibrage et
Reconstruction 3D

TP de calibrage



Eric Van Reeth
Mehdi Ayadi

Organisation du TP

Objectifs

L'objectif de ce TP consiste à **calibrer la webcam fournie** en début de séance (Logitech C270). Pour cela, une mire composée de 11×8 carrés de 20 mm de côté sera utilisée. Le calibrage sera réalisé en implémentant **la méthode de Tsai** introduite en cours.

Déroulement

Ce TP s'effectue en binôme par poste informatique (sous LINUX).

Le langage utilisé sera Python (attention à l'interpréteur), avec les librairies suivantes :

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
```

L'utilisation de l'IDE `VSCode` est fortement conseillée pour faciliter l'implémentation et le debug des codes. Il est recommandé d'implémenter votre TP sous le format Jupyter Notebook, en sauvegardant votre script au format `.ipynb`.

Évaluation

Un **compte-rendu au format pdf** devra être rendu par binôme dans un délai d'une semaine après la dernière séance de TP. Il contiendra les réponses aux questions posées, la description du protocole expérimental mis en place pour l'acquisition de la mire, une description claire et synthétique de la méthode implémentée, ainsi qu'une interprétation/discussion sur les paramètres estimés.

De plus, le **code python** sera demandé. Le cas échéant, le code devra être **accompagné des images capturées** sur lesquelles le calibrage est effectué, de sorte à ce que le correcteur puisse exécuter le code de façon autonome. Les paramètres estimés lors du calibrage devront être clairement affichés à la fin de l'exécution du script.

1 Acquisition de la mire

Dans un premier temps, il s'agit d'établir la correspondance entre les coordonnées 3D des points caractéristiques de la mire (coins des cases), et leur coordonnées pixel 2D de ces mêmes points sur les images acquises. Dans un premier temps, vous utiliserez les images de mires fournies avec le TP. Le protocole d'acquisition de ces images est décrit en Figure 1.

Par la suite, vous pourrez utiliser les mires fournies pour acquérir vos propres images (pour cela, repartez du code implémenté lors du TP d'acquisition pour réaliser la capture d'images à partir du flux vidéo).

1. Réalisez la **détection automatique des coins de la mire sur les 2 images** grâce à la fonction :

```
cv2.findChessboardCorners()
```

Si vous faites l'acquisition de vos propres images, vérifiez que la détection des coins de la mire a fonctionné avant de sauvegarder les images.

2. Créez le vecteur `coord_px` contenant toutes les **coordonnées pixels 2D** de l'ensemble des coins détectés sur les 2 images. Relevez avec attention l'ordre dans lequel sont ordonnés ces points.

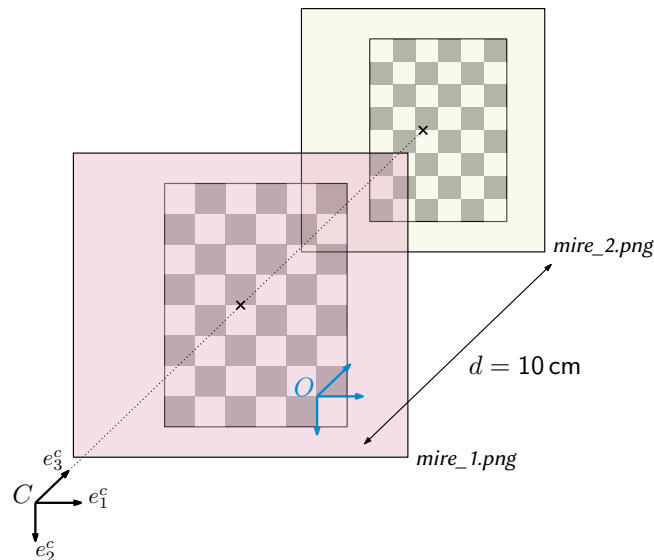


FIGURE 1 : Configuration de la prise de vue des images de mires fournies pour ce TP.

3. Créez le vecteur `coord_mm` contenant les **coordonnées spatiales 3D** de ces mêmes coins en veillant à ce que les coordonnées soient stockées dans le même ordre que les coordonnées pixel 2D.
Conseil : aidez-vous de la Figure 1 pour clarifier le positionnement respectif des repères caméra (d'origine C), et monde (d'origine O). Vous pouvez également choisir votre propre positionnement. Dans tous les cas, **votre compte-rendu devra contenir un schéma spécifiant l'orientation et le positionnement de ces deux repères.**

2 Calibrage de la caméra

2.1 Données du problème

- Vous considérerez que le centre optique se projette au centre de l'image, soit $(i_1, i_2) = (h/2, l/2)$ ou $(l/2, h/2)$
- Les images de mire fournies ont été acquises à 10 cm l'une de l'autre, en conservant le parallélisme de la mire et le centrage par rapport à la webcam (voir Figure 1).
- Les cases de l'échiquier mesurent $20 \text{ mm} \times 20 \text{ mm}$
- Trouvez la valeur de la focale de votre webcam sur le site support Logitech : <https://support.logi.com/hc/fr>, dans l'onglet « Caractéristiques Techniques »
- La résolution de systèmes linéaires de la forme $AX = B$, se fera en utilisant la pseudo-inverse de la matrice A qui s'obtient grâce à la fonction `np.linalg.pinv`

2.2 Implémentation de la méthode de Tsai

Les notations utilisées ici sont identiques à celles du cours disponible sur cpe-campus.

1. Construire la matrice A , de taille $(N \times 7)$, et le vecteur U_1 , de taille $(N \times 1)$, afin de résoudre le système $AL = U_1$.
2. En déduire la valeur des paramètres de la caméra contenus dans le vecteur L . Discutez le signe du paramètre o_2^c .
3. Comment déduire les paramètres (r_{31}, r_{32}, r_{33}) de la matrice de rotation, à partir des paramètres estimés précédemment ? En déduire les trois angles de rotation : (ϕ, γ, ω) .

4. Construire la matrice B , de taille $(N \times 2)$, et le vecteur R , de taille $(N \times 1)$, afin de résoudre le système :

$$B \begin{pmatrix} o_3^c \\ f_2 \end{pmatrix} = R$$

5. Déterminer la valeur des paramètres (s_1, s_2) , et en déduire la taille (en mm) du capteur de votre webcam.
6. Analyser les résultats obtenus et discuter de leur cohérence par rapport aux résultats attendus.
7. Afin de valider votre calibrage, projetez sur une des images de la mire, les coordonnées objets 3D utilisés pour le calibrage. Si votre calibrage est correct, ils doivent se projeter sur les coins des cases de l'échiquier.
8. Si vous ne l'avez pas déjà fait, calibrez maintenant la caméra à partir d'images acquises par vous-mêmes.

2.3 Question subsidiaire

En utilisant les mêmes vecteurs de points `coord_px` et `coord_mm` que pour votre calibrage, calibrer à nouveau votre webcam en utilisant la fonction dédiée d'opencv (`cv2.calibrateCamera`) qui estime également les déformations géométriques dues à l'optique. Comparez vos résultats avec ceux obtenus par openCV et discuter l'importance de la prise en compte de ces déformations. Vous pourrez par exemple quantifier l'erreur moyenne sur la position des points rétro-projetés.