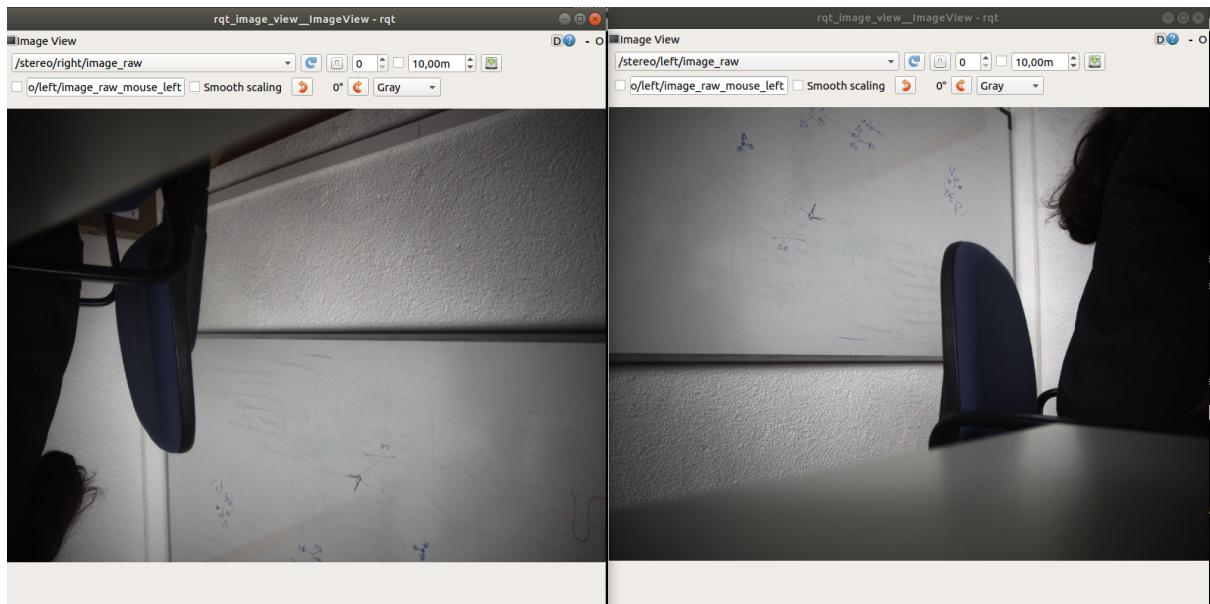


Modificacions flir_camera_driver per les CH3

Per fer servir el parell estéreo CH3 amb el `flir_camera_driver` s'ha afegit la rotació de la imatge de la càmera dreta ja que aquesta càmera està físicament girada 180° per a què hi càrga millor dins l'encapsulat.

Sense l'opció de flip les imatges obtingudes per cada càmera quedaven de la següent forma:



Pases prèvies:

Per fer funcionar aquest driver s'han d'instal·lar els següents paquets:

- Flycapture SDK
- Spinnaker SDK

Aquests paquets es poden descarregar des del següent enllaç:

<https://www.flir.com/products/chameleon3-usb3/>

RELATED PRODUCTS

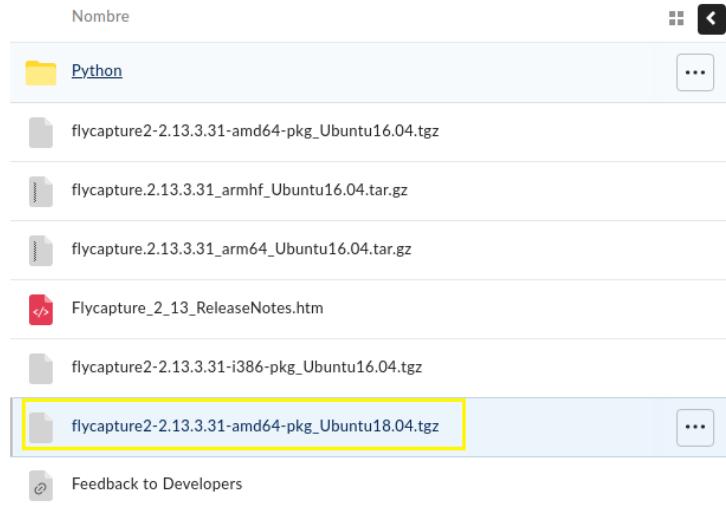


FLYCAPTURE SDK



SPINNAKER SDK

Per instal·lar el paquet Flycapture s'ha de descarregar i descomprimir el paquet. Dins el paquet descarregat hi ha un README amb les instruccions d'instal·lació.

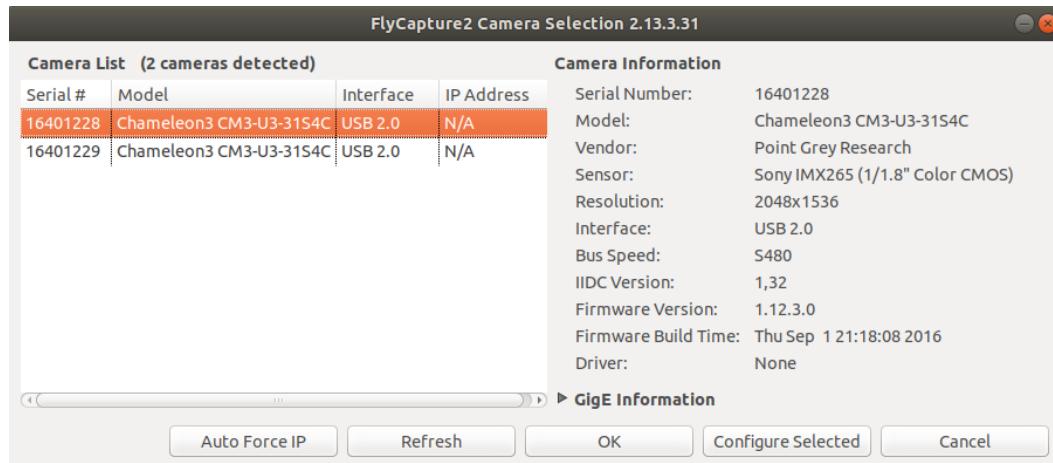


Per canviar el límit de tamany de les imatges que passem per USB fer:

```
$sudo sh -c 'echo 1024 > /sys/module/usbcore/parameters/usbfs_memory_mb'  
$ cat /sys/module/usbcore/parameters/usbfs_memory_mb
```

Un cop instal·lat si executam la comanda *flycap* hauríem de veure que detecta les dues càmeres:

```
$ flycap
```



*Després de instal·lar pot ser necessari reiniciar l'ordinador

Un cop instal·lat el paquet instal·lam l'Spinnaker seguint el mateix procediment: ens baixem el paquet *spinnaker-2.4.0.143-amd64*, el descomprimim i seguim les instruccions del README per a realitzar l'instal·lació.

Modificacions del driver

Per tal de girar la imatge s'ha modificat el paquet *spinnaker_camera_driver* per afegir una opció de rotació de la imatge de manera que les imatges de la càmera dreta es girin 180° i quedin en el mateix sentit que les de la càmera esquerra.

Per a fer-ho s'ha afegit el paràmetre *flip* a la funció *grabImage* de la classe *SpinnakerCamera*:

```
void SpinnakerCamera::grabImage(sensor_msgs::Image* image, const std::string& frame_id, bool& flip)
```

Quan aquest paràmetre està setejat es gira la imatge 180° i es canvia l'encoding per tal de que la imatge girada es visualitzi correctament:

```
uint8_t* raw_data = (uint8_t*) image_ptr->GetData();
//Flip if necessary (right camera is physically rotated 180 degrees)
if (flip==true){
    imageEncoding = sensor_msgs::image_encodings::BAYER_BGGR16;
    for(int i = 0; i < (height*width); i++){
        size_t idx2 = (height*width*2)-i;
        uint8_t aux = raw_data[i];
        raw_data[i]=raw_data[idx2];
        raw_data[idx2]=aux;
    }
}
```

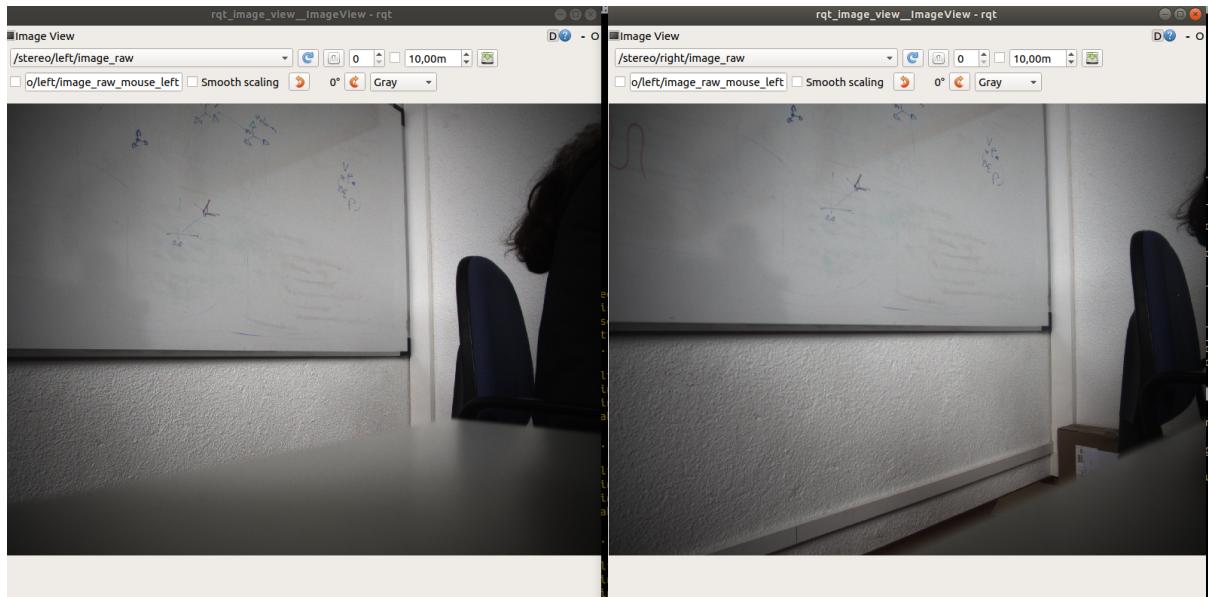
La funció *grabImage* de la classe *Spinnaker* es crida des de el node *nodelet.cpp* així que s'ha afegit el paràmetre *flip* també en aquest fitxer.

Finalment passam la variable *flip* per paràmetre des del launch *stereo.launch*. Donat que la càmera que es vol girar és la dreta setejam *flip* a *true* només per aquesta càmera:

```
<group ns="$(arg camera_name)" >
    <group ns="left" >
        <!-- Camera nodelet -->
        <node pkg="nodelet" type="nodelet" name="camera_nodelet"
              args="load"
spinnaker_camera_driver/SpinnakerCameraNodelet/camera_nodelet_manager">
            <param name="frame_id" value="camera_left" />
            <param name="serial" value="$(arg left_camera_serial)" />
            <param name="flip" value="false" />

<group ns="right" >
    <!-- Camera nodelet -->
    <node pkg="nodelet" type="nodelet" name="camera_nodelet"
          args="load"
spinnaker_camera_driver/SpinnakerCameraNodelet/camera_nodelet_manager">
        <param name="frame_id" value="camera_right" />
        <param name="serial" value="$(arg right_camera_serial)" />
        <param name="flip" value="true" />
```

Amb aquesta configuració s'haurien d'obtenir les imatges corregint la rotació física de la càmera dreta:



Cal·libració

Per cal·librar serà necessari el paquet `camera_calibration`:

```
$ rosdep install camera_calibration  
$ rosmake camera_calibration
```

Per iniciar el cal·librador s'ha d'executar la següent instrucció:

```
$ rosrun camera_calibration cameracalibrator.py --approximate 0.1 --size  
6x8 --square 0.04 right:=/stereo/right/image_raw  
left:=/stereo/left/image_raw right_camera:=/stereo/right  
left_camera:=/stereo/left
```

Moure el patró fins que s'activi l'opció de `calibrate`. Guardar la calibració amb `save` i pujar-la a les càmeres amb `commit`.

Per més detall sobre el procés de cal·libració es pot seguir aquest tutorial de ROS:

http://wiki.ros.org/camera_calibration/Tutorials/StereoCalibration

*`rqt_reconfigure` `rqt_reconfigure`