



**Année universitaire
2024 - 2025**

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

TP de recalage

Eric Van Reeth

Organisation du TP

Objectifs

Le but de ce TP est d'effectuer la stabilisation de vidéos en effectuant un recalage successifs des trames. Les recalages seront effectués à partir de la mise en correspondance de points d'intérêt détectés sur les trames. Pour cela, la librairie openCV sera utilisée, via l'extension opencv-contrib-python à installer avec pip :

```
python -m pip install opencv-contrib-python.
```

Plusieurs transformations seront utilisées pour effectuer le recalage et leurs performances seront comparées.

Déroulement

Ce TP s'effectue en binôme, sous VSCode idéalement.

L'utilisation du format Jupyter Notebook (`.ipynb`) est fortement conseillée pour faciliter l'implémentation et le debug des exercices de manière indépendante. Veillez bien à utiliser le même environnement virtuel Python que pour les autres TP du module. Les librairies à importer sont :

```
import cv2
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
```

Évaluation

Ce TP n'est pas évalué, mais les concepts vus en TP seront au programme du DS.

1 Détection et mis en correspondance de points d'intérêt

Dans un premier temps, téléchargez les données du TP ici.

Pour démarrer le TP, choisir une des deux vidéos. L'objectif de cette partie est d'effectuer un recalage entre les deux premières trames de cette vidéo. Une fois que celui-ci sera fonctionnel, le recalage sera appliqué à l'ensemble des trames pour effectuer la stabilisation de la vidéo complète.

1. Effectuer l'extraction de points d'intérêts (ORB par exemple) sur les deux premières trames de la vidéo choisie, et calculer les descripteurs associés :

```
orb.compute()
```

Affichez les points détectés et modifier les paramètres de détection si celle-ci ne vous paraît pas satisfaisante.

2. Grâce à la fonction `bf = cv2.BFMatcher()`, effectuer la mise en correspondance des points d'intérêts détectés grâce à la fonction `matches = bf.match()`

Affichez les points mis en correspondance avec la fonction : `cv2.drawMatches()`

2 Recalage de deux trames

Pour effectuer le recalage entre les deux premières trames, trois transformations seront comparées. Les fonctions Python permettant de calculer ces transformations à partir des points mis en correspondances sont les suivantes :

- Similarité : `cv2.findHomography()`

- Affine : `cv2.estimateAffine2D()`
 - Homographie : `cv2.estimateAffinePartial2D()`
1. Mettre les points détectés au format attendu par les fonctions de recalage (ou `kp_ref` sont les points détectés sur l'image de référence, et `kp_mov` ceux de l'image d'entrée) :

```
src_pts = np.float32([ kp_ref[m.queryIdx].pt for m in matches
↪ ]).reshape(-1,1,2)
dst_pts = np.float32([ kp_mov[m.trainIdx].pt for m in matches
↪ ]).reshape(-1,1,2)
```

2. Extraire et affichez la matrice (3×3) représentant la transformation estimée et analyser sa cohérence.
3. Réaliser l'opération pour les 3 transformations et comparer les matrices obtenues.

3 Stabilisation de la vidéo

Une fois le processus de recalage fonctionnel sur deux trames, effectuez la stabilisation de la vidéo complète :

1. Clarifier la méthode de stabilisation (généralisation de la méthode précédentes sur l'ensemble des trames)
2. Chaque trame recalée sera :
 - stockée dans une liste contenant toutes les trames recalées
 - exportée au format jpg dans un dossier différent des images d'entrée, avec la fonction `plt.imshow()`
3. Une fois l'ensemble des trames recalées, la vidéo de sortie sera générée avec le script suivant (dans lequel `oframes` est la liste qui contient l'ensemble des trames recalées, et `w` et `h` les dimensions des images) :

```
# Create mp4 from jpg list
path_out = "OutputPath/"
fourcc = cv2.VideoWriter_fourcc(*"mp4v")
vidwriter = cv2.VideoWriter(path_out + 'output.mp4', fourcc, 24,
↪ (w,h))
for frame in oframes:
    vidwriter.write(cv2.cvtColor(frame, cv2.COLOR_RGB2BGR))
vidwriter.release()
```

4. Effectuer le recalage complet pour les 3 transformations et analyser quantitativement et qualitativement les résultats obtenus.