

12-4-2020

VISIÓN ARTIFICIAL PRÁCTICA 1



Jorge Alonso Vivar, Álvaro Robles Sánchez,
Chengian Li.
CURSO 4

1. Ejercicio 1.....	2
2. Ejercicio 2.....	2
3. Ejercicio 3.....	2
Imágenes	3
Ejercicio 1	3
Ejercicio 2	3
Ejercicio 3	4
Bibliografía	4

Ejercicio 1

En primer lugar, leemos todas las imágenes de entrenamiento y obtenemos sus keypoints (función “createBase”), para ello, creamos un flann y recorremos todas las imágenes almacenando sus descriptores y vectores. Los vectores se obtienen en la función “kps” realizando cálculos algebraicos.

A continuación, procedemos a leer las imágenes del test. En la función “detectImage” donde hemos cambiado las imágenes para que busque mejor los keypoint con las opciones vistas en clase con “filter2D” para hacer un filtrado espacial para las imágenes borrosas y “getStructuringElement” para realizar un borde morfológico con gradiente. Luego ya usamos el método “detectAndCompute” para obtener los keypoints de las imágenes y usamos sus descriptores para crear matches con los descriptores del flann usando el método “knnMatch”.

Con esos matches llamamos a la función “tabla_votación”, en la que usaremos la fórmula vista en clase para crear un vector de votación y trasladar ese voto a la tabla.

Por último, se extrae la celda con el máximo numero de votos y se dibuja un círculo en esa posición de la imagen para identificar el centro.

Ejercicio 2

Para cada una de las imágenes leemos el archivo xml para obtener los datos de los coches con el método “CascadeClassifier”.

Tras ello, identificamos los donde se encuentra el coche con el método “detectMultiScale”, una vez conocida su posición, usamos el mismo método para localizar su matrícula y dibujamos un rectángulo en ella para identificarla.

Ejercicio 3

Primero obtenemos los frames del video y analizamos cada uno de ellos, identificamos si hay un coche y definimos donde se encuentra usando la función “detectImage” del ejercicio 1.

Imágenes

Ejercicio 1



Ejercicio 2



Ejercicio 3



Bibliografía

https://docs.opencv.org/4.2.0/d1/d89/tutorial_py_orb.html

https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html

<https://docs.scipy.org/doc/numpy/reference/generated/numpy.power.html>

<https://unipython.com/feature-matching-comparacion-funciones/>

<https://pythonexamples.org/python-opencv-cv2-resize-image/>

<https://www.it-swarm.dev/es/python/python-convertir-radianes-grados/942021488/>

<https://jonathanmelgoza.com/blog/tratamiento-de-imagenes-en-python-con-opencv/>

https://docs.scipy.org/doc/numpy/reference/generated/numpy.unravel_index.html

<https://es.wikihow.com/encontrar-la-magnitud-de-un-vector>

<https://www.geeksforgeeks.org/numpy-where-in-python/>

<http://vision.cs.utexas.edu/378h-fall2015/slides/lecture8.pdf>

<https://stackoverflow.com/questions/9170906>

<http://opencvexamples.blogspot.com/2013/10/create-structuring-element-for.html>