

## Rapport TP Calibrage

### Question 1 on Panorama.py :

- La transformation géométrique qui est pré-implémentée est une homographie (une combinaison de rotations et de translations) qui permet facilement de superposer deux images qui ont subi des rotations sur le centre optique lors de leur prise de vue. Elle est appropriée car elle conserve les lignes droites.
- En modifiant le seuil on modifie le nombre d'éléments qui marchent entre les deux images.

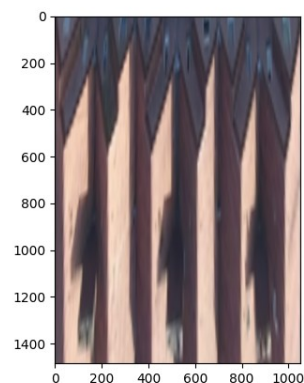
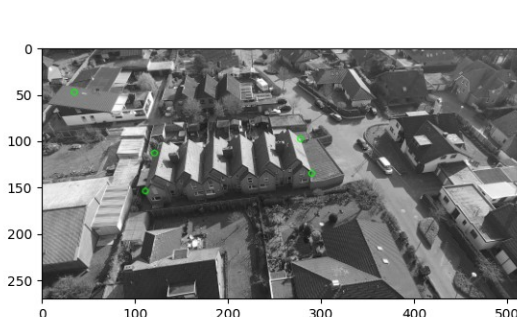
### Question 2 on Corners.py :

Le fichier python fonctionne de la manière suivante :

- On commence par ouvrir l'image avec opencv et la convertir en nuances de gris et on lui applique un rapide pré-processing (gaussian Blur pour enlever le bruit)
- On applique ensuite Canny à l'image qui permet d'obtenir les contours de l'image en niveaux de gris. L'algorithme est fait à base de gradients de l'image et de filtres comme le filtre de Sobel. On utilise ensuite cv2.HoughLines pour déterminer les lignes de l'image (on a les lignes dans une liste).
- On regarde ensuite les intersections entre ces lignes et on les affiche, en superposition sur l'image.

### Question 3 on redressA4.py :

- Appliquer un bruit gaussien à l'image permet de la flouter, enlevant ainsi le bruit s'il y en a.
- Nous aurions pu utiliser d'autres filtres comme le filtre médian qui remplace juste un pixel par la médiane des voisins (idéal si un pixel est blanc dans une zone uniforme noire par exemple) ou un filtre moyenneur qui fait la moyenne des voisins.
- Sur des bâtiments, on obtient le résultat suivant :



- avec une valeur aberrante, les résultats ne sont plus satisfaisants, le modèle n'est pas robuste aux outliers.