

E: Camera Calibration

本次實驗目的為確認相機正常運作並且運行校正程式讓 Lane following 能夠正常運作。

軟體與硬體準備

- 確認已完成 duckietown 環境安裝 ([02-04-duckietown-setup](#)):
 - 確認 duckiebot 已下載安裝 duckietown
 - 確認 duckiebot 已下載安裝 duckiefleet
- 先認識 [byobu](#) 基本操作指令.

E: Camera Calibration	1
軟體與硬體準備	1
總覽	1
實驗任務	2
Task 1 確認相機正常運作	2
Check Point:	3
Task 2 相機校正 - intrinsic calibration	4
Task 2-1 啟用 ForwardX11	4
Task 2-2 設定 ROS 並開始校正	4
Check Point:	6
Task 3 相機校正 - extrinsic calibration	6
Check point:	7
Reference	7

總覽

預計完成時間: 40 分鐘

完成次實驗後應具備以下能力

- 能夠使用 `rviz` 看到相機畫面
- 會簡單使用 `roslaunch`, `rosservice` 指令與操作 `byobu`.
- 對 ROS 能有基本認識

實驗任務

Task 1 確認相機正常運作

duckiebot 上輸入指令

```
duckiebot $ cd ~/duckietown
```

```
duckiebot $ source environment.sh
```

```
duckiebot $ source set_ros_master.sh robotname
```

開啟相機

```
duckiebot $ roslaunch duckietown camera.launch veh:=robotname raw:=true
```

laptop 上設定 ROS 環境

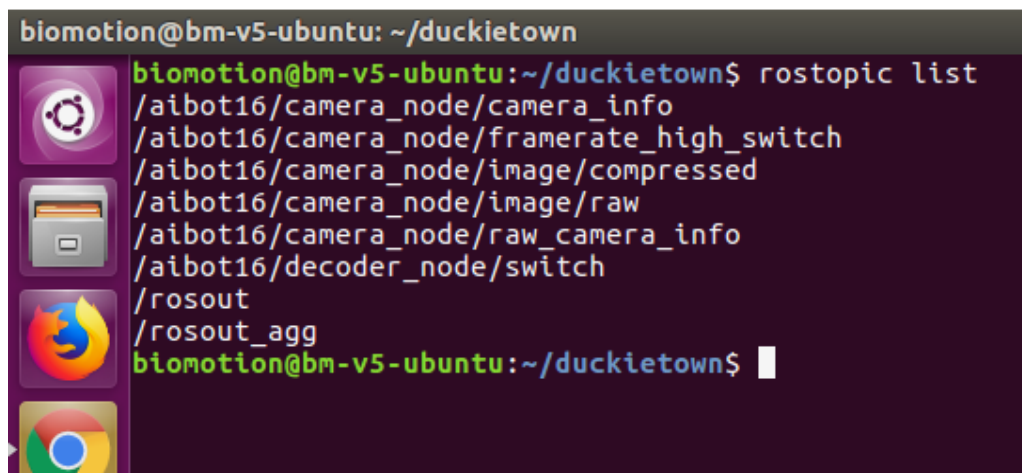
```
laptop $ cd ~/duckietown
```

```
laptop $ source environment.sh
```

```
laptop $ source set_ros_master.sh robotname
```

確認 camera.launch 上的 topic

```
laptop $ rostopic list
```

A terminal window screenshot from a laptop. The terminal title is 'biomotion@bm-v5-ubuntu: ~/duckietown'. The command 'rostopic list' has been executed, and the output lists several ROS topics: '/aibot16/camera_node/camera_info', '/aibot16/camera_node/framerate_high_switch', '/aibot16/camera_node/image/compressed', '/aibot16/camera_node/image/raw', '/aibot16/camera_node/raw_camera_info', '/aibot16/decoder_node/switch', '/rosout', and '/rosout_agg'. The prompt is now 'biomotion@bm-v5-ubuntu:~/duckietown\$'.

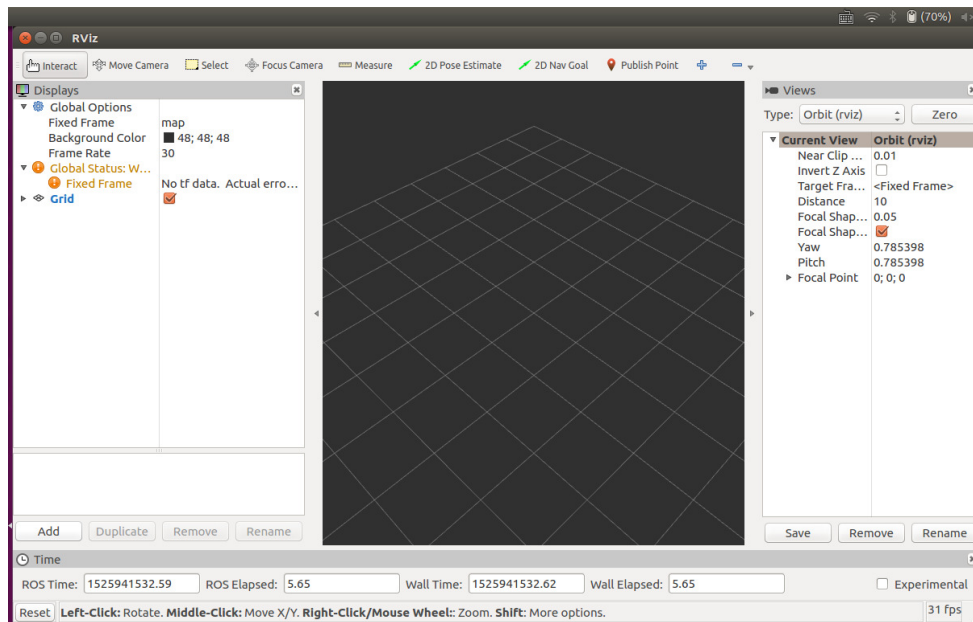
```
biomotion@bm-v5-ubuntu: ~/duckietown
biomotion@bm-v5-ubuntu:~/duckietown$ rostopic list
/aibot16/camera_node/camera_info
/aibot16/camera_node/framerate_high_switch
/aibot16/camera_node/image/compressed
/aibot16/camera_node/image/raw
/aibot16/camera_node/raw_camera_info
/aibot16/decoder_node/switch
/rosout
/rosout_agg
biomotion@bm-v5-ubuntu:~/duckietown$
```

應該要找到一行寫著 **robotname**/camera_node/image/raw

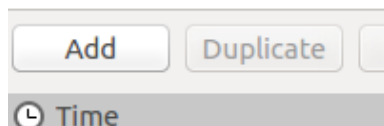
繼續在筆電上輸入指令確認相機畫面

```
laptop $ rviz
```

這將會開啟 RViz 並且顯示以下畫面



在 RViz 中，視窗左下角可以看到下圖按鈕

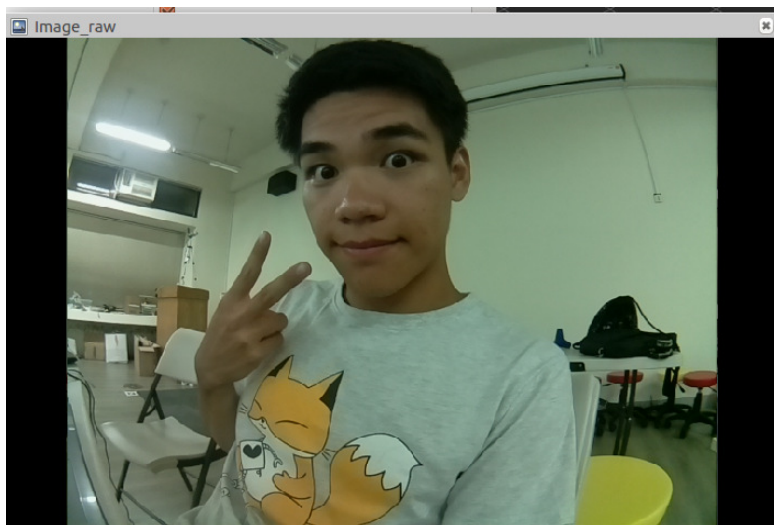


點選 Add->By topic->/duckiebot->/camera_node/image/raw/image

這將會顯示出相機擷取到的畫面

Check Point:

1. 使用 RViz 拍一張自己的臉並截圖(樓下是帥哥)<-it's not true



Task 2 相機校正 - intrinsic calibration

Task 2-1 啟用 ForwardX11

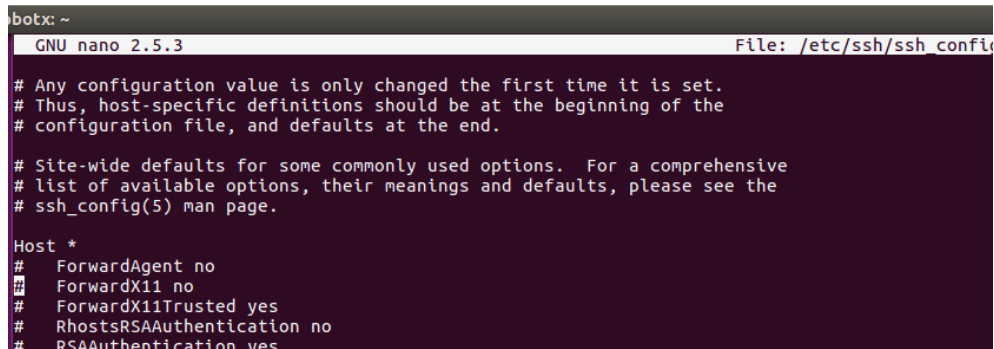
laptop \$ sudo vim /etc/ssh/ssh_config

更改 “**#ForwardX11 no**” 成為 “**ForwardX11 yes**”.

請注意該行最前面的 “#” 號需要被拿掉

存檔之後重開終端機

參考以下圖片：



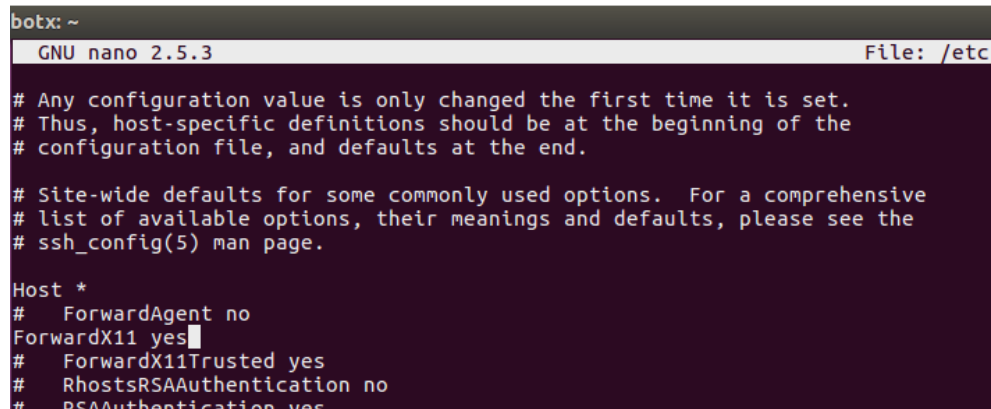
```
botx: ~
GNU nano 2.5.3 File: /etc/ssh/ssh_config

# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Host *
#   ForwardAgent no
#   ForwardX11 no
#   ForwardX11Trusted yes
#   RhostsRSAAuthentication no
#   RSAAuthentication yes
```

更改後應該長這樣：



```
botx: ~
GNU nano 2.5.3 File: /etc/ssh/ssh_config

# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Host *
#   ForwardAgent no
ForwardX11 yes
#   ForwardX11Trusted yes
#   RhostsRSAAuthentication no
#   RSAAuthentication yes
```

Task 2-2 設定 ROS 並開始校正

duckiebot \$ cd ~/duckietown

duckiebot \$ source environment.sh

duckiebot \$ source set_ros_master.sh **robotname**

duckiebot \$ byobu

(F2 開新分頁, F3&F4 切換分頁, F6 離開)

執行後畫面如下

```
mozard@mozard (10.0.1.56) - byobu
mozard@mozard:~$
```

下一步在 byobu 裡面執行指令

duckiebot \$ roslaunch duckietown camera.launch veh:=robotname raw:=true

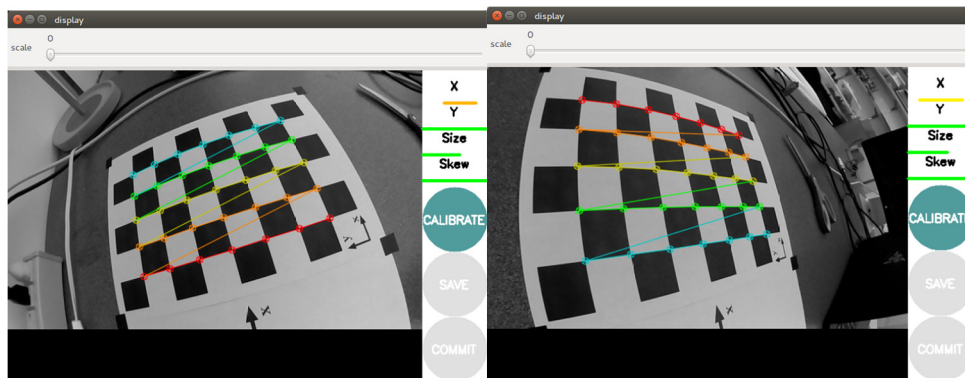
```
mozard@mozard (10.0.1.56) - byobu
mozard@mozard:~$ roslaunch duckietown camera.launch veh:=robotname raw:=true
```

按 F2 開啟新的分頁並輸入指令

```
mozard@mozard (10.0.1.56) - byobu
mozard@mozard:~$ roslaunch duckietown intrinsic_calibration veh:=robotname raw:=true
```

duckiebot \$ roslaunch duckietown intrinsic_calibration.launch veh:=robotname raw:=true

執行指令後會有視窗跳出。請將方格紙放在相機前並更換不同位置、角度，程式會自動收集校正樣本。



等到校正資料齊全後，右上角的參數條都會變成綠色 而且 'CALIBRATE' 按鈕可按下。按下 'CALIBRATE' 後大約等待 10 分鐘。如果你對校正結果滿意，按下 'COMMIT' 鍵，它將會自動存在 **duckiebot**。

位置在：

~/duckiefleet/calibrations/camera_intrinsic/**robotname**.yaml

Check Point:

1. 在 duckiebot 的資料夾 `~/duckiefleet/calibration/camera_intrinsic/` 中有一個 `robotname.yaml`

2. 你可以使用以下指令來看到校正後的圖片

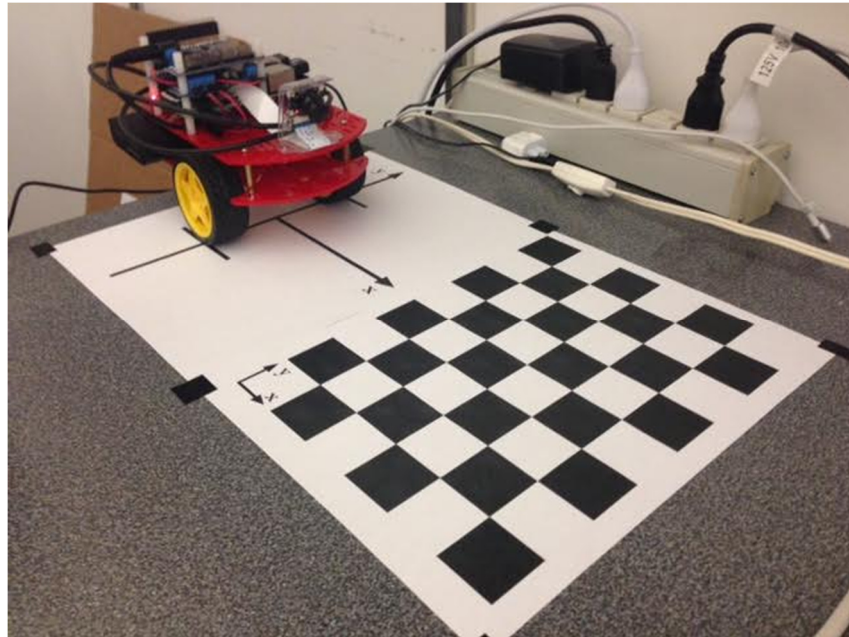
```
duckiebot $ roslaunch duckietown camera.launch veh:=robotname raw:=true rect:=true
```

rviz: Add->By topic->/duckiebot->/camera_node/image/rect/image 校正後的圖

rviz: Add->By topic->/duckiebot->/camera_node/image/raw/image 無校正的圖

Task 3 相機校正 - extrinsic calibration

將你的鴨子車如下圖擺放：



在 `duckiebot` 上執行 `byobu`，按下 **F2** 開啟兩個新的分頁，分別在兩個分頁中執行以下兩部分在第一個分頁中執行，

```
duckiebot $ cd ~/duckietown
```

```
duckiebot $ source environment.sh
```

```
duckiebot $ source set_ros_master.sh robotname
```

```
duckiebot $ roslaunch duckietown camera.launch veh:=robotname raw:=true
```

第二個分頁中，執行 `ground_projection.launch`

```
duckiebot $ roslaunch ground_projection ground_projection.launch veh:=robotname local:=1
```

另外開一個新的 terminal，確認 rosservice 存在 estimate_homography

```
laptop $ cd ~/duckietown
```

```
laptop $ source environment.sh
```

```
laptop $ source set_ros_master.sh robotname
```

```
laptop $ rosservice list
```

```
...
```

```
/veh_name/ground_projection/estimate_homography
```

```
/veh_name/ground_projection/get_ground_coordinate
```

```
...
```

在第三個終端機，執行以下的指令

```
laptop $ rosservice call /robotname/ground_projection/estimate_homography
```

Check point:

1. 假如以上都有成功，在 duckiebot 的資料夾
~/duckiefleet/calibration/camera_extrinsic/
中會有 **robotname**.yaml 檔

Reference

<https://docs.duckietown.org/DT19/index.html>