

Robots Móviles

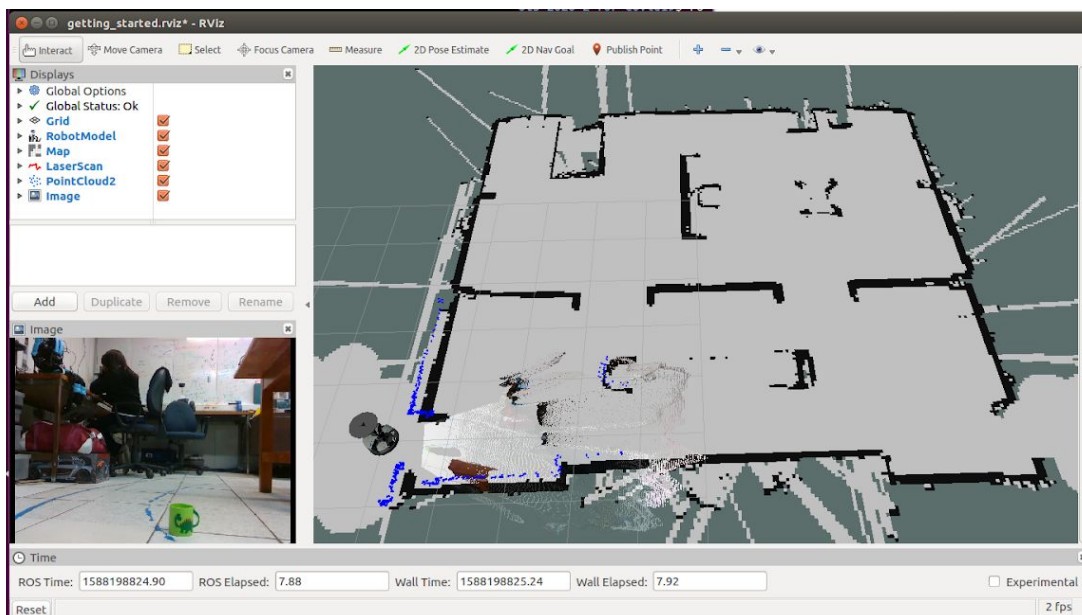
Rivera Esquivel Jennifer Estefanny

Práctica No.6

Segmentación por color.

Entregables:

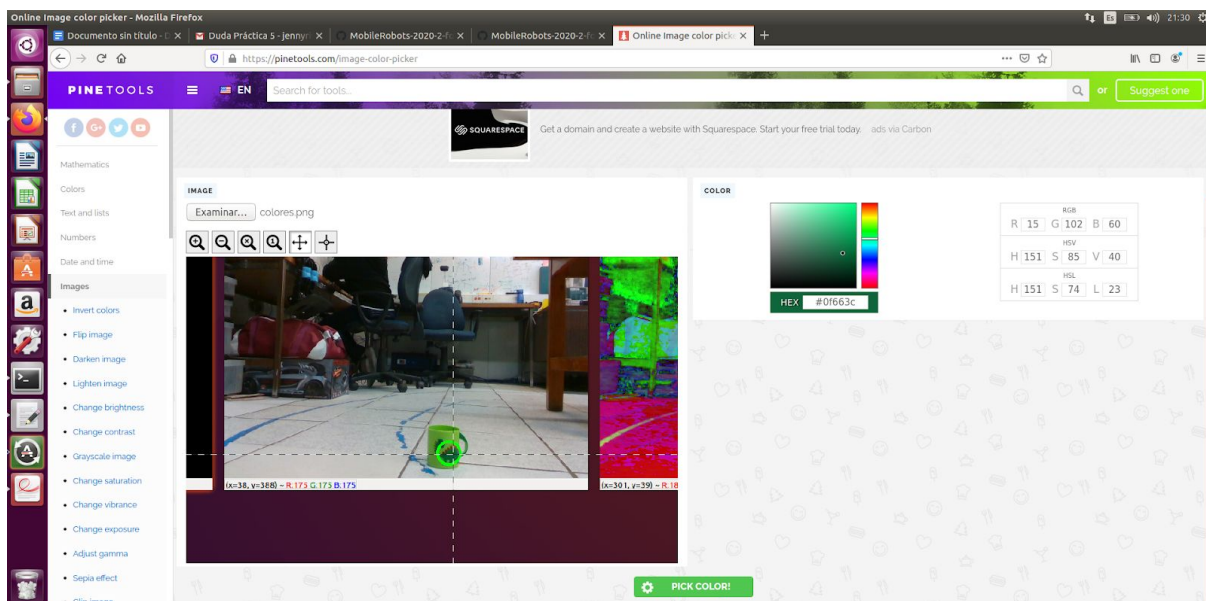
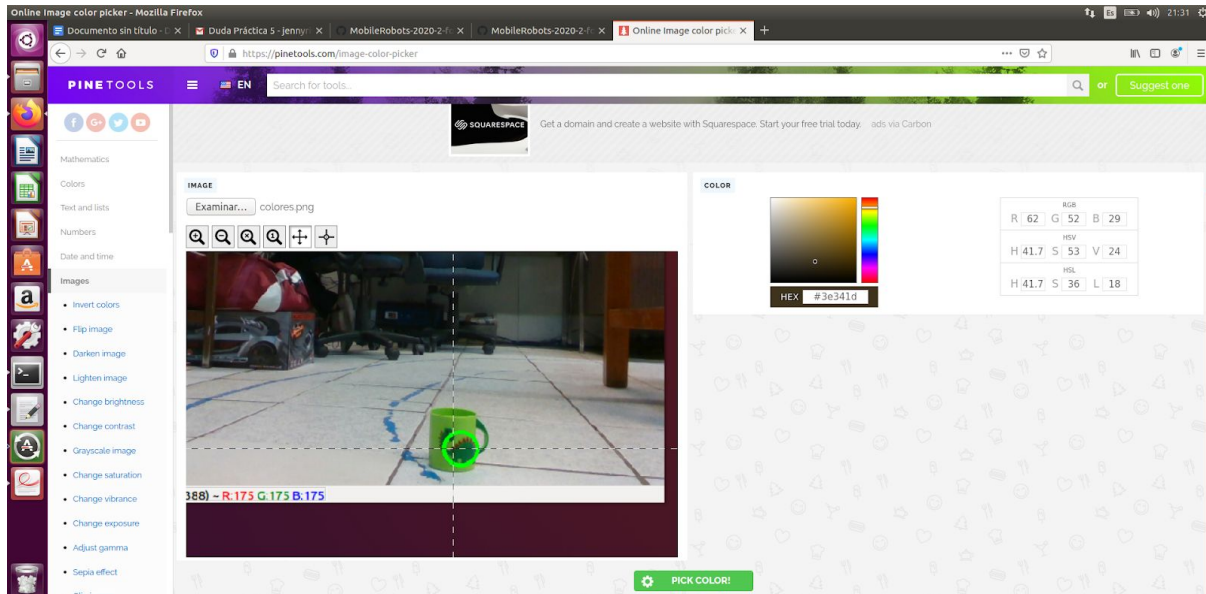
Ejecutar el comando `roslaunch bring up robotino simul.launch`

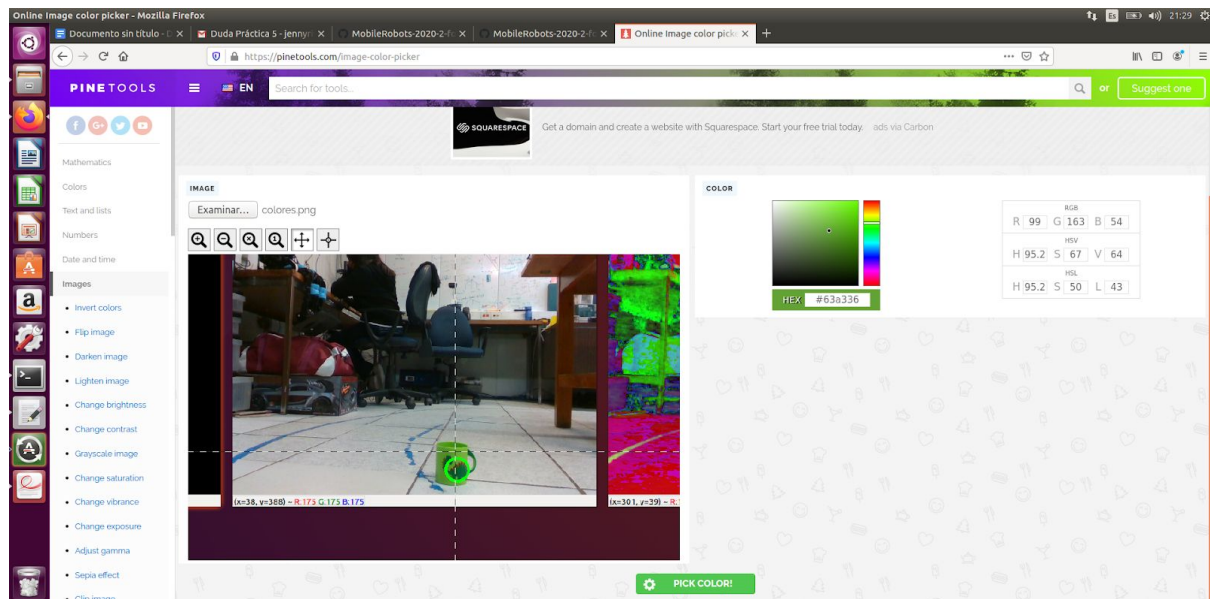


Ejecutar el nodo `roslaunch vision color segmentation.py`



Utilizando la ventana “Image BGR”, obtener una captura de pantalla, guardarla y abrirla con cualquier editor de imágenes (Kolour Paint, por ejemplo) para obtener los valores HSV de la taza o de la lata de Cola-Cola.





Resultados de los valores HSV

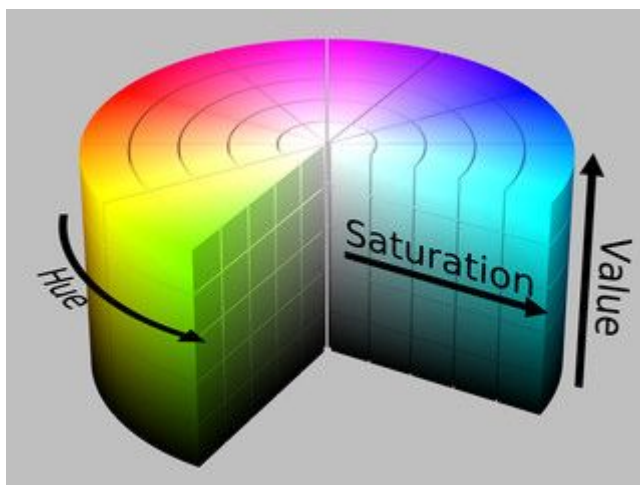
Cafe: 41.7, 53, 24

Verde fuerte: 151, 85, 40

Verde claro: 95.2, 67, 64

Modificar el archivo catkin ws/src/vision/scripts/color segmentation.py, en la función inRange, para segmentar correctamente la taza verde o la lata de Coca-Cola.

InRange compara cada píxel de la imagen para determinar si se encuentra dentro de estos límites.



El espacio de color HSV (matiz, saturación, valor) es un modelo para representar el espacio de color similar al modelo de color RGB. Dado que el canal de tono modela el tipo de color, es muy útil en tareas de procesamiento de imágenes que necesitan segmentar objetos en función de su color. La variación de la saturación pasa de no saturada a representar tonos

de gris y completamente saturada (sin componente blanco). El canal de valor describe el brillo o la intensidad del color. La siguiente imagen muestra el cilindro HSV.

```
segmentation.py (~/Escritorio/Robotics/MobileRobots-2020...or-Covid19/catkin_ws/src/vision/scripts) - gedit
Abrir  [icon]

#!/usr/bin/env python

import rospy
import cv2
import cv_bridge #
import numpy
from sensor_msgs.msg import Image

def callback_rgb(msg):
    bridge = cv_bridge.CvBridge()
    img_bgr = bridge.imgmsg_to_cv2(msg, desired_encoding="bgr8")
    img_hsv = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2HSV)
    img_bin = cv2.inRange(img_hsv, numpy.array([40,110,85]), numpy.array([85, 255, 255]))
    #img_bin = cv2.inRange(img_hsv, numpy.array([60,200,60]), numpy.array([80, 255, 255]))

    idx = cv2.findNonZero(img_bin)
    [centroid_x, centroid_y, a, b] = cv2.mean(idx)
    #circunferencia 40/20
    cv2.circle(img_bgr, (int(centroid_x), int(centroid_y)), 40, [0, 255, 0], thickness=3)
    cv2.imshow("Image BGR", img_bgr)
    cv2.imshow("Image HSV", img_hsv)
    cv2.imshow("Image Binary", img_bin)
    cv2.waitKey(1)

def main():
    print "INITIALIZING COLOR SEGMENTATION NODE..."
    rospy.init_node("color_segmentation")
    rospy.Subscriber("/camera/color/image_raw", Image, callback_rgb )
    loop = rospy.Rate(30)
    while not rospy.is_shutdown():
        loop.sleep()

if __name__ == "__main__":
    main()
```

Resultado:



Problemas encontrados:

Me tomó tiempo elegir los valores para segmentar la taza verde adecuadamente, a pesar de tener los valores HSV, no sabía cómo establecer los límites en la función `inRange`. La forma en la que trabaja esta función no es tan complicada, sin embargo tuve que estar jugando con los valores para reconocer la taza de manera completa. Para que se hiciera más grande la circunferencia aumente al doble el valor 20 en `cv2.circle`.