***Código Ordenador pruebas***

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

"""

Created on Mon Jun 12 16:12:22 2017

@author: samuel

"""

#importamos las librerias con las que vayamos a trabajar

import numpy as np

import cv2

import time

import math

nContadorError = 0

def region\_of\_interest(img,vertices):

mask = np.zeros\_like(img)

if len(img.shape) > 2:

channel\_count = img.shape[2]

ignore\_mask\_color = (255,) \* channel\_count

else:

ignore\_mask\_color = 255

cv2.fillPoly(mask, vertices, ignore\_mask\_color)

masked\_image = cv2.bitwise\_and(img,mask)

return masked\_image

def draw\_lines(img\_without,img):

global nContadorError

linesP = cv2.HoughLines(img\_without,1,np.pi/180,120)

nContadorRecta = 0

nContadorDcha = 0

nContadorIzq = 0

try:

for rho,theta in linesP[0]:

a = np.cos(theta)

b = np.sin(theta)

x0 = a\*(rho)

y0 = b\*(rho)

x1 = int(x0 + 1000\*(-b))

y1 = int(y0 + 1000\*(a))

x2 = int(x0 - 1000\*(-b))

y2 = int(y0 - 1000\*(a))

#izquierda

if math.degrees(theta)>105 and math.degrees(theta)<115:

if y1<600 and y2<680:

nContadorIzq = nContadorIzq + 1

nContadorError = 0

else:

#recto

if ((math.degrees(theta)>50 and math.degrees(theta)<75) and rho>500) or ((math.degrees(theta)<140 and math.degrees(theta)>115) and rho<200):

nContadorRecta = nContadorRecta + 1

else:

if (math.degrees(theta)>78 and math.degrees(theta)<83) and rho<750:

nContadorDcha = nContadorDcha + 1

except TypeError, UnboundLocalError:

print 'Error, no se econtraron lineas'

print (nContadorRecta, nContadorDcha, nContadorIzq)

if (nContadorRecta >= nContadorDcha):

if (nContadorRecta >= nContadorIzq):

print ('delante')

else:

print ('izquierda')

else:

if (nContadorDcha >= nContadorIzq):

print ('derecha')

else:

print ('izquierda')

if (nContadorRecta ==0) and (nContadorIzq ==0) and (nContadorDcha == 0):

nContadorError = nContadorError + 1

print nContadorError

return img

def filters(img):

img = cv2.erode (img,cv2.getStructuringElement(cv2.MORPH\_RECT,(3,3)),iterations = 1)

img = cv2.dilate (img,cv2.getStructuringElement(cv2.MORPH\_RECT,(5,5)),iterations = 1)

return img

cv2.namedWindow('webcam')

vc = cv2.VideoCapture('/Users/samuel/Desktop/La charly 3/tfg/sombraarribacu.mp4')

#Define la region de interest que deseamos en este caso un trapecio

bot\_left = [0,950]

bot\_right = [1900,950]

appex\_left = [0,600]

appex\_right = [1900,600]

v = [np.array([bot\_left,bot\_right,appex\_right,appex\_left], dtype=np.int32)]

# Define el rango de blanco en HSV

lower\_blanco = np.array([5,5,225])

upper\_blanco = np.array([179,85,235])

while (True):

global nContadorError

# Leemos de lo que estamos capturando

ret, frame = vc.read()

#Selección region de interes

frame = region\_of\_interest(frame,v)

# Convierte BGR a HSV

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

# Limpa la imagen de imperfecciones con los filtros erode y dilate

hsv = filters (hsv)

#Difumina la mascara para suavizar los contornos y aplicamos filtro canny

hsv = cv2.GaussianBlur(hsv, (5, 5), 0)

# Filtro color blanco

hsv = cv2.inRange(hsv, lower\_blanco, upper\_blanco)

#Saca el contorno con la funcion Canny

prueba = cv2.Canny(hsv,100,300,apertureSize = 3)

#Funcion para dibujar las lineas rectas en nuestra imagen 'frame'

frame = draw\_lines(prueba,frame)

hsv = cv2.resize(hsv,(640, 480), interpolation = cv2.INTER\_CUBIC)

frame = cv2.resize(frame,(640, 480), interpolation = cv2.INTER\_CUBIC)

cv2.imshow('frame',frame)

cv2.imshow('res',hsv)

if nContadorError == 100:

cv2.DestroyAllWindows()

break

tecla = cv2.waitKey(33) & 0xFF

if tecla!=255:

if tecla==27:

cv2.DestroyAllWindows()

break

elif tecla==13:

time.sleep(10)

print 'pausado'

elif tecla==-1:

continue

else:

print ('Debe pulsar ESC para salir no: ',tecla)

cv2.DestroyAllWindows()

***Código prueba motores***

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

GPIO.setup(22, GPIO.OUT)

GPIO.setup(23, GPIO.OUT)

GPIO.setup(26, GPIO.OUT)

GPIO.setup(5, GPIO.OUT)

GPIO.setup(27, GPIO.OUT)

GPIO.setup(24, GPIO.OUT)

GPIO.setup(16, GPIO.OUT)

GPIO.setup(6, GPIO.OUT)

def delante():

GPIO.output(22,GPIO.HIGH)

GPIO.output(23,GPIO.HIGH)

GPIO.output(26,GPIO.HIGH)

GPIO.output(5,GPIO.HIGH)

time.sleep(1)

GPIO.output(22,GPIO.LOW)

GPIO.output(23,GPIO.LOW)

GPIO.output(26,GPIO.LOW)

GPIO.output(5,GPIO.LOW)

time.sleep(1)

def atras():

GPIO.output(27,GPIO.HIGH)

GPIO.output(24,GPIO.HIGH)

GPIO.output(16,GPIO.HIGH)

GPIO.output(6,GPIO.HIGH)

time.sleep(1)

GPIO.output(27,GPIO.LOW)

GPIO.output(24,GPIO.LOW)

GPIO.output(16,GPIO.LOW)

GPIO.output(6,GPIO.LOW)

time.sleep(1)

def izquierda():

GPIO.output(27,GPIO.HIGH)

GPIO.output(23,GPIO.HIGH)

GPIO.output(16,GPIO.HIGH)

GPIO.output(5,GPIO.HIGH)

time.sleep(1)

GPIO.output(27,GPIO.LOW)

GPIO.output(23,GPIO.LOW)

GPIO.output(16,GPIO.LOW)

GPIO.output(5,GPIO.LOW)

time.sleep(1)

def derecha():

GPIO.output(22,GPIO.HIGH)

GPIO.output(24,GPIO.HIGH)

GPIO.output(26,GPIO.HIGH)

GPIO.output(6,GPIO.HIGH)

time.sleep(1)

GPIO.output(22,GPIO.LOW)

GPIO.output(24,GPIO.LOW)

GPIO.output(26,GPIO.LOW)

GPIO.output(6,GPIO.LOW)

time.sleep(1)

while (True):

opcion = input()

if opcion == 'w':

delante()

elif opcion == 'a':

izquierda()

elif opcion == 's':

atras()

elif opcion == 'd':

derecha()

else:

print 'quieto'

opcion = '2'