



**Année universitaire**

**2019 - 2020**

---

**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 20mm 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, en utilisant les librairies `numpy` et `opencv` (`import cv2`).

L'utilisation d'IDE (PyCharm par exemple) est conseillée pour faciliter l'implémentation et le debug des codes. Veillez bien à utiliser Python 2.7 (et non pas Python 3).

Pour lancer la bonne version d'openCV (obligatoire) et PyCharm, entrer dans un terminal les commandes suivantes :

```
export PYTHONPATH=/sync/IMI/python_libs  
/sync/Robotic/pycharm/bin/pycharm.sh &
```

## 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 la description du protocole expérimental mis en place, 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 cette partie, il s'agit de mettre en place le protocole d'acquisition de la mire fournie de sorte à établir la correspondance entre les points de la mire et leur position sur l'image.

Pour cela, repartez du code implémenté lors du TP d'acquisition pour réaliser la capture d'images à partir du flux vidéo acquis.

1. Lors de l'appui sur une touche de votre choix, réalisez la **détection des coins de la mire** grâce à la fonction :  
`cv2.findChessboardCorners`.  
Sauvegardez l'image courante uniquement si la détection a fonctionné.
2. Répétez l'étape 1 autant que vous le jugez nécessaire.
3. Créez le vecteur `coord_px` contenant toutes les **coordonnées pixels** des coins détectés. Relevez avec attention l'ordre dans lequel sont ordonnés ces points.
4. Créez le vecteur `coord_mm` contenant les **coordonnées spatiales** de tous les coins détectés en respectant le même ordre que précédemment. Le référentiel objet (origine et direction des axes) est choisi arbitrairement.

### 3 Calibrage de la caméra

#### 3.1 Données du problème

- Vous considérerez que le centre optique se projette au centre de l'image, soit  $(i_1, i_2) = (h/2, l/2)$
- 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  $\mathbf{A}$  ( $N \times 7$ ) et le vecteur  $\mathbf{U}_1$  ( $N \times 1$ ), afin de résoudre le système  $\mathbf{A}\mathbf{L} = \mathbf{U}_1$ .
2. En déduire la valeur des paramètres de la caméra contenus dans le vecteur  $\mathbf{L}$ . 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  $\mathbf{B}$  ( $N \times 2$ ), et le vecteur  $\mathbf{R}$  ( $N \times 1$ ), afin de résoudre le système  $\mathbf{B} \begin{pmatrix} o_3^c \\ f^2 \end{pmatrix} = \mathbf{R}$ .
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. Tester différentes orientations de la mire et vérifier la cohérence des résultats obtenus. Discuter la reproductibilité des paramètres intrinsèques estimés lors des différents calibrages.

#### 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.