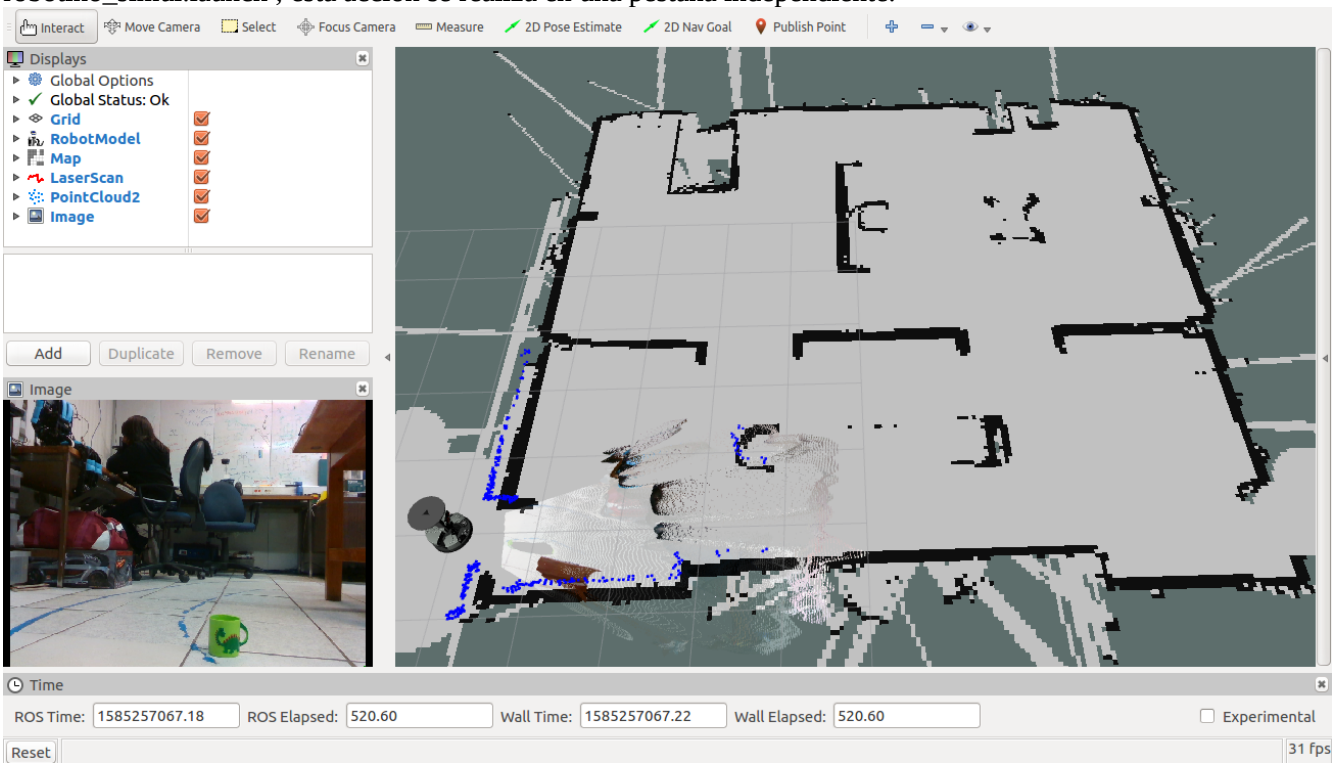
Gómez Castro Joel Francisco Robots Móviles Grupo 2 Práctica 3

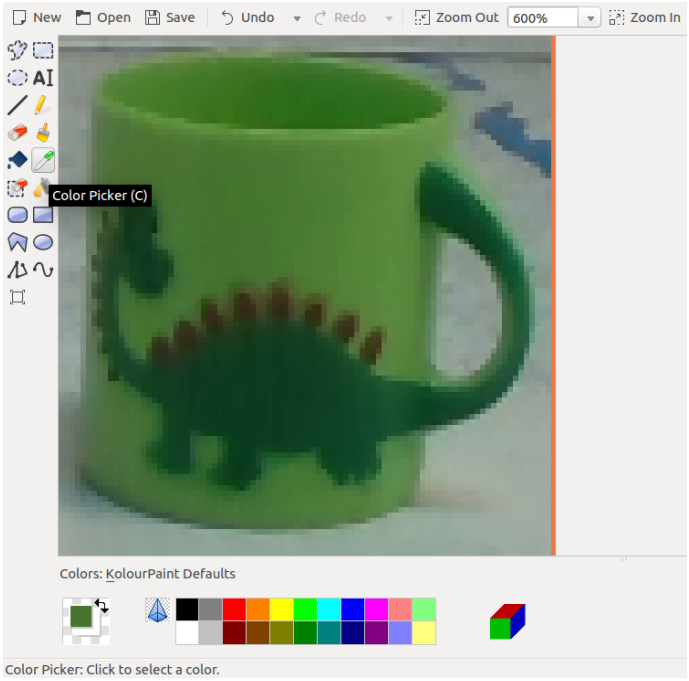
Segmentación por color

Para esta actividad, primero en la consola, introducimos y ejecutamos el siguiente comando: roslaunch bring_up robotino_simul.launch ; esta acción se realiza en una pestaña independiente.

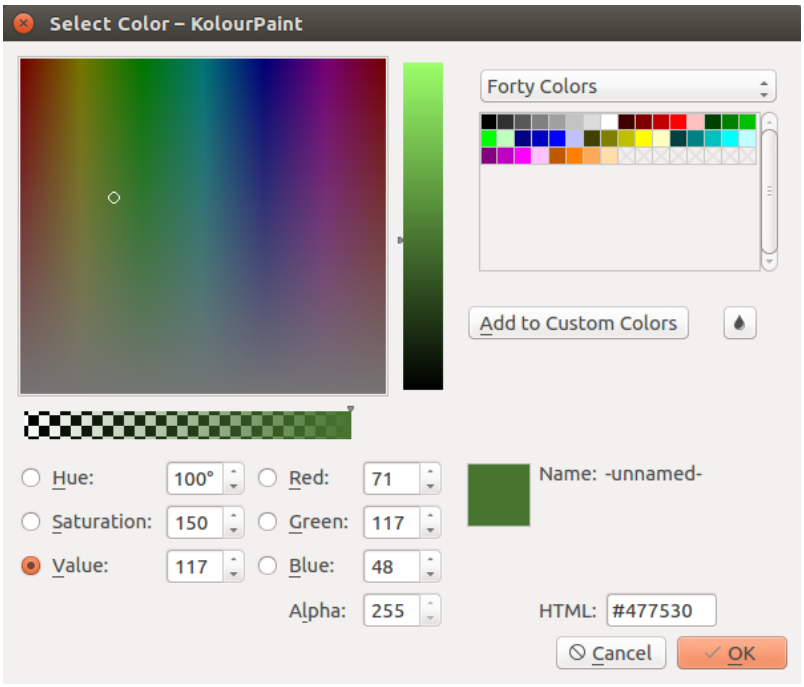


Si abrimos otra pestaña de la consola (ctrl+shift+T) debemos introducir y ejecutar el siguiente nodo: rosrun vision color_segmentation.py, y aparecerán tres pestañas, una con la visualización de la imagen de la interfaz de RVIZ en sus colores RGB, otra con la codificación de la imagen en sus valores HSV y una en valores binarios.

En la última pestaña de la consola, introducimos ctrl+C y luego des esto se hace lo mismo en donde se abrio RVIZ. Ahora nos vamos a archivos (no en la consola) y de ahí nos vamos al directorio catkin_ws/src/vision/training y abrimos en KoulourPaint el archivo GreenCup.png.



Tenemos como objetivo principal segmentar el fondo de color verde claro de la tasa, sin importar el dinosaurio. Una vez en KolourPaint, le damos color picker y seleccionamos un color de la taso (dando mayor prioridad al color objetivo que detecte para convertirlo en binario). Una vez que le damos clic en donde queremos saber los valores HSV, le damos en el color donde se identifica dos veces clic, y ahí aparece una ventana con esas y otras características del color detetcado.



Se tomaron varias muestras, en muchas partes en donde nosotros queremos segmentar, para obetener un rango de valores HSV aproximado.
En consola, nos dirigimos a la carpeta catkin_ws/src/vision/scripts/ y en la consola abriremos el archivo color_segmentation.py, para ello introduciremos atom color_segmentation.py.

Dentro del archivo, vamos a modificar la función in range, ya que en su sintaxis cv2.inRange(img_hsv, numpy.array([H_min,S_min,V_min]), numpy.array([H_max, S_max, V_max])). El rango que tenemos que modificar va de acuerdo a las muestras obtenidas. En este caso el rango es:

H_min=45	H_max=58
S_min=60	S_max=255
V_min=90	V_max=255

Ahora, vamos a introducir y ejecutar de nuevo los comandos mencionados al inicio, y se tiene como resultado (en el campo de binario):

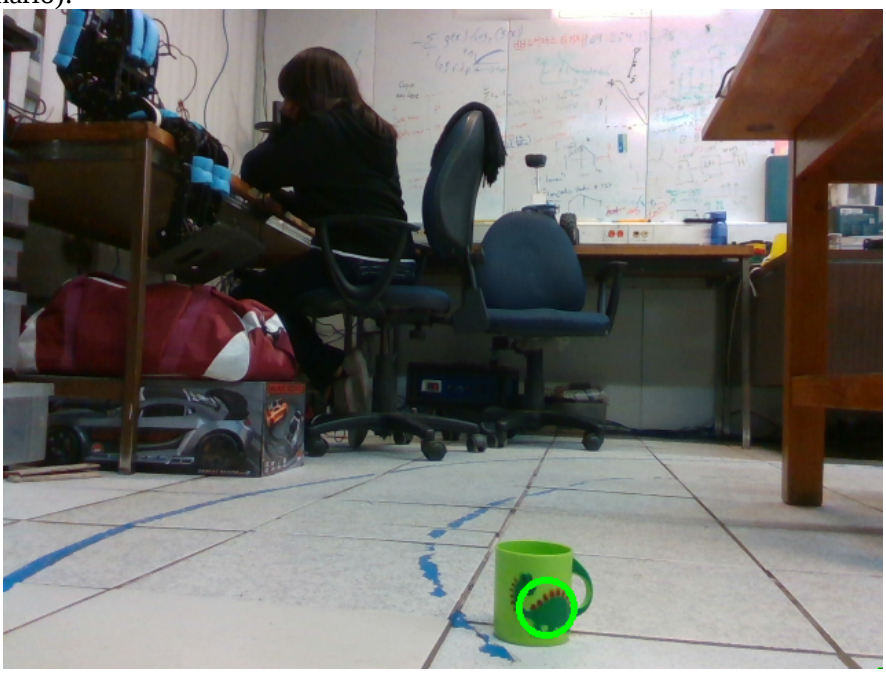


Imagen BGR

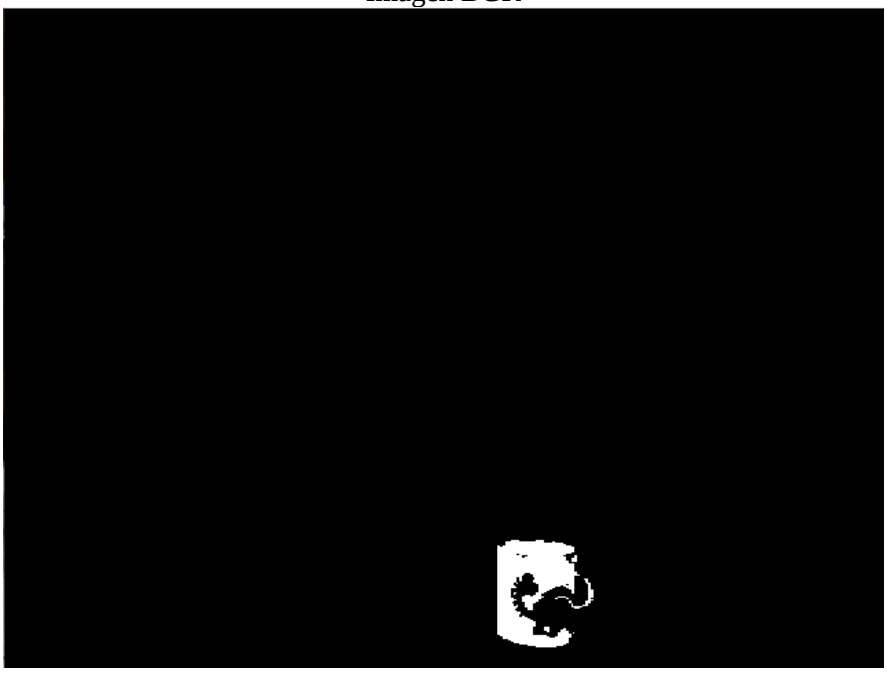


Imagen en Binario o el resultado de la segmentación

Conclusiones

La segmentación por colores es uno de los fundamentos importantes de la visón artificial, ya que nuestros sistemas digitales solo detectan señales binarias (valores 0 y 1). En este caso el valor 1 (color blanco) es el objeto que puede detectar nuestro sistema, pero parte de esa detección se debe conocer muy bien el rango de valores HSV para que se efectue esa conversión RGB a binario, con ayudar de color_segmentation.py. Con respeto al resultado obtenido, se logro segmentar muy bien la región deseada de la taza, debido al numero de muestras tomadas de los distintos valores de HSV en diferentes partes de la región objetivo (fueron cuarenta muestras). Cabe destacar que uno de los problemas encontrados en la realización de la práctica fue la confusión del valor H (Hue), ya que en Kolour paint se muestra un rango de 0 a 360 y OpenCV acepta solo hasta 180, entonces de los valores máximos y mínimos que se obtuvieron gracias al muestreo, en color_segmentation.py se introdujeron esos valores pero divididos entre dos.