ActividadInterfsas

Generated by Doxygen 1.9.7

Chapter 1

Actividad Interfases

1.1 Detección del objeto verde

```
cd catkin_ws
source devel/setup.bash
roslaunch green_object_detection object_detection.launch
```

1.1.1 Nodos del launch file

1. publish_image.py: Inicializa el nodo para publicar imagen de la cámara en el tópico ∗/camera/image_raw∗

2. green_object_detection.py: Inicializa el nodo para la detección de objetos verdes en la imagen, usa una librería .so para multiplicar las coordenadas y publica el resultado en el tópico ∗/green_object_coordinates∗

3. coordServer.py: Inicializa un nodo para levantar un servidor gRPC que recibe un mensaje vacío y devuelve las coordenadas obtenidas del tópico de las coordenadas del objeto

1.1.2 Cliente en CSharp para obtener las coordenadas del objeto

```
cd ../csharp/csGrpc/csGrpc/bin/Debug/
mono csGrpc.exe
```

1.2 Servidor gRPC REST de Go

Para ejecutar el servidor primero hay que volver al root del repositorio y entrar en la carpeta de Go
`cd ../../../../../grpcgw`

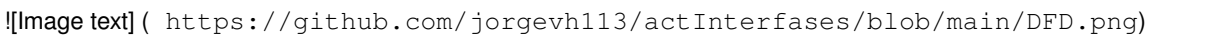
Ya en la carpeta de Go, se ejecuta en terminal:
`go run server/serverPS.go`

1.3 Servidor implementado en Flask con Python

Finalmente, hay que pasar a la carpeta de flask y ejecutar el código para poner en marcha el servidor
`python3 conn.py`

La página será visible en `http://localhost:50000/call-go-api`

1.4 Diagrama de Flujo de Datos



1.5 Aplicación Flask

La aplicación de Flask se utiliza para hacer una prueba del servidor REST creado con Golang. Para correr la aplicación, se utiliza conn.py y en la carpeta templates se encuentra el html donde se despliegan los datos obtenidos del servidor gRPC.

1.6 gRPC Gateway

1.6.1 Carpeta Server

Aquí se encuentra el ejecutable del servidor gRPC que actúa como un Gateway entre el servidor gRPC implementado en Python y otros clientes.

1.7 Proyecto de CSharp

1.7.1 Paquetes del proyecto

1. Google.Protobuf.3.2.0
2. Grpc.1.2.2
3. Grpc.Core.1.2.2
4. Grpc.Tools.1.2.0
5. System.Interactive.Async.3.1.1

1.7.2 Compilcación del ejecutable

Después de compilar el proyecto, se genera un archivo ejecutable que se puede correr con las herramientas de mono

1.8 Workspace de los nodos de ROS

1.8.1 Paquete green_object_detection

En este paquete se encuentran todos nodos de ROS para la transmisión de coordenadas de objetos dentro de la carpeta `src`

1.8.2 Carpeta msgs_proto

En esta carpeta se encuentra el archivo proto donde se definen los mensajes y servicios gRPC. Además, se encuentran el código `coordServer.py` que pone en servicio el servidor gRPC encargado de transmitir las coordenadas del tópico de ∗/gren_object_coordinates∗.

1.8.3 Carpeta systemObject

En esta carpeta se encuentran archivos necesarios para poner a disposición de uso la librería de sistema, tanto el código de c++ donde se declara la función, como los archivos compilados para generarla.

1.8.4 Carpeta src

green_object_detection.py es el nodo encargado de analizar la imagen con la librería opencv y encontrar objetos verdes. *publish_image.py* es un nodo que se utiliza para publicar la imagen en el tópico ∗/camera/image_raw∗

Index