

Note d'application
Pepper range ses affaires
Calibration de la caméra de Pepper

Table des matières

Introduction.....	3
Prérequis	4
Calibration de la caméra frontale de Pepper	4
Mettre à jour les paramètres de la caméra.....	6

Introduction

Pour réaliser le projet « Pepper range ses affaires » proposé par Polytech Clermont-Ferrand, il a fallu programmer le robot Pepper pour qu'il puisse trouver et ranger des objets disposés dans la zone de Pepper en autonomie. Pour commander le robot Pepper, on pouvait utiliser trois outils de développement. Au final, on a choisi ROS (Robot Operating System) comme l'outil de développement. Pour repérer l'objet, on a utilisé différents Qr codes comme les différents objets, et l'outil ArUco (un paquet sous ROS) qui permet au robot de détecter les marqueurs et d'avoir les informations sur les marqueurs détecté (identifiant, position...). Afin d'obtenir la position de l'objet de façon plus précise, on a réalisé la calibration de la caméra frontale de Pepper sous l'environnement ROS.

Prérequis

Tout d'abord, il faut installer le paquet *camera_calibration* sur le PC sous l'environnement ROS, en faisant la ligne de commande suivante sur le terminal, ce paquet va permettre de calibrer une caméra.

```
apt-get install ros-indigo-camera-calibration
```

Calibration de la caméra frontale de Pepper

Après l'installation du paquet *camera_calibration*, exécuter le fichier de lancement de *Naoqi Driver* en faisant les lignes de commandes suivantes, avec l'adresse IP du robot comme 192.168.1.101.

```
cd Bureau/catkin_ws  
catkin_make  
source devel/setup.bash  
roslaunch naoqi_driver naoqi_driver.launch nao_ip:= 192.168.1.101
```

Ensuite, il faut ouvrir un autre terminal, faire les lignes de commande suivantes pour lancer le Node *cameracalibrator.py* (un fichier du paquet *camera_calibration*), en fournissant les caractéristiques du modèle d'étalonnage (le nombre de lignes moins 1, le nombre de colonnes moins 1, la taille d'un carré), ainsi que le topic avec des images de la caméra frontale de Pepper. Dans la dernière ligne de commande, un damier 8x6 avec la taille d'un carré de 35mm est utilisé.

```
source devel/setup.bash  
roslaunch camera_calibration cameracalibrator.py --size 8x6 --square 0.035 image:=/naoqi_driver_node/camera/front/image_raw
```

Une fenêtre de calibration apparaît alors, dans laquelle le damier va être souligné. Il faut prendre le damier dans le sens où il y a le plus de carreaux sur l'horizontal que sur la vertical comme sur la figure 1.

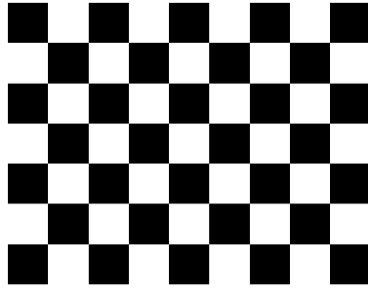


Figure 1 : Damier 8x6

Mettre le damier devant la caméra frontale de Pepper, tourner et bouger le damier dans des différents angles, directions et échelles, jusqu'au moment où le bouton "Calibrate" devient accessible, puis appuyer le bouton "Calibrate" et attendre les résultats. Voir figure 2.

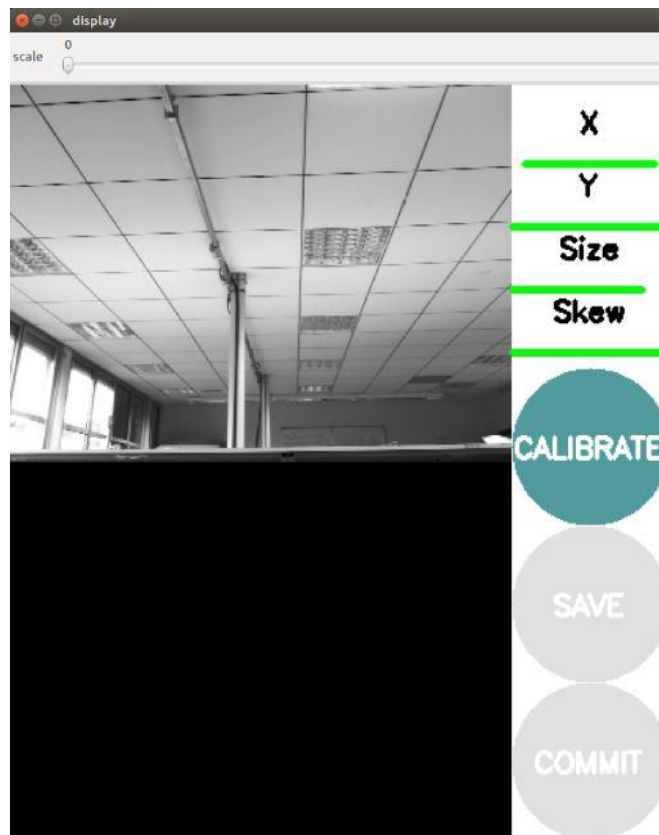


Figure 2 : Fenêtre de la calibration de la caméra du front de Pepper

Les différentes calibrations dans la figure 2 :

- Calibration X : le damier est détecté sur les bords gauche et droit du champ de vision.
- Calibration Y : le damier est détecté sur les bords haut et bas du champ de vision.
- Calibration Skew : le damier est détecté à divers angles de la caméra.
- Calibration Size : le damier remplit tout le champ de vision.
- Scale : la valeur recommandée pour ce curseur est 0.

Le temps de calcul peut être court ou très long en fonction de la taille des données qu'on obtient pour les différentes calibrations au-dessus. Le temps de calcul pour le cas de la figure 2 était 1h30.

Mettre à jour les paramètres de la caméra

Mettre à jour les nouvelles valeurs pour les différentes matrices dans le fichier *camera_info_definitions.hpp*, ce qui se situe dans le paquet *Naoqi Driver* (*naoqi_driver/src/converters/camera_info_definitions.hpp*). Ce fichier contient des informations pour toutes les caméras des robots Nao, Pepper et Romeo, ces trois robots ont une résolution de caméra différente. Pour Pepper, la résolution de la caméra est 320x240. La figure 3 montre la partie de la caméra frontale de Pepper dans ce fichier.

```
inline sensor_msgs::CameraInfo createCameraInfoTOPQVGA()
{
    sensor_msgs::CameraInfo cam_info_msg;

    cam_info_msg.header.frame_id = "CameraTop_optical_frame";

    cam_info_msg.width = 320;
    cam_info_msg.height = 240;
    cam_info_msg.K = boost::array<double, 9>{{ 274.139508945831, 0, 141.184472810944, 0, 275.741846757374, 106.693773654172, 0, 0, 1 }};

    cam_info_msg.distortion_model = "plumb_bob";
    cam_info_msg.D = boost::assign::list_of(-0.0870160932911717)(0.128210165050533)(0.003379500659424)(-0.00106205540818586)(0).convert_to_container<boost::array<double, 5>>();

    cam_info_msg.R = boost::array<double, 9>{{ 1, 0, 0, 0, 1, 0, 0, 0, 1 }};

    cam_info_msg.P = boost::array<double, 12>{{ 272.423675537109, 0, 141.131930791285, 0, 0, 273.515747070312, 107.391746054313, 0, 0, 0, 1, 0, 0 }};

    return cam_info_msg;
}
```

Figure 3 : Informations de la caméra du front du robot Pepper avec une résolution de 320x240

https://github.com/ros-naoqi/naoqi_driver/blob/master/src/converters/camera_info_definitions.hpp.

Les résultats de la calibration donnent la résolution de la caméra, on pourra la vérifier également en utilisant la ligne de commande suivante, qui permet d'afficher les informations de la caméra frontale de Pepper.

```
rostopic echo /naoqi_driver/camera/front/camera_info
```

Après la mise à jour du fichier, arrêter tous les programmes, compiler le code en faisant la ligne de commande suivante dans un terminal.

```
catkin_make
```