

## TP 4 - Robot Dobot Magician : Vision artificielle (webcam)

Jean-Louis Boimond  
Université Angers

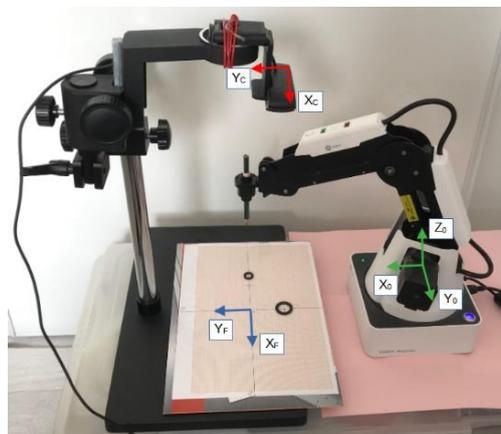
**Objectif :** Des cercles, de couleur noire, sont positionnés de manière quelconque sur une feuille de papier blanc de format A4 (210 × 297 mm) sachant qu'un tel format s'avère relativement bien adapté à l'espace d'atteignabilité du robot Dobot. Une caméra 'Logitech Webcam C930e' (possédant 1920 × 1080 pixels), sous laquelle la feuille est placée, permet la détection des cercles. Il s'agit de programmer le robot, muni de l'outil 'Feutre', afin de diriger la pointe du feutre aux centres de chacun des cercles, voir le montage décrit ci-dessous.



Pour réaliser cela, il va s'agir :

- dans une **1<sup>ère</sup> étape**, de disposer dans le repère  $(O_C, X_C, Y_C)$  associé à la caméra des coordonnées (en pixel) des points correspondant aux centres de chacun des cercles situés sur la feuille A4 et détectés par la caméra ;
- dans une **2<sup>ème</sup> étape**, d'en déduire l'expression dans le repère  $(O_F, X_F, Y_F)$  associé à la feuille A4 des coordonnées (en mm) des points ;
- dans une **3<sup>ème</sup> étape**, d'en déduire l'expression des coordonnées des points dans le repère  $(O_0, X_0, Y_0, Z_0)$  associé à la base du robot ;
- dans une **4<sup>ème</sup> étape**, de programmer le robot afin de permettre le déplacement de la pointe du feutre vers les centres de chacun des cercles.

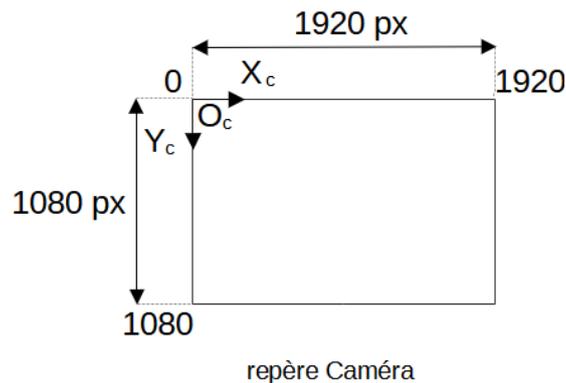
Ces différents repères sont décrits dans la figure qui suit. Le repère associé à la feuille A4 permet de faciliter le calcul des coordonnées dans le repère associé à la base du robot des points exprimés dans le repère associé à la caméra. Vous noterez que les axes  $Z$  des repères associés à la caméra et à la feuille ne sont pas pris en compte car les points exprimés dans ces repères ont tous une composante nulle selon l'axe  $Z$ .



- ✓ Charger dans votre répertoire de travail l'ensemble des fichiers contenus dans le fichier zip accessible *via* le lien [Tools TP4.zip](#). Notamment, des scripts Python sont fournis pour vous aider à réaliser les quatre étapes décrites précédemment.
- ✓ Installer le logiciel de la caméra (LogiCameraSettings\_2.12.8.exe) afin que les scripts Python disposent des drivers de la caméra, ainsi que la version 1.9.4. du logiciel DobotStudio (en utilisant le lien [DobotStudioSetup V 1 9 4.zip](#), voir la page 5 du TP1 concernant l'installation) pour disposer des drivers permettant de communiquer avec le robot.

**Etape 1 : obtention des coordonnées (en pixel) des points correspondant aux centres de chacun des cercles dans le repère associé à la caméra**

Le repère Caméra ( $O_c, X_c, Y_c$ ) est défini comme suit.



- ✓ Après avoir placé une feuille A4 sous la caméra (voir photos ci-dessus), régler la hauteur de la caméra par rapport à la feuille à une valeur approximative de 35 cm (cette valeur sera ajustée lors de la phase de calibration de la caméra réalisée durant l'étape 2). Disposer de manière quelconque les cercles noirs sur la feuille.

La détection de ces cercles par la caméra résulte du traitement suivant :



La transformée de Hough, utilisée lors de la binarisation par seuillage, est décrite à l'adresse :

[https://fr.wikipedia.org/wiki/Transformée\\_de\\_Hough](https://fr.wikipedia.org/wiki/Transformée_de_Hough)

- ✓ Régler le contraste de la caméra pour avoir une bonne visibilité en utilisant son logiciel dédié (accessible *via* l'icône 'Paramètres de la caméra Logitech' apparue dans le bureau suite à l'installation du logiciel).

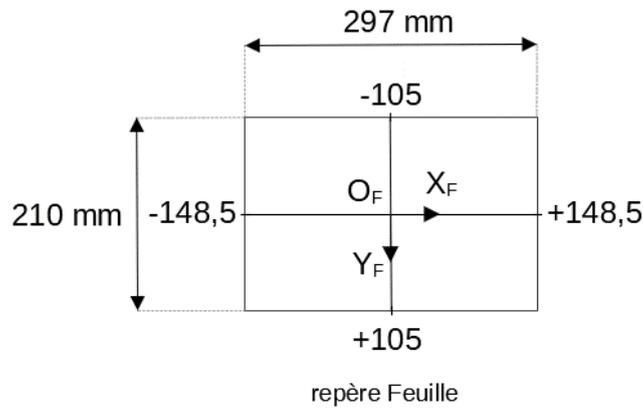
**NB :** Pensez à quitter ce logiciel une fois le réglage terminé afin de pouvoir exécuter les scripts Python impliquant la caméra.

- ✓ Identifier dans le script 'Dobot\_repere\_camera.py' les différents traitements d'image décrits ci-dessus.

- ✓ Régler à l'aide du script 'Dobot\_histogramme.py' la valeur du seuil (initialement égale à 100) en vous aidant d'un histogramme de l'image, l'objectif étant de détecter les centres de chacun des cercles (situés sur la feuille A4).

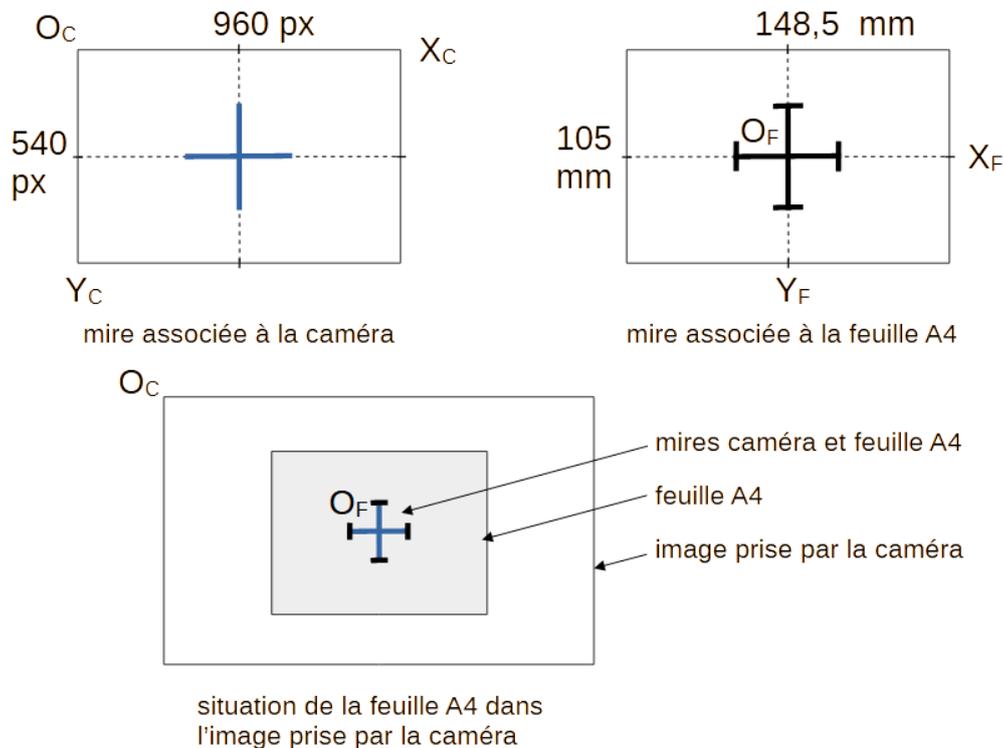
**Etape 2 : expression des coordonnées (en mm) des points correspondant aux centres de chacun des cercles dans le repère associé à la feuille A4**

Le repère Feuille ( $O_F, X_F, Y_F$ ), dont l'origine est située au centre de la feuille, est défini comme suit.



**NB :** Veiller à disposer la feuille afin de positionner le repère Feuille par rapport au repère Caméra comme indiqué dans la 2<sup>ème</sup> figure de la page 1 du TP.

- ✓ A l'aide du script 'Dobot\_calibration.py', régler la hauteur de la caméra et déplacer la feuille A4 (placée sous la caméra) afin de superposer la mire associée à la caméra à la mire associée à la feuille, voir ci-dessous une description de ces deux mires, ainsi que la situation de la feuille A4 dans l'image prise par la caméra.



**Hypothèse :** Dans la suite du TP, les 2 mires sont supposées être superposées (penser à fixer la feuille pour ne pas avoir à revenir sur le réglage).

- 1) Dédurre du script la longueur et la largeur (en pixel) de la mire associée à la caméra. Mesurer la longueur et la largeur (en mm) de la mire associée à la feuille.
- 2) A quoi correspond un pixel en mm<sup>2</sup> ? Est-ce que la caméra est située dans l'espace d'atteignabilité du robot ?
- 3) Etablir la relation permettant de déduire les coordonnées  $x_F, y_F$  (en mm) d'un point  $P$  situé sur la feuille à partir de ses coordonnées  $x_C, y_C$  (en pixel).
- 4) Modifier la ligne :

```
center_mm[j, :] = [i[0], i[1]]
```

du script 'Dobot\_repere\_feuille.py' afin de disposer dans la variable `center_mm` des coordonnées des points correspondant aux centres de chacun des cercles détectés par la caméra. Vous noterez que, par rapport au script 'Dobot\_repere\_camera.py', ce script réalise un masquage au format A4 de l'image avant son passage en niveaux de gris, ceci pour garantir qu'aucun cercle situé en dehors de la feuille ne soit détecté. Trouver dans le script le traitement relatif à ce masquage.

### Etape 3 : expression des coordonnées des points correspondant aux centres de chacun des cercles dans le repère associé à la base du robot ( $R_0$ )

- 5) Calculer dans le repère de base du robot les coordonnées  $(x_0, y_0, z_0)$  d'un point  $P$  situé sur la feuille sachant que les coordonnées  $(x_F, y_F, z_F)$  de ce point sont connues dans le repère Feuille. Vous disposez pour cela du robot, du logiciel DobotStudio et éventuellement d'un rapporteur.

Le robot étant muni de l'outil 'Feutre', sélectionnez dans DobotStudio l'outil « **Advanced** » avec les valeurs préalablement configurées comme suit :

```
EndType : PF
xBias = 61; yBias = 0; zBias = 82,7
```

ceci afin que les valeurs  $X, Y, Z$  affichées dans l'encadré « Operation Panel » (situé à droite) désignent les coordonnées de la pointe du Feutre dans le repère associé à la base du robot ( $R_0$ ).

D'autre part, pensez à calibrer au préalable le robot (à l'aide de la touche HOME située en haut à droite) afin que vos mesures soient correctes.

### Etape 4 : programmation du robot pour permettre le déplacement de la pointe du feutre vers les centres de chacun des cercles

'Dobot\_commande.py' est un script issu du fabricant (décrivant les instructions de commande standard utilisées pour réaliser une trajectoire) dans lequel quelques modifications ont été apportées.

#### NB :

- a) Indiquer le numéro du port de communication dans l'instruction :

```
state = dType.ConnectDobot(api, "", 115200) [0]
```

Par exemple, remplacer "" par "COM3" pour indiquer le port COM3. Reportez-vous au numéro de port utilisé dans DobotStudio lors de la connexion pour trouver le bon numéro de port.

- b) L'instruction :

```
dType.SetEndEffectorParams(api, 61, 0, 82.7, isQueued = 1)
```

définit la pointe du Feutre comme étant le centre de l'outil (Tool Center Point) du robot.

- 6) Expliquer les différentes fonctions réalisées dans le script.

Considérer la boucle 'for i in center\_mm', quel est le contenu de la variable center\_mm ?  
Expliquer ce que réalise l'instruction qui suit, située dans cette même boucle :

```
center_cm_R_0=numpy.add(O_0_O_F, numpy.dot(R,numpy.array([[i[0]], [i[1]], [0]]))).
```

Que représente le vecteur multiplié (à gauche) par R ? Pourquoi sa 3<sup>ème</sup> composante est égale à 0 ?

- 7) Compléter le script 'Dobot\_commande.py' en insérant dans une première partie le script 'Dobot\_repere\_feuille.py' ; puis modifier le script de commande Dobot (à savoir la section repérée par le commentaire # Partie à compléter) afin de programmer la mise en mouvement en ligne droite (mode Linéaire) de la pointe du Feutre vers les centres des cercles détectés par la caméra sachant que le bras du robot devra rejoindre le point de coordonnées :  $x = 250 \text{ mm}$ ,  $y = 0$ ,  $z = 50 \text{ mm}$  à l'issue de chaque mouvement du Feutre vers le centre d'un cercle.