

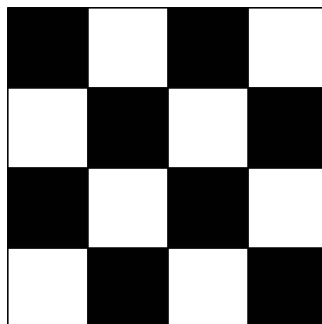
**Année universitaire  
2020 - 2021**

---

**Majeure IMI – Partie 3 – 5ETI**  
**Acquisition Calibrage et  
Reconstruction 3D**

**TP de calibrage**

---



**Eric Van Reeth**

# 1 Organisation du TP

## 1.1 Objectifs

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

## 1.2 Déroulement

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

Le langage utilisé sera Python, 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 également recommandé d'implémenter votre TP sous le format Jupyter Notebook, en sauvegardant votre script au format `.ipynb`. Veillez bien à utiliser `Python3`, et non pas `Python2.7`.

## 1.3 Évaluation

Un **compte-rendu au format pdf** devra être rendu 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. Il 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.

# 2 Acquisition de la mire

Dans un premier temps, il s'agit d'établir la correspondance entre les points de la mire et leur position sur l'image. Vous pourrez, au choix, utiliser les images de mires fournies avec le TP ou 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 des coins de la mire** 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** des coins détectés. Relevez avec attention l'ordre dans lequel sont ordonnés ces points.
3. Créez le vecteur `coord_mm` contenant les **coordonnées spatiales** de tous les coins détectés en respectant le même ordre que précédemment. Pour simplifier l'interprétation des paramètres estimés, faites en sorte que les orientations des référentiels objet et image soient proches.

### 3 Calibrage de la caméra

#### 3.1 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 12 cm l'une de l'autre
- 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 des systèmes linéaires de la forme  $AX = B$ , se fera en utilisant la pseudo-inverse qui s'obtient grâce à la fonction `np.linalg.pinv`

#### 3.2 Implémentation de la méthode de Tsai

Les notations utilisées ici sont identiques à celles des slides du cours disponibles 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 :  $(\phi, \gamma, \omega)$ .
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. Vérifier la cohérence de l'ensemble des paramètres estimés.
7. Afin de valider votre calibrage, projetez sur une des images de la mire, les points objets utilisés pour le calibrage. Si votre calibrage est correct, ils doivent se projeter sur les pixels contenant les points caractéristiques de la mire.
8. Si vous ne l'avez pas déjà fait, calibrez maintenant la caméra à partir d'images acquises par vous-mêmes.

#### 3.3 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.