

Detección de personas para un robot TurtleBot usando OpenCV — Guía de uso

Estructura de archivos

informe_Aguado_Ender.pdf — El informe
evaluacion/ — Lo referido a la evaluación de algoritmos
evaluacion/clasificadores/ — Los clasificadores utilizados
evaluacion/ejecutables/ — Ejecutables compilados para Linux x64
evaluacion/resultados/ — Resultados sobre los sets de prueba
evaluacion/scripts/ — *Scripts* de evaluación
evaluacion/src/ — Código fuente de los programas de evaluación
implementacion/ — Nodos de ROS para funcionamiento en línea.
otros/calibracion_turtlebot/ — Archivos de calibración de las cámaras
otros/raspicam_node-master.zip — Nodo RASPICAM
resultados/ — Imágenes y videos de los resultados sobre los sets de prueba y validación

Nodos de detección

El *workspace* de ROS contiene el nodo de detección y un nodo publicador de video para la *webcam*. Para poder ejecutarlos, primero hay que compilarlos.

Los pasos a seguir:

1. Modificar el código (`src/detpeople/src/ImageConverter.cpp`, líneas 48 y 51) para que se suscriba a los mensajes de la KINECT o de la *webcam* o de la cámara de la RASPBERRY PI. Por defecto está suscripto a los mensajes bajo el topic `/camera/image`, que son los de la *webcam* o de la cámara de la RASPBERRY.
2. Compilar:
 - a) Situar dentro de la carpeta `implementacion`
 - b) Ejecutar `catkin_make`. Esto compila los nodos.
 - c) Se le informa a ROS la existencia de esos paquetes con `source devel/setup.bash`
3. Se ejecutan los nodos ¹ con `roslaunch`
 - a) [Opcional] `roslaunch detpeople publicador_video` — Ejecuta un nodo que publica video desde la *webcam* de la PC.
 - b) `roslaunch detpeople detpeople` — Ejecuta el detector.

Para recibir imágenes desde el TURTLEBOT, ver la Sección 7.2. Para transmitir imágenes desde la RASPBERRY PI, referirse a la Sección 7.3; el nodo RASPICAM se puede compilar de forma similar a lo realizado en el punto 2 anterior.

Programas de evaluación

El programa que realiza las detecciones es `dp_opencv`. Este programa guarda en un archivo *csv* los datos de las detecciones, y en un *txt* los parámetros que se utilizaron. Fue diseñado para que fácilmente puedan añadirse y utilizarse otros detectores. De hecho, para extraer manualmente las coordenadas reales se utilizó una clase detectora. Aquí, `DetectorFinal` es el detector en cascada. Los parámetros de los detectores están explicados en la Sección 5.

```
Uso: dp_opencv < carpeta/con/imagenes/rgb >  
          < carpeta/con/imagenes/profundidad | 0 > (0 no carga nada)  
          < numero de imagen inicial > (suponiendo que empiezan en 1)  
          < numero de imagen final | 0 > (0 procesa hasta el final)  
          < dir/al/archivo_de_resultados > (sin extensión)  
          < 1 | 0 > (mostrar_detecciones)  
          < clase_de_detector > [parámetro_1_nombre parámetro_1_valor ...]  
Crea un archivo csv con los resultados y un txt con información sobre el detector usado.
```

¹Cada nodo se ejecuta en una terminal o consola diferente.

```

Detectores válidos:
* Extractor
* DetectorHOG pasoEscala double umbralAgrupamiento int hit_threshold float setSVMDetector <
  getDefaultPeopleDetector | getDaimlerPeopleDetector> convertir_a_gris bool
  ecualizar_histograma bool blurear bool tamaño_blur int filtro_enfoque bool
  filtro_repujado bool filtro_enfoque_y_repujado bool filtro_repujado_y_enfoque bool
  usar_profundidad_altura bool escala_inicial float
* DetectorFinal direccion_a_cascada string escala_inicial float convertir_a_gris bool
  ecualizar_histograma bool scaleFactor float minNeighbors int usar_profundidad_altura bool
  blurear bool tamaño_blur int filtro_enfoque bool filtro_repujado bool
  filtro_enfoque_y_repujado bool filtro_repujado_y_enfoque bool

```

El programa **dp_resultados** compara las detecciones de un csv con las coordenadas reales del set. Guarda un archivo con los resultados.

```

Uso: dp_resultados /ruta/a/set/de/imagenes
                        /ruta/a/datos_reales (sin extensión)
                        /ruta/a/datos_estimados (sin extensión)
                        /ruta/a/archivo_de_resultados.txt (con extensión)

```

La evaluación completa de los sets mediante *scripts* de BASH es compleja. Puede ejecutarse **./_llamartodos.sh** (previamente darle permiso de ejecución a todos los scripts²) si se indican las direcciones a los sets de imágenes correspondientes; para esto, referirse a los *scripts* **set*.sh**. Si se desea analizar el funcionamiento del conjunto, recomendamos empezar por **./_llamartodos.sh** y luego dirigirse a los *scripts* que se van importando con **source**.

Pablo Aguado

Fabrizio Emden

San Juan, mayo de 2016

²chmod +x *.sh