POLYTECH°
ANGERS

Matrice extrinsèque et matrice de projection.
(2 séances)

L'objectif de ces exercices de se familiariser avec les transformations entre le repère 3D "monde" et le repère 3D "caméra" (matrice extrinsèques), combinées à la matrice intrinsèque.

Exercice 1
On considère que l'état de référence correspond à l'image de référence, avec la matrice extrinsèque suivante :

$$M_{ext} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

Cela revient à considérer que le repère 3D caméra et "monde" sont confondus (aucune translation, ni rotation, ni changement d'échelle). Les coordonnées, dans le repère monde, des coins du damier et du carré sont donc les coordonnées (homogènes !) dans le repère caméra.

1. A partir des corners obtenus via une méthode du TP précédent, calculer dans le repère caméra les coordonnées des 4 sommets `s1,s2,s3,s4` du carré rouge présents dans la figure 1.

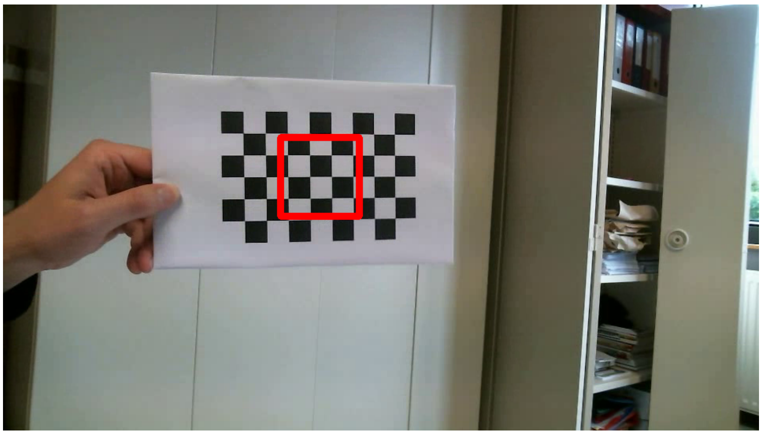

Figure 1 – Image attendue à la fin de cet exercice.

2. Calculer les coordonnées des projetés des sommets dans le repère image.
3. Ajouter le carré sur l'image avec une fonction `opencv`.

Exercice 2
On souhaite appliquer une translation (dans le monde 3D) au carré pour qu'il soit décalé de 100 mm dans le repère monde afin d'apparaître à droite dans l'image pixels 2D :

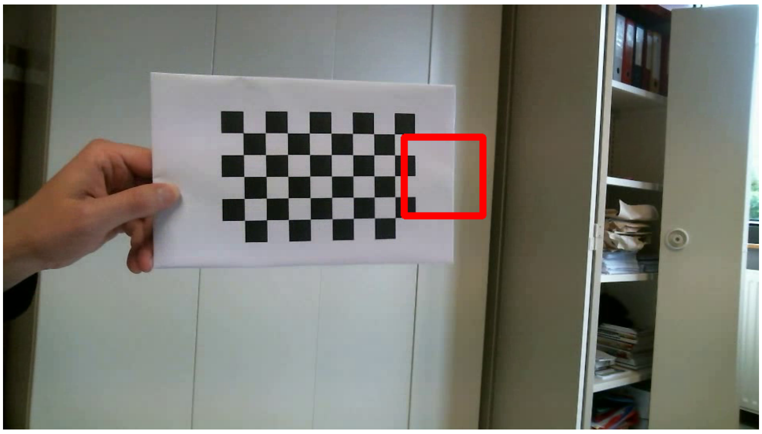

FIGURE 2 – Image attendue à la fin de cet exercice.

1. Définir la matrice extrinsèque appropriée,
2. Intégrer cette opération dans le script Python précédent. On prendra garde de disposer des coordonnées *homogènes* dans les repères monde ou caméra de sorte qu'une translation soit linéaire. Le script suivant permet d'ajouter une *colonne de 1* aux coordonnées 3D de l'objet virtuel. On pourra ajouter au helper la fonction `coord2homogeneous(coords)` qui réalise cette opération :

```
new_col=np.ones((coords.shape[0],1))
coords_h=np.hstack((coords,new_col))
```

Exercice 3 (Recalage manuel)
Le but de cet exercice est d'estimer intuitivement la matrice M_{ext} permettant de replacer le carré initial sur le damier pour différentes poses. Cette estimation "intuitive" revient à construire manuellement la matrice $[R|T]$ en fonction de la pose observée du damier par rapport à l'image de référence (3) et de la connaissance de la géométrie du système.

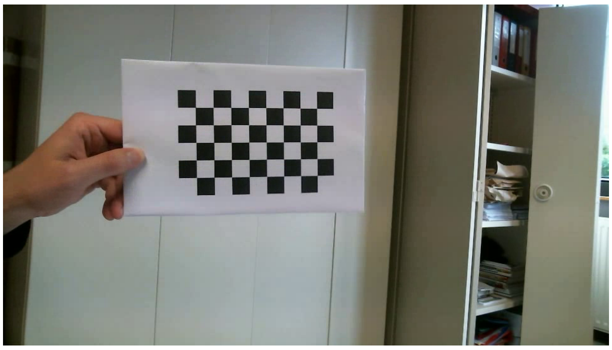

Figure 3 – Image de référence.

1. Pour la pose 1, l'objet semble éloigné de la caméra Translation selon X,Y,Z de -20, -25, 300.

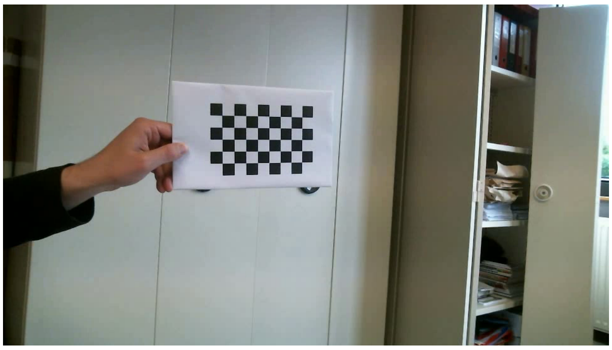

Figure 4 – Pose 1 : l'objet semble éloigné de la caméra Translation selon X,Y,Z de -20, -25, 300.

2. Pour la pose 2, le damier semble avoir été déplacé sans être éloigné.

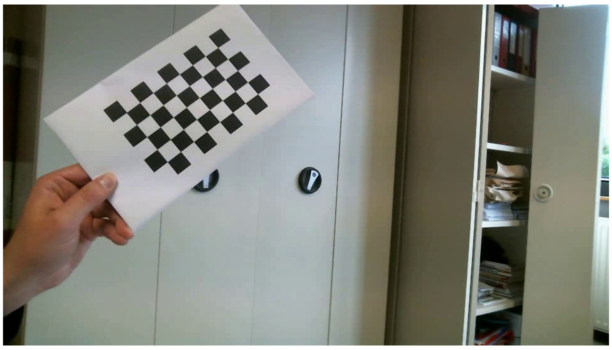

FIGURE 5 – Pose 2 : le damier semble avoir été déplacé sans être éloigné. Il a également subi une rotation. On pourra tester avec une rotation autour de l'axe (Oz) de -35 degrés, et un translation selon X, Y de -60, -70.

3. Pour cette pose 3, on prendra garde que l'objet initial n'est pas placé au centre du repère "monde". La conséquence est qu'une rotation (par rapport au centre du repère) appliqué à l'objet entraîne également une translation qu'il faudra compenser par la translation "inverse" appropriée.

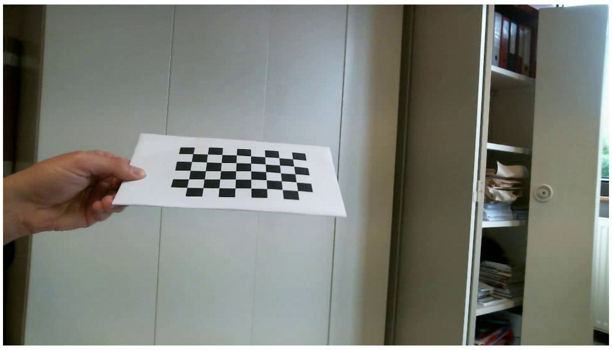

FIGURE 6 – Pose 3 : le damier ne semble pas avoir été déplacé. Par contre, il a subi une rotation. On pourra tester avec une rotation autour de l'axe (Ox) de -55 degrés, et un translation selon X, Y, Z de -30,-390,300.